Filomat 35:9 (2021), 3071–3090
https://doi.org/10.2298/FIL2109071K

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

On the CRI Method for Solving Sylvester Equation with Complex Symmetric Positive Semi-Definite Coefficient Matrices

Gholamreza Karamali[a], Akbar Shirilord[a], Mehdi Dehghan[b]

[a]*Faculty of Basic Sciences, Shahid Sattari Aeronautical University of Sciences and Technology, South Mehrabad, Tehran, Iran*
[b]*Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology, No. 424, Hafez Ave.,15914, Tehran, Iran*

Abstract. Combination of real and imaginary parts (CRI) method is an efficient method for solving a class of large sparse linear systems with complex symmetric positive semi-definite coefficient matrices. In this work we will extend CRI approach to determine the approximate solution of Sylvester equation with complex symmetric semi-definite positive coefficient matrices. We show that the new algorithm converges unconditionally to the unique exact solution of the Sylvester matrix equation. In the end we test the new scheme by solving some numerical examples.

1. Introduction

Algebraic Sylvester matrix equations are observed in many areas from different regions such as, control theory and many other branches of engineering [7–9, 23].

The so-called bilinear control system can be described by the following state-space

$$\begin{cases} \dot{x}(t) = Ax(t) + \sum_{j=1}^{m} N_j x(t) u_j(t) + Bu(t), \\ y(t) = \tilde{C}x(t), \;\; x(0) = x_0, \end{cases} \tag{1}$$

where t is the time variable, $x(t) \in \mathbb{C}^n$, $u(t) = [u_1(t), ..., u_m(t)]^T \in \mathbb{C}^m$ and $y(t) \in \mathbb{C}^n$ are the stable, input and output vectors, respectively. Also $B(t) \in \mathbb{C}^{n\times m}$, $\tilde{C}$, $A \in \mathbb{C}^{n\times n}$. Reachability and observability are two important issues for the system (1), such that the reachability is defined by

$$R = \sum_{k=1}^{\infty} \int_0^{\infty} ... \int_0^{\infty} R_k R_k^T \, dt_1 ... dt_k,$$

that is the solution of Eq. (1), where

$$R_1 = e^{At_1} B \qquad \text{and} \qquad R_k(t_1, ..., t_k) = e^{At_k}[N_1 R_{k-1}, ..., N_m R_{k-1}], \qquad k = 2, 3, ...$$

2020 *Mathematics Subject Classification*. 15A30, 15A69, 65F10.
Keywords. Complex Sylvester matrix equation, CRI iteration method, Convergence.
Received: 07 March 2021; Revised: 26 May 2021; Accepted: 20 September 2021
Communicated by Dragan S. Djordjević
Corresponding author: Gholamreza Karamali
Email addresses: rezakaramali918@gmail.com (Gholamreza Karamali), akbar.shirilord@aut.ac.ir (Akbar Shirilord), mdehghan@aut.ac.ir (Mehdi Dehghan)

Also the observability is the solution of the dual equation for

$$AY + YA^T + \sum_{j=1}^{m} N_j Y N_j^T = \tilde{C}^T \tilde{C},$$

where $Y \in \mathbb{C}^{n\times n}$ must be determined. It is well-known that these matrix equations have applications in various areas and have been widely used in engineering and scientific computations. Liao and et.al [19] introduced best approximate solution of matrix equation

$$AXB + CYD = Q. \tag{2}$$

Benner [3] has proposed a new method for solving stable Sylvester equations right-hand side given in factored form

$$AX + XB = FG,$$

that arise in model reduction problems.

He, Wang and Zhang [15] provided some necessary and sufficient conditions for the existence and the general solution to the system of four coupled one-sided Sylvester-type real quaternion matrix equations

$$\begin{cases} A_1X_1 + X_2B_1 = C_1, \\ A_2X_2 + X_3B_2 = C_2, \\ A_3X_3 + X_4B_3 = C_3, \\ A_4X_4 + X_5B_4 = C_4. \end{cases}$$

Author of [5] has introduced a numerical method for solving algebraic Riccati equations

$$A^TX + XA - XFX + G = 0,$$

based on a modification of matrix sign-function. Dehghan and Hajarian [6] considered second-order Sylvester matrix equation

$$EVF^2 - AVF - CV = BW, \tag{3}$$

and proposed an efficient iterative method for solving it. Some useful articles about matrix equation can be found in [3–5, 11, 14–17, 20, 21, 24, 26, 27].

Here we focus on the Sylvester matrix equation of the form

$$AX + XB = C, \tag{4}$$

where $A = W + iT \in \mathbb{C}^{m\times m}$ and $B = U + iV \in \mathbb{C}^{n\times n}$, W, T, U and V are real symmetric positive semi-definite matrices and $i = \sqrt{-1}$. Using the Kronecker sum, it is easy to prove that Eq. (4) has a unique solution [2] when there is no common eigenvalues of A and $-B$. Eq. (4) is equivalent to the linear system

$$Zx = c, \tag{5}$$

where $Z = I_n \otimes A + B^T \otimes I_m,\ c = \mathrm{vec}(C)$ and $x = \mathrm{vec}(X)$, where $\otimes$ is Kronecker product, I_n is identity matrix of dimension $n \times n$ and for any matrix $A = (a_1, ..., a_n)$ with the columns a_k, vec(A) is an operator such that $\mathrm{vec}(A) = (a_1^T, ..., a_n^T)^T \in \mathbb{C}^{mn}$. Obtaining the solution of equation (4), by solving the linear system (5) is not a suitable method and it can be costly, because the dimension of the linear system (5) may be very large. Eq. (4) can be solved by direct methods such as Bartels-Stewart [1] and the Hessenberg-Schur methods [13]. But for efficiently solving the Sylvester matrix Eq. (4), iterative methods can be used.

In [2] Bai proposed HSS approach for solving Eq. (4). Authors of [29] improved the method of Bai [2] by presenting the MHSS iterative method for the Sylvester equation. Authors of [12] applied PMHSS approach for solving Eq. (4). Salkuyeh and Bastani [23] introduced two-parameter generalized Hermitian and skew-Hermitian splitting (TGHSS) iteration method. Dehghan and Shirilord [10] by parameterizing the MHSS method presented a generalized MHSS (GMHSS) iteration method. In fact for different parameters in GMHSS iteration method, different methods are formed. Authors of [10] proved that there exists at least one region $\Omega \in \mathbb{R}^2$ that GMHSS iteration method is convergent. In the following, we will briey describe the CRI iterative method for finding solution of the linear systems. To do this first consider the problem

$$Zx = b, \tag{6}$$

where $Z \in \mathbb{C}^{n\times n}$ and $x, b \in \mathbb{C}^n$. Let Z be complex symmetric matrix of the form $Z = F + iG$, where $F, G \in \mathbb{R}^n$ are real, symmetric, and with F and G are positive semi-defnite matrices. CRI method [28] can be expressed as follows.

1.1. *Combination of real and imaginary parts (CRI) method [28]*

For a given initial approximation $x_{(0)} \in \mathbb{C}^n$, we obtain next iterate $x_{(j+1)}$ from:

$$\begin{cases}(\alpha G + F)x_{(j+1/2)} = (\alpha - i)Gx_{(j)} + b,\\ (\alpha F + G)x_{(j+1)} = (\alpha + i)Fx_{(j+1/2)} - ib, \qquad j = 0, 1, 2, ...,\end{cases} \tag{7}$$

where $\alpha > 0$.
Suppose the matrices F and G are semi-positive definite, so before introducing the convergence theorem of the new method, we should pay attention to useful information about these matrices. To do so first recall the following lemma [28].

