A NOVEL ITERATIVE METHOD FOR SOLVING THE COUPLED SYLVESTER-CONJUGATE MATRIX EQUATIONS AND ITS APPLICATION IN ANTILINEAR SYSTEM

Wenli Wang\textsuperscript{1} and Caiqin Song\textsuperscript{1,2,†}

Abstract This paper is devoted to constructing a modified relaxed gradient based iterative (MRGI) algorithm to solve the coupled Sylvester-conjugate matrix equations (CSCMEs) based on the hierarchical identification principle. Convergence analysis shows that the proposed algorithm is effective for arbitrary initial matrices. Further, we apply the MRGI algorithm to study a more general coupled Sylvester conjugate matrix equations and give a sufficient condition to guarantee that the iterative solution converges to the exact solution. Two numerical experiments are provided to demonstrate that the MRGI algorithm has better efficiency and accuracy than the three existing algorithms, which are presented by Wu et al. (2010) and Huang and Ma (2018). Finally, we derive an application of MRGI algorithm in discrete-time antilinear system.

Keywords Generalized coupled Sylvester conjugate matrix equations, relaxation factor, gradient based iterative algorithm, modified relaxed gradient based iterative algorithm.


1. Introduction

Solving matrix equations is a common research topic in system theory, control theory and stability analysis [7, 12, 16, 28, 29, 33]. For example, the mean-square stability of discrete-time Markovian jump linear system

\[ x(k + 1) = A_{r(k)}x(k), \quad x(0) = x_0, \quad r(0) = r_0, \quad \text{(1.1)} \]

can be determined by solving coupled discrete-time Markovian jump Lyapunov matrix equations [16, 29]:

\[ A_i (\sum_{j=1}^{N} \pi_{ij} P_j) A_i^T - P_i + S_i = 0, \quad i \in [1, N]. \quad \text{(1.2)} \]

One of the most important iterative methods is the gradient based iterative (GI) algorithm. Due to the effectiveness and superiority of GI method, it was enriched by

\textsuperscript{†}The corresponding author. Email: songcaiqin1983@163.com (C. Song)

\textsuperscript{1}School of Mathematical Science, University of Jinan, Jinan 250022, China

\textsuperscript{2}Department of Mathematics and Statistics, University of Nevada, Reno 89503, USA
many researchers to compute the numerical solutions of various matrix equations. For instance, Wu et al. solved the solutions to the coupled Sylvester-conjugate matrix equations \[24\]

\[\sum_{\eta=1}^{p} (A_{\eta}X_{\eta}B_{\eta} + C_{\eta}X_{\eta}D_{\eta}) = F_i, \quad i \in I[1, N], \tag{1.3}\]

the extended Sylvester-conjugate matrix equations \[26\]

\[AXB + C\overline{X}D = F, \tag{1.4}\]

and a class of complex conjugate and transpose matrix equations \[25\]

\[\sum_{l=1}^{s_1} (A_lXB_l) + \sum_{l=1}^{s_2} (C_lXD_l) + \sum_{l=1}^{s_3} (G_lX^TH_l) + \sum_{l=1}^{s_4} (M_lX^HN_l) = F, \tag{1.5}\]

by the GI method. Huang and Ma \[14\] gave a new convergence proof of the GI method for solving Eq.(1.3) and answered the problem proposed by Wu et al. \[24\]. Song et al. \[20\] and Hajarian \[8\] investigated the solutions to the coupled Sylvester transpose matrix equations

\[\sum_{\eta=1}^{p} (A_{\eta}X_{\eta}B_{\eta} + C_{\eta}X_{\eta}^TD_{\eta}) = F_i, \quad i \in I[1, N], \tag{1.6}\]

and general Sylvester discrete-time periodic (GSDTP) matrix equations

\[\sum_{j=1}^{m} (A_{ij}X_{ij}B_{ij} + C_{ij}X_{i+1}D_{ij} + E_{ij}Y_iF_{ij} + G_{ij}Y_{i+1}H_{ij}) = M_i, \quad i = 1, 2, \cdots, \tag{1.7}\]

by the GI method, respectively. Zhang et al. \[30–32\] also extended the GI algorithm to the case where the unknown matrices are conjugate, transpose and conjugate transpose, the case where the unknown matrix is nonlinear, the case where the coefficient matrix is a column (row) reduced-rank matrix. In addition, Wang et al. \[23\] provided the optimal convergence factor of the GI algorithm in order to solve some linear matrix equations. Deghan and Hajarian \[4, 9\] constructed the GI method for solving the constraint solutions to linear matrix equations.