Lemma 1.1. *Let $F \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{n\times n}$ be symmetric positive semi-definite matrices satisfying* $\mathrm{null}(F) \cap \mathrm{null}(G) = \{0\}$, *where* $\mathrm{null}(G)$ *denotes null space of any matrix G. Then there exists a nonsingular matrix $P \in \mathbb{R}^{n\times n}$ such that*

$$F = P^T D_F P, \qquad G = P^T D_G P,$$

where $D_F = Diag(\mu_1, ..., \mu_n)$, $D_G = Diag(\lambda_1, ..., \lambda_n)$, λ_l and μ_l satisfy

$$\mu_l + \lambda_l = 1, \qquad \lambda_l \geq 0,\ \mu_l \geq 0,\ l = 1, ..., n.$$

In the next section, we will apply CRI method (7) for solving large sparse complex Sylvester Eq. (4).

2. The use of CRI method for solving Sylvester matrix equations

We see that the CRI iterative method [28] was derived for finding solution of the linear systems of the form

$$Zx = b \ \equiv (W + iT)x = b,$$

where W and T are symmetric positive semi-defnite matrices. Our original scientific contribution in this paper is to extend CRI iterative method for solving Sylvester matrix equation of the form:

$$AX + XB = C \ \equiv (W + iT)X + X(U + iV) = C,$$

where W, T, U and V are real symmetric positive semi-definite matrices. For doing this, first we write Eq. (4) as

$$WX + XU = -iXV - iTX + C. \tag{8}$$

Assume that $\alpha > 0$ is an arbitrary number. Then, adding αTX and αXV to both sides of the above relation yields

$$(\alpha T + W)X + X(\alpha V + U) = (\alpha - i)[TX + XV] + C. \tag{9}$$

On the other hand multiplying both sides of (8) by $-i$ and then, adding αWX and αXU to both sides of it yield:

$$(\alpha W + T)X + X(\alpha U + V) = (\alpha + i)[WX + XU] - iC. \tag{10}$$

Now by considering relations (9) and (10) we obtain the following method for solving Eq. (4).

2.1. The CRI procedure for solving Sylvester matrix Eq. (4)

Compute $X_{(k+1)} \in \mathbb{C}^{m\times n}$ for $k = 0, 1, 2, ...$ by using the following procedure such that $\{X_{(k)}\}_{k=0}^{\infty} \subseteq \mathbb{C}^{m\times n}$ converges:

$$\begin{cases}(\alpha T + W)X_{(k+\frac{1}{2})} + X_{(k+\frac{1}{2})}(\alpha V + U) = (\alpha - i)[TX_{(k)} + X_{(k)}V] + C,\\ (\alpha W + T)X_{(k+1)} + X_{(k+1)}(\alpha U + V) = (\alpha + i)[WX_{(k+\frac{1}{2})} + X_{(k+\frac{1}{2})}U] - iC,\end{cases} \tag{11}$$

where $\alpha > 0$ is constant and $X_{(0)} \in \mathbb{C}^{m\times n}$ is an initial guess. It is clear that the matrices $\alpha W + T, \alpha U + V, \alpha T + W$ and $\alpha V + U$ are symmetric positive defnite, therefore, the two half-steps involved in the CRI iteration can be solved effectively using efficient direct algorithms.

Here we introduce the convergence analysis of new iteration method (11). Based on Theorem 2.1 in [2] and Theorem 2.1 in [28], the following convergence theorem will be obtained for the CRI iteration method for solving Sylvester matrix Eq. (4).

Theorem 2.1. *Let $A = W + iT \in \mathbb{C}^{m\times m}$ and $B = U + iV \in \mathbb{C}^{n\times n}$, where W, T, U and V are real symmetric positive semi-definite matrices and let $\alpha > 0$ be constant. Suppose that Eq.* (4) *has a unique solution. Denote*

$$Q = I_n \otimes W + U \otimes I_m \in \mathbb{R}^{nm\times nm}, \quad R = I_n \otimes T + V \otimes I_m \in \mathbb{R}^{nm\times nm}. \tag{12}$$

Then the iteration matrix of CRI method (11) *is*

$$L(\alpha) = (\alpha^2 + 1)(\alpha Q + R)^{-1}Q(\alpha R + Q)^{-1}R, \tag{13}$$

and the spectral radius of the matrix $L(\alpha)$ satisfies

$$\rho(L(\alpha)) \le \delta(\alpha) \equiv \frac{\alpha^2 + 1}{(\alpha + 1)^2} < 1, \quad \forall \alpha > 0, \tag{14}$$

then the CRI iteration (11) *converges unconditionally to the unique exact solution $X_* \in \mathbb{C}^{m\times n}$ of Eq.* (4) *for any initial guess $X_{(0)}$.*

Proof. By using Kronecker product, we can write scheme (11) in the following form:

$$\begin{cases}[I_n \otimes (\alpha T + W) + (\alpha V + U)^T \otimes I_m]X_{(k+\frac{1}{2})} = (\alpha - i)[I_n \otimes T + V^T \otimes I_m]X_{(k)} + c,\\ [I_n \otimes (\alpha W + T) + (\alpha U + V)^T \otimes I_m]X_{(k+1)} = (\alpha + i)[I_n \otimes W + U^T \otimes I_m]X_{(k+\frac{1}{2})} - ic,\end{cases} \tag{15}$$

where $c = \mathrm{vec}(C)$ and $x = \mathrm{vec}(X)$. Note that:

$$I_n \otimes (\alpha T + W) + (\alpha V + U)^T \otimes I_m = \alpha(I_n \otimes T + V \otimes I_m) + (I_n \otimes W + U \otimes I_m) = \alpha R + Q,$$

and

$$I_n \otimes (\alpha W + T) + (\alpha U + V)^T \otimes I_m = \alpha(I_n \otimes W + U \otimes I_m) + (I_n \otimes T + V \otimes I_m) = \alpha Q + R,$$

where R and Q are defined in (12). Then Eq. (15) can be rewritten as

$$\begin{cases}(\alpha R + Q)x_{(k+\frac{1}{2})} = (\alpha - i)Rx_{(k)} + c,\\ (\alpha Q + R)x_{(k+1)} = (\alpha + i)Qx_{(k+\frac{1}{2})} - ic.\end{cases} \tag{16}$$

It is clear that, the scheme (16) is the CRI method [28] for solving Eq. (6), with $\mathcal{A} = Q + iR$ and $b = c$. Suppose that $\lambda^{Q}_{p,q}, \lambda^{R}_{p,q}, \lambda^{W}_{p}, \lambda^{T}_{p}, \lambda^{U}_{p}$ and λ^{V}_{q}, $(p = 1, ..., m,\ q = 1, .., n)$ denote the eigenvalues of Q, R, W, T, U and V $(p = 1, ..., m,\ q = 1, .., n)$, respectively. Since

$$\lambda^{Q}_{p,q} = \lambda^{W}_{p} + \lambda^{U}_{q} \geq 0, \quad \lambda^{R}_{p,q} = \lambda^{T}_{p} + \lambda^{V}_{q} \geq 0, \quad p = 1, ..., m,\ q = 1, ..., n,$$

then Q and R are symmetric positive semi-definite matrices. On the other hand we assume that Eq. (4) has a unique solution, therefore the matrix

$$I \otimes A + B^T \otimes I = I \otimes (W + iT) + (U + iV)^T \otimes I = Q + iR = \mathcal{A},$$

is nonsingular, this yields null$(R) \cap$ null$(Q) = \{0\}$, hence according to Lemma 1.1, there exists a nonsingular matrix $P \in \mathbb{R}^{nm\times nm}$

$$Q = P^T D_Q P, \qquad Q = P^T D_Q P, \tag{17}$$

where $D_Q = \mathrm{Diag}(\mu_1, ..., \mu_{nm})$ and $D_R = \mathrm{Diag}(\lambda_1, ..., \lambda_{nm})$, λ and μ satisfy

$$\mu_l + \lambda_l = 1, \quad \lambda_l \geq 0,\ \mu_l \geq 0,\ l = 1, ..., nm.$$

Removing $x_{(k+\frac{1}{2})}$ from (16) gives $x_{(k+\frac{1}{2})} = L(\alpha)X_k + K(\alpha)c$, where $L(\alpha)$ is iteration matrix for new method (11) and is defined in (13) and $K(\alpha) = \frac{1}{\alpha}(\alpha R + Q)(Q - iR)^{-1}(\alpha Q + R)$. We know that CRI procedure (11) is convergent if $\rho(L(\alpha)) < 1$. But

$$\begin{aligned}\rho(L(\alpha)) &= (\alpha^2 + 1)\rho((\alpha P^T D_Q P + P^T D_R P)^{-1} P^T D_Q P (\alpha P^T D_R P + P^T D_Q P)^{-1} P^T D_R P)\\ &= (\alpha^2 + 1)\rho((\alpha D_Q + D_R)^{-1} D_Q (\alpha D_R + D_Q)^{-1} D_R)\\ &= (\alpha^2 + 1)\max_{\mu_l,\lambda_l}\left\{\frac{\mu_l\lambda_l}{(\alpha\mu_l + \lambda_l)(\alpha\lambda_l + \mu_l)}\right\}\\ &= (\alpha^2 + 1)\max_{\mu_l,\lambda_l}\left\{\frac{\mu_l\lambda_l}{(\alpha^2 + 1)\mu_l\lambda_l + 2\alpha\mu_l\lambda_l}\right\} \leq \frac{\alpha^2 + 1}{(\alpha + 1)^2} < 1.\end{aligned}$$

This shows that new iteration (11) converges unconditionally to the unique exact solution $X_* \in \mathbb{C}^{m\times n}$ of Eq. (4). □

3. Numerical Results

In this section, a test problem is given to show the efficiency of CRI method for approximating the solution of Sylvester matrix equation by comparing it with PMHSS, Method (A) (see [21]), Method (B) (see [3]) and GlGMRES(5) (GlGMRES(10) and GlGMRES(15)) method [4]. The numerical experiments are performed in Matlab on an Intel (R) Pentium (R) CPU N3700 or (1.60 GHz, 4 GB RAM). In our test performed, we used $X_0 = 0$ (zero matrix) for the initial guess and the stopping criteria for outer iterations is

$$\| C - AX_{(k)} - X_{(k)}B \|_F \ / \ \| C \|_F \leq 5 \times 10^{-8}.$$

Also we set

$$\mathrm{E}(k) = \log_{10} \| C - AX_{(k)} - X_{(k)}B \|_F \ .$$

Example 3.1. *We want to determine the approximate solution of the Sylvester matrix equation*

$$[(K + \sigma_1 I_n) + i\sigma_2 I_n]X + X[(K + \sigma_1 I_n) + i\sigma_2 I_n] = C, \tag{18}$$

where the matrix K is of the form $K = I_m \otimes V_m + V_m \otimes I_m$, *with* $V_m = h^{-2}$ *Tridiagonal* $(-1, 2, -1) \in \mathbb{R}^{m\times m}$. *Therefore K is a block tridiagonal matrix of size* $n \times n$, *with* $n = m^2$. *Also* I_n *and* I_m *are identity matrices of the dimensions n and m, respectively. Here* $h = \frac{1}{m+1}$. *We set* $\sigma_1 = 1$ *and* $\sigma_2 = 10$. *Moreover for matrix C, we consider two cases as given in the following:*
Case (a): Consider the matrix C, such that $X_* = (x_{i,j})$ *with*

$$x_{i,j} = \sin(x_i) + \sin(y_j), \qquad i, j = 1, 2, ..., n, \tag{19}$$

can be exact solution of (18), *where* $x_i = -4 + 8(i-1)/(n-1)$ *and* $y_j = -4 + 8(j-1)/(n-1)$, $\quad i, j = 1, 2, ..., n$.
Case (b): Set $C = FG$, *where the matrices* $F \in \mathbb{R}^{n\times 1}$ *and* $G \in \mathbb{R}^{1\times n}$ *have normally distributed random entries.*

Example 3.2. *Consider the equation* $AX + XA = C$, *with*

$$T = I \otimes V + V \otimes I, \qquad and \qquad W = 10(I \otimes V_c + V_c \otimes I) + 9(e_1 e_m^T + e_m e_1^T) \otimes I,$$

where

$$V = \text{Tri}(-1, 2, -1) = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}_{m\times m},$$

and $V_c = V - e_1 e_m^T - e_m e_1^T \in \mathbb{R}^{m\times m}$, $e_1 = (1, 0, 0, ..., 0)^T \in \mathbb{R}^m$, $e_m = (0, 0, ..., 1)^T \in \mathbb{R}^m$. *Therefore, the dimension of the matrices W, T, U and V will be* $n = m^2$. *Also right hand side matrix C is such that:*
Case (a): The matrix $X_* = (x_{i,j})$ *with*

$$x_{i,j} = \exp\left[-\left(x_i^2 + y_j^2\right)\right], \qquad i, j = 1, 2, ..., n, \tag{20}$$

is exact solution of $AX + XA = C$,, *where* $x_i = -1 + 2(i-1)/(n-1)$ *and* $y_j = -1 + 2(j-1)/(n-1)$, $\quad i, j = 1, 2, ..., n$.
Case (b): Set $C = FG$, *where the matrices* $F \in \mathbb{R}^{n\times 1}$ *and* $G \in \mathbb{R}^{1\times n}$ *have normally distributed random entries.*

Example 3.3. *Consider the Sylvester equation*

$$\left(-\beta I_n + iU^T(-\text{Diag}(1, \frac{1}{2}, ..., \frac{1}{n}) + e_1 e_n^T)U\right) X + X \left(-\beta I_n + iV^T(-\text{Diag}(1, \frac{1}{2}, ..., \frac{1}{n}) + e_1 e_n^T)V\right) = FG,$$

where I_n *is* $n \times n$ *identity matrix,* $U, V \in \mathbb{R}^{n\times n}$ *are the orthogonal factors of the QR decomposition of random* $n \times n$ *matrices,* $e_1 = (1, 0, ..., 0)^T \in \mathbb{R}^n$, $e_n = (0, 0, ..., 0, 1)^T \in \mathbb{R}^n$ *and* β *is constant. The matrices* $F \in \mathbb{R}^{n\times 1}$ *and* $G \in \mathbb{R}^{1\times n}$ *have normally distributed random entries. The optimal parameters of the both CRI and PMHSS methods used here are included in Tables 1 and 2. Note that these parameters are experimentally determined by minimizing the number of iterations. All numerical result for cases (a) and (b) are listed in these tables. By results of Table 1, we see that the number of iterations for these methods has not changed much with increasing dimension of the problem, which indicates that these methods are efficient versus increasing dimension of the problem. The logarithm of the residual error versus iteration number for PMHSS, CRI, Methods (A) and (B) is plotted in Figs. 1, 2 and 3. The outcome of this graph is that, the CRI method is faster than the PMHSS method. Also we see that Method (A) and Method (B) can not solve this test for case (b). The logarithm of the residual error versus iteration number for Example 3.3 PMHSS, CRI, Method (A) and Method (B) is plotted in Figs 4. From this figure CRI method is faster than Methods (A) and (B) when the parameter* β *is large (see the case* $\beta = 100$.*) Figs. 7 and 8 show approximate solutions for imaginary and real parts for Example 3.1 (case (a)) and Example 3.1 (case (a)). The dispersion of the eigenvalues of iteration matrices* *is plotted in Fig. 5. According to this figure, the modulus of the eigenvalues of the iteration matrix of PMHSS method is of a large size, which it influences modulus of the spectral radius of iteration matrix for this method. In contrast, the iteration matrix CRI method has small real part (in absolute terms) and therefore, the spectral radius of the iteration matrix will be small, which results rapid convergence of this method. Also according to Table 1, CRI method* (11) *is more efficient than PMHSS method, because it requires fewer number of iterations and less CPU time which shows the fast convergence of new method. In general from tables and figures that are ready in this section we conclude new CRI scheme is an efficient method for solving complex Sylvester matrix equation.*

Table 1: The comparison of iteration number (IT), logarithm of the residual error (E(.)) and CPU time for Example 3.1; case (a).