In order to reduce the time cost of computation, it is necessary to improve the efficiency of GI method. Hence, several new iterative methods were researched in recent years. For instance, a relaxed gradient based iterative algorithm \[15, 18, 19\] was established to solve the Sylvester matrix equation, the generalized Sylvester matrix equation and coupled Sylvester matrix equations. Subsequently, a modified gradient based iterative algorithm \[22\] was investigated to obtain the solution of the Sylvester matrix equation. Based on the two iterative methods mentioned above, an accelerated gradient based iterative algorithm \[27\] was studied to solve the Sylvester transpose matrix equation. Besides, there are various iterative methods that aim to calculate the solutions of some linear equations, the readers are suggested to refer to \[1–3, 5, 10–12, 13, 17, 21\] for more detailed information.

Inspired by \[6, 15, 24, 34\], we construct a modified relaxed gradient based iterative (MRGI) algorithm for solving the coupled Sylvester-conjugate matrix equations...
(CSCMEs)

\[
\sum_{j=1}^{p} (A_{ij}X_jB_{ij} + C_{ij}\overline{X_j}D_{ij}) = F_i, \quad i \in I[1, q],
\]

(1.8)

where \( A_{ij}, C_{ij} \in C^{m_i \times l_i}, B_{ij}, D_{ij} \in C^{n_j \times p_j}, F_i \in C^{m_i \times p_i}, i \in I[1, q], j \in I[1, p] \) are the given matrices, and \( X_j \in C^{l_j \times n_j}, j \in I[1, p] \) are the unknown matrices. It should be noted that the MRGI algorithm has not been mentioned in previous studies. Numerical results illustrate that the proposed algorithm has better convergence performance than the gradient based iterative (GI) algorithm [24], the relaxed gradient based iterative (RGI) algorithm [15] and the generalized relaxed gradient based iterative (GRGI) algorithm [15]. Further, we generalize the MRGI algorithm to a more general coupled Sylvester conjugate matrix equations

\[
\sum_{j=1}^{s_{i1}} A_{i1j}X_1B_{i1j} + \sum_{j=1}^{w_{i1}} C_{i1j}\overline{X_1}D_{i1j} + \cdots + \sum_{j=1}^{s_{ip}} A_{ipj}X_pB_{ipj} + \sum_{j=1}^{w_{ip}} C_{ipj}\overline{X_p}D_{ipj} = F_i,
\]

(1.9)

for \( i \in I[1, q] \). In addition, only a sufficient condition is shown to analyze the convergence of the GI algorithm in [24]. In this work, by applying real representation of complex matrix, Kronecker product and vector operator, the necessary and sufficient conditions are determined to guarantee the convergence of the GI algorithm [24]. Meanwhile, the optimal convergence factor of the GI algorithm [24] is gave.

The organization of this work is as follows. Section 2 introduces some important preliminaries and lemmas. Section 3 derives a MRGI algorithm to solve the CSCMEs (1.8). Section 4 shows detailed analysis of the convergence for the MRGI algorithm and the GI algorithm [24]. Then a class of more general coupled Sylvester conjugate matrix equations is considered in Section 5. The numerical results are given to explore the effectiveness, efficiency and accuracy of the MRGI algorithm in Section 6 and an application in antilinear system is given in Section 7.

Notations. For \( A \in C^{n \times n} \), we use \( A^T, \overline{A}, A^H \) and \( tr(A) \) to denote the transpose, the conjugate, the conjugate transpose and the trace of \( A \), respectively. Then \( \sigma_{\text{max}}(A), \sigma_{\text{min}}(A), \lambda_{\text{max}}(A), \lambda_{\text{min}}(A) \) and \( \rho(A) \) represent the maximal singular value, the minimal nonzero singular value, the maximal eigenvalue, the minimal eigenvalue and the spectral radius of the matrix \( A \), respectively. For any integers \( m \) and \( n \) with \( m \leq n \), we denote \( I[m, n] = \{m, m+1, \cdots, n\} \). We use \( A \odot B \) to denote the Kronecker product of two matrices \( A \in C^{n \times m} \) and \( B \in C^{p \times q} \). For a matrix \( X = (x_1, x_2, \cdots, x_n) \in C^{m \times n} \), the vector stretching function \( vec(\cdot) : X \rightarrow vec(X) \) is defined as \( vec(X) = (x_1^T, x_2^T, \cdots, x_n^T)^T \). By combining vector operator with Kronecker product, we get \( vec(AXB) = (B^T \otimes A)vec(X) \). The inner product of two matrices is defined as \( \langle A, B \rangle = tr(A^H B) \) with \( A, B \in C^{n \times m} \). The spectral norm of the matrix \( A \) is denoted by \( \|A\|_2 = \sqrt{\lambda_{\text{max}}(A^H A)} = \sigma_{\text{max}}(A) \) and the Frobenious norm of the matrix \( A \) is denoted by \( \|A\|_F = \sqrt{\langle A, A \rangle} = \sqrt{tr(A^H A)} \). The symbol \( I_n \) represents the identity matrix of size \( n \times n \) and the symbol \( \text{rand}(m), \text{diag}(A), \text{tril}(A), \text{triu}(A), \text{eye}(A) \) are functions in MATLAB.
2. Preliminaries

First, the real representation of complex matrix and its properties are reviewed. The definition of the real representation was first introduced in [17]. Let \( A \in \mathbb{C}^{m \times n} \) be an arbitrary complex matrix, then \( A \) can be uniquely decomposed into the form of \( A_1 + A_2i \) with \( A_1, A_2 \in \mathbb{R}^{m \times n} \). Now we are in a position to define the real representation of a complex matrix \( A \) as

\[
A_\sigma := \begin{bmatrix}
    A_1 & A_2 \\
    A_2 & -A_1
\end{bmatrix} \in \mathbb{R}^{2m \times 2n}.
\]

(2.1)

Then, let

\[
\overline{A}_\sigma := (\overline{A})_\sigma, A^T_\sigma := (A^T)_\sigma, A^H_\sigma := (A^H)_\sigma,
\]

(2.2)

and

\[
P_j := \begin{bmatrix}
    I_j & 0 \\
    0 & -I_j
\end{bmatrix}, Q_j := \begin{bmatrix}
    0 & I_j \\
    -I_j & 0
\end{bmatrix},
\]

(2.3)

where \( I_j \) is the identity matrix of size \( j \times j \). The properties of the real representation of complex matrix are given by the following lemmas, which are given in [17].

**Lemma 2.1** ([17]). The properties of real representation matrices.

1. If \( A, B \in \mathbb{C}^{m \times n}, \alpha \in \mathbb{R} \), then

\[
\begin{align*}
(A + B)_\sigma &= A_\sigma + B_\sigma, \\
(\alpha A)_\sigma &= \alpha A_\sigma, \\
P_m A_\sigma P_n &= (\overline{A})_\sigma;
\end{align*}
\]

2. If \( A \in \mathbb{C}^{m \times n}, B \in \mathbb{C}^{n \times r}, C \in \mathbb{C}^{r \times p} \), then

\[
\begin{align*}
(AB)_\sigma &= A_\sigma P_n B_\sigma = A_\sigma \overline{B}_\sigma P_r, \\
(ABC)_\sigma &= A_\sigma \overline{B}_\sigma C_\sigma, \\
Q_m A_\sigma Q_n &= A_\sigma;
\end{align*}
\]

3. If \( A \in \mathbb{C}^{m \times n} \), then

\[
\begin{align*}
(A^T)_\sigma &= (A_\sigma)^T, \\
(A^H)_\sigma &= P_m (A^T)_\sigma P_m; \\
P_m (A^H)_\sigma &= (P_m A_\sigma)^T, \\
(A^H)_\sigma P_m &= (A_\sigma P_n)^T.
\end{align*}
\]

**Lemma 2.2** ([26]). Given a complex matrix \( A \) with appropriate dimensions, the following relations hold.
A novel iterative method for solving the CSCMEs

\[ \|A\sigma\|_F^2 = 2\|A\|_F^2; \]
\[ \|A\sigma\|_2 = \|A\|_2. \]

**Lemma 2.3** ([34]). For the complex matrix equation \(AXB = F\), if \(A\) is a column-full rank matrix and \(B\) is a row-full rank matrix, then the iterative solution \(X(k)\) generated by the gradient-based iterative algorithm

\[ X(k + 1) = X(k) + \mu A^H(F - AX(k)B)B^H, \]

converges to the exact solution \(X^*\) (that is, \(\lim_{k \to \infty} X(k) = X^*\)) for any initial matrix \(X(0)\) if and only if

\[ 0 \leq \mu \leq \frac{2}{\|A\|_F^2\|B\|_F^2}. \]