PMHSS Method[12]			
$n \times n$	64×64	100×100	400×400
α_{opt}	1	1	1
Iteration	46	47	48
CPU time(s)	1.03	3.17	249.75
E(.)	4.4×10^{-5}	8.2×10^{-5}	9.3×10^{-4}
GlGMRES(5) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	7	10	32
CPU time(s)	0.44	1.12	108.49
E(.)	3.8×10^{-4}	2.6×10^{-4}	4.7×10^{-3}
GlGMRES(10) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	3	4	10
CPU time(s)	0.41	1.21	89.91
E(.)	3.7×10^{-6}	9.0×10^{-5}	1.7×10^{-3}
GlGMRES(15) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	1	2	5
CPU time(s)	0.35	1.45	91.52
E(.)	3.4×10^{-5}	2.7×10^{-5}	6.8×10^{-3}
CRI Method (11)			
$n \times n$	64×64	100×100	400×400
α_{opt}	0.85	0.85	0.85
Iteration	15	14	12
CPU time(s)	0.32	0.86	62.15
E(.)	7.8×10^{-6}	4.0×10^{-5}	1.8×10^{-4}

Table 2: The comparison of iteration number (IT), logarithm of the residual error (E(.)) and CPU time for Example 3.1; case (b).

PMHSS Method[12]			
$n \times n$	64×64	100×100	400×400
α_{opt}	1	1	1
Iteration	43	44	46
CPU time(s)	0.85	2.25	203.60
E(.)	8.1×10^{-7}	1.4×10^{-6}	6.8×10^{-6}
GlGMRES(5) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	12	17	54
CPU time(s)	0.70	2.37	182.08
E(.)	1.0×10^{-6}	1.0×10^{-6}	5.4×10^{-6}
GlGMRES(10) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	4	6	16
CPU time(s)	0.51	1.84	142.27
E(.)	1.2×10^{-7}	2.7×10^{-7}	5.3×10^{-6}
GlGMRES(15) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	2	3	9
CPU time(s)	0.54	2.01	152.15
E(.)	2.8×10^{-7}	1.7×10^{-7}	2.6×10^{-6}
CRI Method (11)			
$n \times n$	64×64	100×100	400×400
α_{opt}	1.1	1.1	1.1
Iteration	21	20	20
CPU time(s)	0.38	0.94	85.15
E(.)	4.7×10^{-7}	1.5×10^{-6}	6.4×10^{-6}

Table 3: The comparison of iteration number (IT), logarithm of the residual error (E(.)) and CPU time for Example 3.2, case (a).

PMHSS Method[12]			
$n \times n$	64×64	100×100	400×400
α_{opt}	0.65	0.69	0.70
Iteration	32	32	31
CPU time(s)	0.2561	0.6684	52.5192
E(.)	9.0×10^{-6}	5.1×10^{-5}	8.9×10^{-5}
GlGMRES(5) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	18	29	71
CPU time(s)	0.79	2.96	235.81
E(.)	6.8×10^{-6}	1.0×10^{-5}	1.9×10^{-4}
GlGMRES(10) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	5	8	22
CPU time(s)	0.57	2.14	188.66
E(.)	1.0×10^{-6}	3.7×10^{-6}	1.1×10^{-5}
GlGMRES(15) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	3	4	11
CPU time(s)	0.68	2.41	184.03
E(.)	2.4×10^{-6}	1.2×10^{-5}	1.0×10^{-5}
CRI Method (11)			
$n \times n$	64×64	100×100	400×400
α_{opt}	1	1	1
Iteration	16	17	20
CPU time(s)	0.1262	0.4431	31.3413
E(.)	3.6×10^{-5}	6.1×10^{-6}	8.3×10^{-5}

Table 4: The comparison of iteration number (IT), logarithm of the residual error (E(.)) and CPU time for Example 3.2, case (b).

PMHSS Method[12]			
$n \times n$	64×64	100×100	400×400
α_{opt}	1	1	1
Iteration	39	39	40
CPU time(s)	0.79	2.11	177.21
E(.)	6.6×10^{-7}	1.3×10^{-6}	4.3×10^{-6}
GlGMRES(5) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	24	33	71
CPU time(s)	1.10	3.63	238.65
E(.)	6.7×10^{-7}	1.4×10^{-6}	6.0×10^{-4}
GlGMRES(10) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	7	10	27
CPU time(s)	0.83	2.82	232.13
E(.)	3.2×10^{-7}	5.8×10^{-7}	6.4×10^{-6}
GlGMRES(15) [4]			
$n \times n$	64×64	100×100	400×400
Iteration	4	4	15
CPU time(s)	0.92	2.48	247.63
E(.)	6.5×10^{-8}	1.2×10^{-6}	3.0×10^{-6}
CRI Method (11)			
$n \times n$	64×64	100×100	400×400
α_{opt}	1	1	1
Iteration	19	20	25
CPU time(s)	0.38	1.01	108.58
E(.)	6.7×10^{-7}	1.5×10^{-6}	3.5×10^{-6}

Table 5: The comparison of iteration number (IT), logarithm of the residual error (E(.)) and CPU time for Example 3.3 by $\beta = 1$.

Method (A) [21]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	5	5	5	5
CPU time(s)	0.42	1.78	5.50	11.31
E(.)	9.4×10^{-11}	1.3×10^{-10}	1.5×10^{-10}	3.8×10^{-10}
Method (B) [3]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	3	3	3	3
CPU time(s)	0.44	1.75	4.73	10.26
E(.)	9.5×10^{-11}	1.3×10^{-10}	1.5×10^{-10}	3.8×10^{-10}
GlGMRES(5) [4]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	3	3	3	3
CPU time(s)	0.51	1.72	4.70	11.14
E(.)	3.0×10^{-7}	2.9×10^{-7}	4.0×10^{-7}	7.7×10^{-7}
PMHSS Method[12]				
$n \times n$	100×100	200×200	300×300	400×400
α_{opt}	1	1	1	1
Iteration	46	47	48	48
CPU time(s)	2.83	20.04	95.46	227.84
E(.)	3.4×10^{-6}	8.0×10^{-6}	9.4×10^{-6}	1.4×10^{-5}
CRI Method (11)				
$n \times n$	100×100	200×200	300×300	400×400
α_{opt}	1.1	1.1	1.1	1.1
Iteration	20	19	19	19
CPU time(s)	1.82	11.06	43.28	104.85
E(.)	2.3×10^{-6}	5.8×10^{-6}	5.6×10^{-6}	1.3×10^{-5}

Table 6: The comparison of iteration number (IT), logarithm of the residual error (E(.)) and CPU time for Example 3.3 by $\beta = 10$.

Method (A) [21]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	7	7	7	7
CPU time(s)	0.63	2.59	7.54	16.82
E(.)	1.2×10^{-9}	2.4×10^{-9}	3.3×10^{-9}	6.1×10^{-9}
Method (B) [3]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	5	5	5	5
CPU time(s)	0.72	2.64	7.48	15.86
E(.)	1.3×10^{-9}	2.1×10^{-9}	2.1×10^{-9}	4.5×10^{-9}
GlGMRES(5) [4]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	2	2	2	2
CPU time(s)	0.33	0.89	2.43	5.46
E(.)	2.3×10^{-12}	2.1×10^{-12}	3.0×10^{-11}	1.9×10^{-11}
PMHSS Method[12]				
$n \times n$	100×100	200×200	300×300	400×400
α_{opt}	1	1	1	1
Iteration	48	49	49	49
CPU time(s)	2.99	20.43	97.05	236.77
E(.)	4.3×10^{-6}	6.6×10^{-6}	4.2×10^{-6}	6.1×10^{-5}
CRI Method (11)				
$n \times n$	100×100	200×200	300×300	400×400
α_{opt}	1.1	1.1	1.1	1.1
Iteration	8	7	7	7
CPU time(s)	0.71	3.96	16.08	37.18
E(.)	8.4×10^{-7}	4.4×10^{-6}	5.3×10^{-6}	7.9×10^{-6}

Table 7: The comparison of iteration number (IT), logarithm of the residual error (E(.)) and CPU time for Example 3.3 by $\beta = 100$.

Method (A) [21]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	10	10	10	10
CPU time(s)	0.81	3.68	10.28	22.64
E(.)	6.3×10^{-7}	1.0×10^{-7}	4.5×10^{-7}	1.3×10^{-7}
Method (B) [3]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	8	8	8	8
CPU time(s)	0.83	3.64	10.02	21.63
E(.)	7.3×10^{-7}	1.3×10^{-7}	2.0×10^{-7}	6.4×10^{-7}
GlGMRES(5) [4]				
$n \times n$	100×100	200×200	300×300	400×400
Iteration	2	2	2	2
CPU time(s)	0.18	0.93	2.29	5.23
E(.)	1.4×10^{-13}	2.0×10^{-11}	9.4×10^{-11}	1.5×10^{-11}
PMHSS Method[12]				
$n \times n$	100×100	200×200	300×300	400×400
α_{opt}	1	1	1	1
Iteration	49	49	49	49
CPU time(s)	2.92	20.42	94.90	232.02
E(.)	3.4×10^{-6}	1.6×10^{-5}	1.4×10^{-5}	3.8×10^{-5}
CRI Method (11)				
$n \times n$	100×100	200×200	300×300	400×400
α_{opt}	1.1	1.1	1.1	1.1
Iteration	4	3	4	4
CPU time(s)	0.34	1.57	8.55	20.38
E(.)	7.5×10^{-7}	8.6×10^{-7}	4.2×10^{-7}	$\times 10^{-7}$

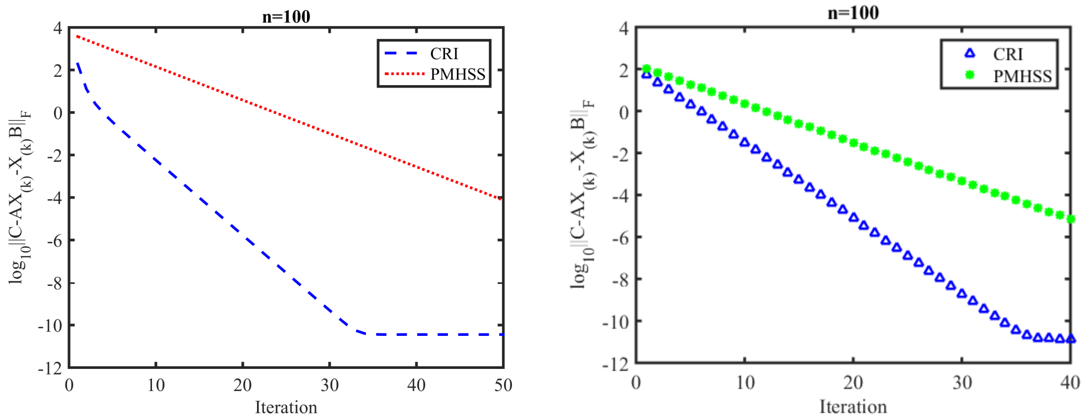

Figure 1: The logarithm of the residual error versus iteration number; Left: Example 3.1 (case (a)); Right: Example 3.2 (case (a)).

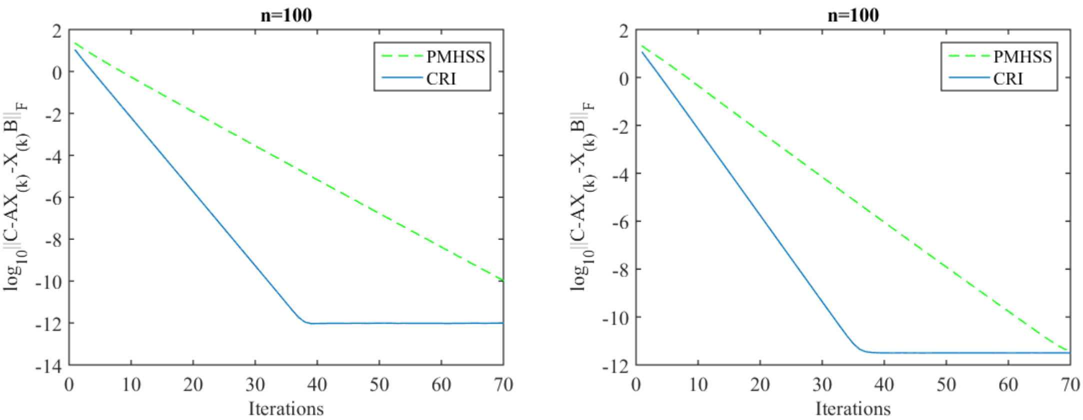

Figure 2: The logarithm of the residual error versus iteration number; Left: Example 3.1 (case (b)); Right: Example 3.2 (case (b)).

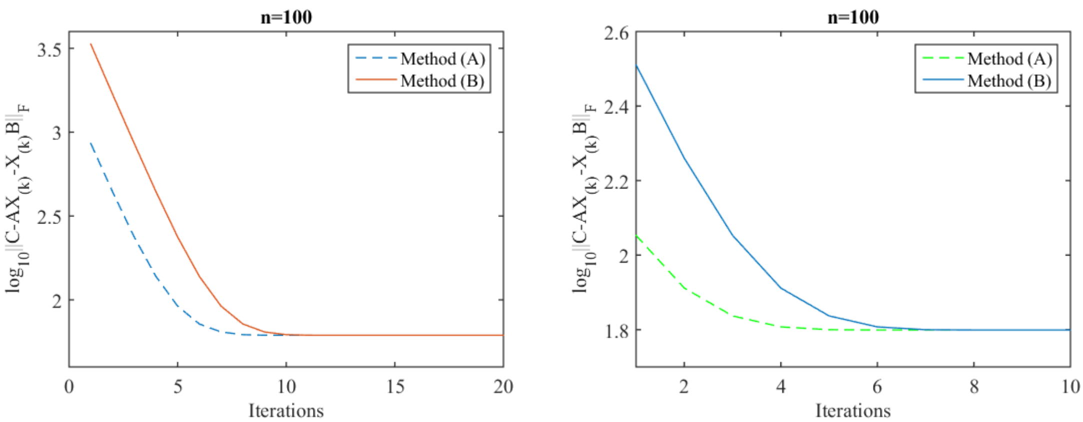

Figure 3: The logarithm of the residual error versus iteration number; Left: Example 3.1 (case (b)); Right: Example 3.2 (case (b)).

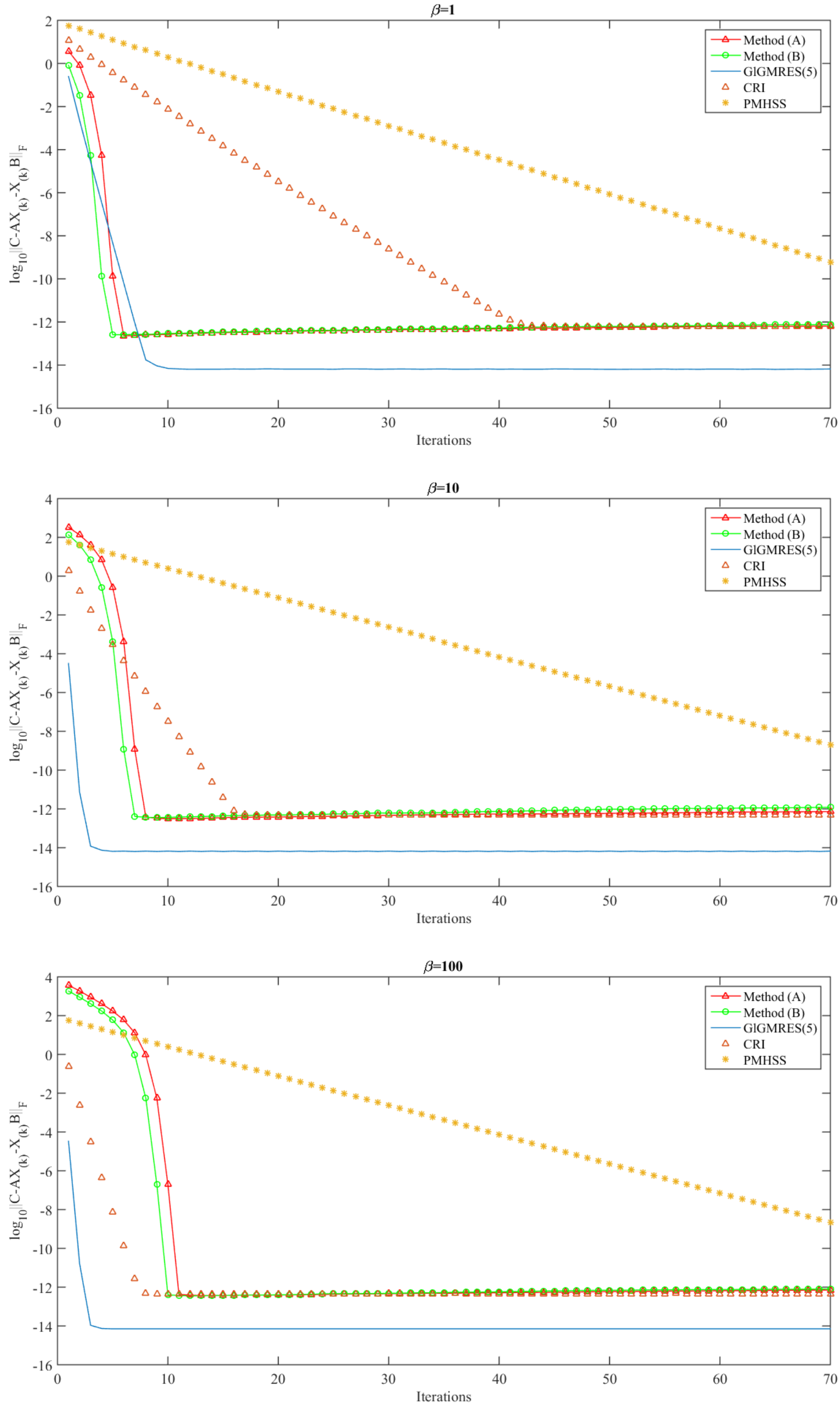

Figure 4: The logarithm of the residual error versus iteration number for Example 3.3 for different parameter β.

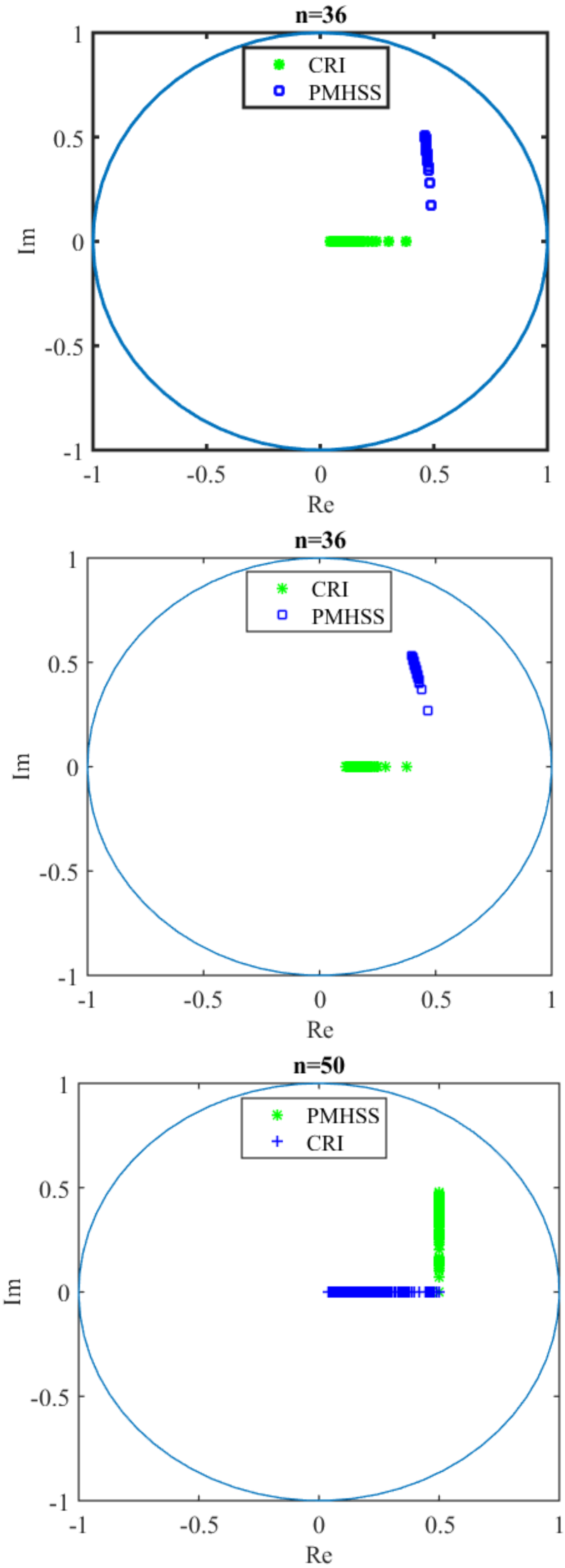

Figure 5: The eigenvalue distribution of the iteration matrices; Up: Example 3.1; Middle: Example 3.2; Bottom: Example 3.3 (for $\beta = 1$).

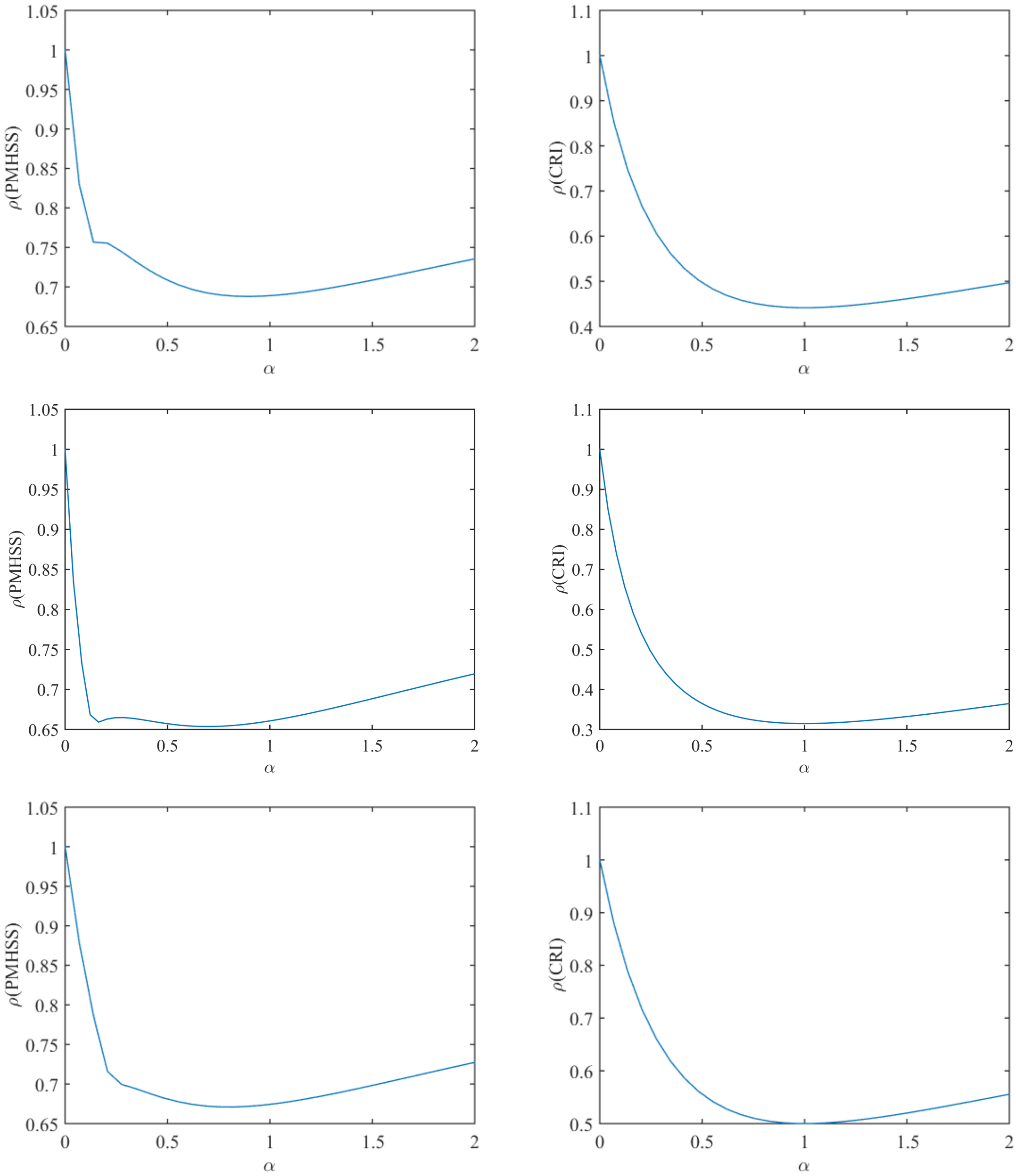

Figure 6: The almost locations of the optimal parameters for PMHSS and CRI methods; Up: Example 3.1; Middle: Example 3.2; Bottom: Example 3.3 (for $\beta = 1$).

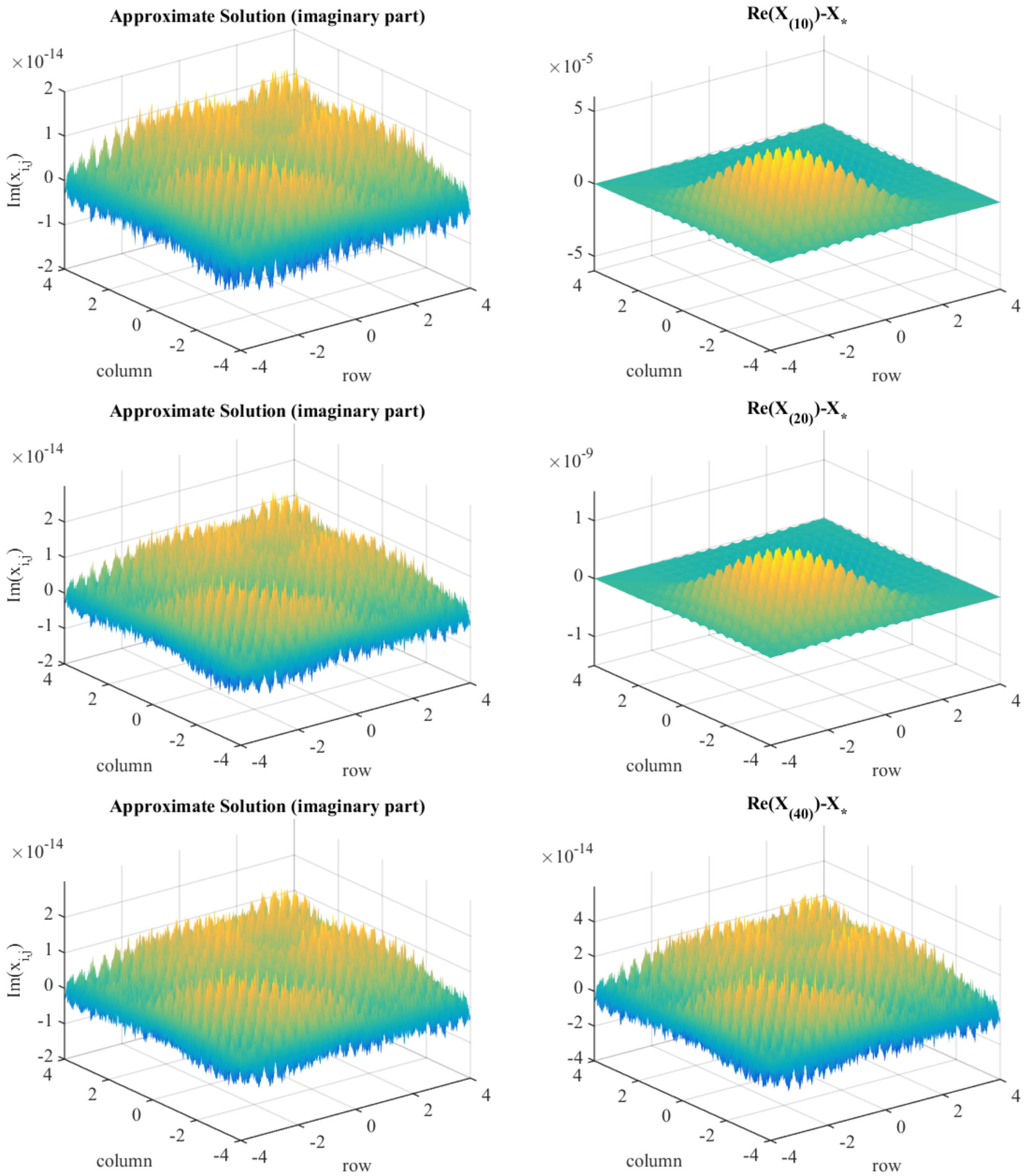

Figure 7: Approximate solutions for imaginary and real parts for Example 3.1 (case (a)); Top (after 10 iterations); Middle (after 20 iterations); Bottom (after 40 iterations).

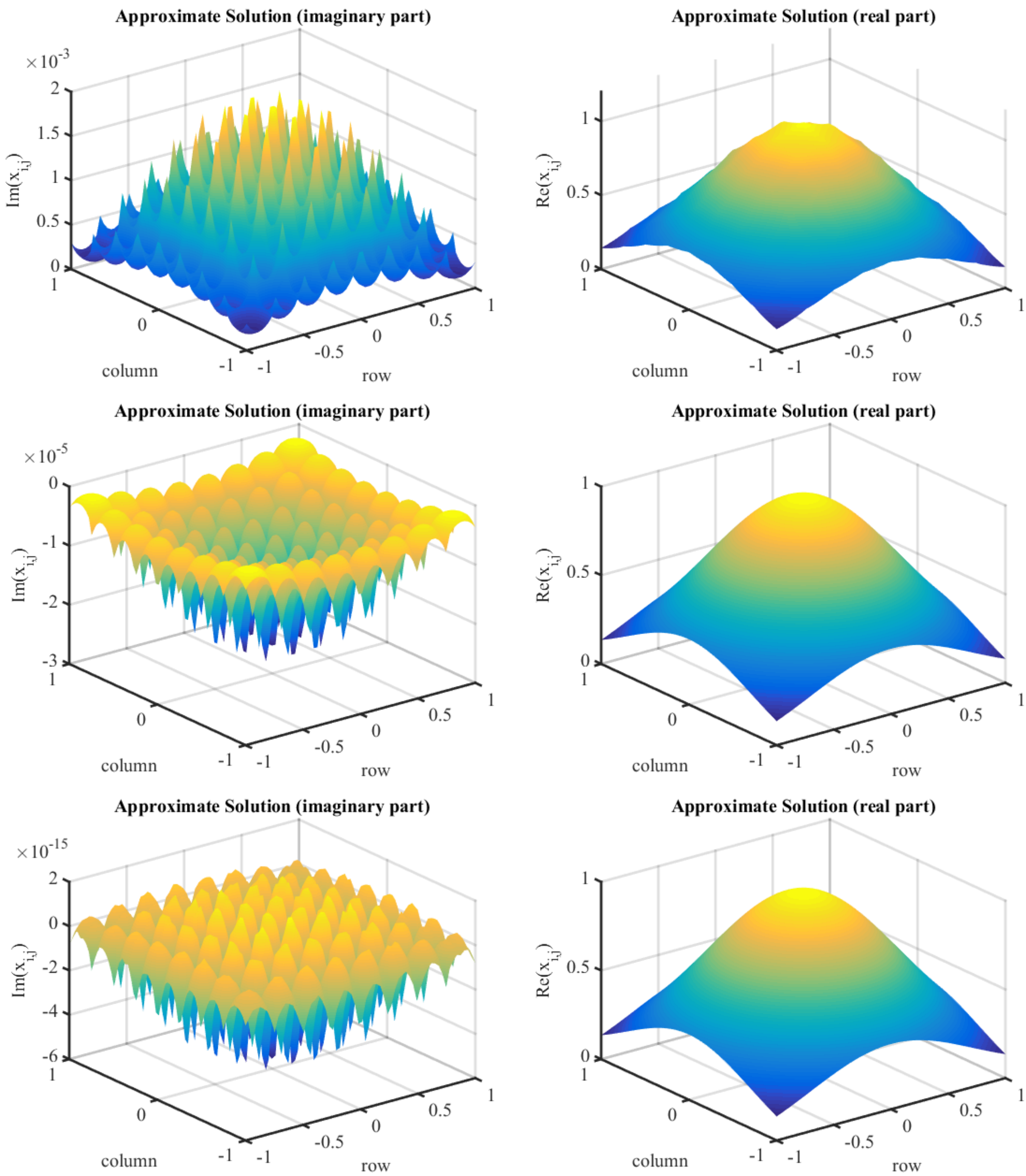

Figure 8: Approximate solutions for imaginary and real parts for Example 3.2 (case (a)); Top (after 3 iterations); Middle (after 10 iterations); Bottom (after 30 iterations).

Acknowledgments:
The authors extend their appreciation to reviewers for their valuable suggestions to revise this paper.

References

[1] R.H. Bartels, G.W. Stewart, Solution of the matrix equation $AX + XB = C$: Algorithm 432, Communications of the ACM 15 (1972) 820–826.
[2] Z.-Z. Bai, On Hermitian and skew-Hermitian spliting iteration methods for continuous Sylvester equations, Journal of Computational Mathematics 29 (2011) 185–198.
[3] P. Benner, Factorized solutions of Sylvester equations with applications in control, in Proc. of the 16th International Symposium on Mathematical Theory of Network and Systems (MTNS 2004), 2004.
[4] A. Bouhamidi, K. Jbilou, A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications, Applied Mathematics and Computation 206 (2008) 687–694.
[5] R. Byers, Solving the algebraic Ricatti equation with the matrix sign function, Linear Algebra and its Applications 85 (1987) 267–279.
[6] M. Dehghan, M. Hajarian, Efficient iterative method for solving the second-order Sylvester matrix equation $EVF^2 - AVF - CV = BW$, IET Control Theory & Applications 3 (2009) 1401–1408.
[7] M. Dehghan, M. Hajarian, Analysis of an iterative algorithm to solve the generalized coupled Sylvester matrix equations, Applied Mathematical Modelling 35 (2011) 3285–3300.
[8] M. Dehghan, M. Hajarian, Two iterative algorithms for solving coupled matrix equations over reflexive and anti-reflexive matrices, Computational and Applied Mathematics 31 (2012) 353–371.
[9] M. Dehghan, A. Shirilord, Solving complex Sylvester matrix equation by accelerated double- step scale splitting (ADSS) method, Engineering with Computers 37 (2021) 489–508.
[10] M. Dehghan, A. Shirilord, A generalized modified Hermitian and skew-Hermitian splitting (GMHSS) method for solving complex Sylvester matrix equation, Applied Mathematics and Computation 348 (2019) 632–651.
[11] B. D. Djordjević and N. Č. Dinčić, Classification and approximation of solutions to Sylvester matrix equation, Filomat 33 (2019) 4261–4280
[12] Y. Dong, C. Gu, On PMHSS iteration methods for Sylvester equations, Journal of Computational Mathematics 35 (2017) 600–619.
[13] G.H. Golub, S.G. Nash, C.F. Van Loan, A Hessenberg-Schur method for the problem $AX+XB = C$, IEEE Transactions on Automatic Control 24 (1979) 909–913.
[14] Z.H. He, The general ϕ-Hermitian solution to mixed pairs of quaternion matrix Sylvester equations, Electronic Journal of Linear Algebra 32 (2017) 475–499.
[15] Z.H. He, Q.W. Wang, Y. Zhang, A system of quaternary coupled Sylvester-type real quaternion matrix equations, Automatica 87 (2018) 25–31.
[16] Q. Hu, D. Cheng, The polynomial solution to the Sylvester matrix equation, Applied Mathematics Letters 19 (2006) 859–864.
[17] D. Y. Hu, L. Reichel, Krylov-subspace methods for the Sylvester equation, Linear Algebra and its Applications 172 (1992) 283–313.
[18] A.P. Liao and Z. Z. Bai, Least-squares solutions of the matrix equation $A^TXA = D$ in bisymmetric matrix set, Mathematica Numerica Sinica-chinese Edition 24 (2002) 9–20.
[19] A.P. Liao, Z. Z. Bai and Y. Lei, Best approximate solution of matrix equation $AXB + CYD = E$, SIAM Journal on Matrix Analysis and Applications 27 (2005) 675–688.
[20] Q. Niu, X. Wang, L.-Z. Lu, A relaxed gradient based algorithm for solving Sylvester equations, Asian Journal of Control 13 (2011) 461–464.
[21] J. D. Roberts, Linear model reduction and solution of the algebraic Ricatti equation by use of the sign function, International Journal of Control 32 (1980) 677–687.
[22] R.A. Smith, Matrix equation $XA + BX = C$, SIAM Journal on Applied Mathematics 16 (1968) 198–201.
[23] D.K. Salkuyeh, M. Bastani, A new generalization of the Hermitian and skew-Hermitian splitting method for solving the continuous Sylvester equation, Transactions of the Institute of Measurement and Control 40 (2018) 303–317.
[24] J. J. Sylvester, Sur l'equation en matrices $px = xq$, C. R. Acad. Sci. Paris, 99 (1884) 67–71 and 115–116.
[25] A. Van der Schaft, L2-Gain and Passivity Techniques in Nonlinear Control, 2nd Edition, Springer-Verlag, London, 2000.
[26] X. Wang, Y. Li, L. Dai, On Hermitian and skew-Hermitian splitting iteration methods for the linear matrix equation $AXB = C$, Computers & Mathematics with Applications 65 (2013) 657–664.
[27] Q.W. Wang, C.K. Li, Ranks and the least-norm of the general solution to a system of quaternion matrix equations, Linear Algebra and its Applications 430 (2009) 1626–1640.
[28] T. Wang, Q. Zheng, L. Lu, A new iteration method for a class of complex symmetric linear systems, Journal of Computational and Applied Mathematics 325 (2017) 188–197.
[29] D. Zhou, G. Chen, Q. Cai, On modified HSS iteration methods for continuous Sylvester equation, Applied Mathematics and Computation 263 (2015) 84–93.