Moreover, the best convergence factor \(\mu_0\) is

\[ \mu_0 = \frac{2}{\lambda_{\max}(A^HA)\lambda_{\max}(B^HB) + \lambda_{\min}(A^HA)\lambda_{\min}(B^HB)}. \]

**Lemma 2.4** ([34]). Assume that \(m_i (i = 1, 2, \ldots, n)\) are some given positive scalars. Denote \(m_{\max} = \max_{1 \leq i \leq n} \{m_i\}\) and \(m_{\min} = \min_{1 \leq i \leq n} \{m_i\}\). Then

\[ \min_{0 < \mu < \frac{2}{m_{\max}}} \max_{1 \leq i \leq n} |1 - \mu m_i| = \frac{m_{\max} - m_{\min}}{m_{\max} + m_{\min}}. \]

Moreover, the unique \(\mu_{\text{opt}}\) such that this relation holds is

\[ \mu_{\text{opt}} = \frac{2}{m_{\max} + m_{\min}}. \]

3. The modified relaxed gradient based iterative algorithm

In this section, we present a modified relaxed gradient based iterative (MRGI) method for solving CSCMEs (1.8) based on the hierarchical identification principle.

First, define the intermediate matrices

\[ \Phi_{il} := F_i - \sum_{j=1}^{p} (A_{ij}X_jB_{ij} + C_{ij}X_jD_{ij}) + A_{il}X_lB_{il}, i \in I[1, q], l \in I[1, p], \]

\[ \Omega_{il} := F_i - \sum_{j=1}^{p} (A_{ij}X_jB_{ij} + C_{ij}X_jD_{ij}) + C_{il}X_lD_{il}, i \in I[1, q], l \in I[1, p]. \]

Thus, the CSCMEs (1.8) can be decomposed into the following matrix equations

\[ A_{il}X_lB_{il} = \Phi_{il}, i \in I[1, q], l \in I[1, p], \]

\[ C_{il}X_lD_{il} = \Omega_{il}, i \in I[1, q], l \in I[1, p]. \]
From Lemma 2.3, we can construct the recursive forms as follows,

\begin{equation}
X^{1,i}_l(k+1) = X_l(k) + \mu_i A_{il}^H (\Phi_{il} - A_{il} X_l(k) B_{il} B_{il}^H), \quad i \in I[1,q], l \in I[1,p],
\end{equation}

\begin{equation}
X^{2,i}_l(k+1) = X_l(k) + \mu_i C_{il}^H (\Omega_{il} - C_{il} X_l(k) D_{il} D_{il}^H), \quad i \in I[1,q], l \in I[1,p].
\end{equation}

Substituting Eqs. (3.1) and (3.2) into Eqs. (3.5) and (3.6), respectively, we get

\begin{equation}
X^{1,i}_l(k+1) = X_l(k) + \mu_i A_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j B_{ij} + C_{ij} X_j D_{ij}) + A_{il} X_l B_{il} \right]
- A_{il} X_l(k) B_{il}, \quad i \in I[1,q], l \in I[1,p],
\end{equation}

\begin{equation}
X^{2,i}_l(k+1) = X_l(k) + \mu_i C_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j B_{ij} + C_{ij} X_j D_{ij}) + C_{il} X_l D_{il} \right]
- C_{il} X_l(k) D_{il}, \quad i \in I[1,q], l \in I[1,p].
\end{equation}

We can’t implement the algorithms in Eqs. (3.7) and (3.8) through the previous expressions because their right-hand sides contain the unknown matrices \(X_j, j \in I[1,p]\). In order to make the algorithms in Eqs. (3.7) and (3.8) work, the unknown matrices \(X_l, l \in I[1,p]\) in Eqs. (3.7) and (3.8) are respectively replaced with their corresponding estimates \(\hat{X}_l(k), l \in I[1,p]\). In this way, one can obtain the following iterative forms for \(i \in I[1,q], l \in I[1,p]\),

\begin{equation}
X^{1,i}_l(k+1) = X_l(k) + \mu_i A_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k) B_{ij} + C_{ij} X_j(k) D_{ij}) \right] B_{il}^H,
\end{equation}

\begin{equation}
X^{2,i}_l(k+1) = X_l(k) + \mu_i C_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k) B_{ij} + C_{ij} X_j(k) D_{ij}) \right] D_{il}^H.
\end{equation}

(3.10)

Taking the average of \(X^{1,i}_l(k), X^{2,i}_l(k), i \in I[1,q]\), one can obtain the following iterative algorithm,

\begin{equation}
X_l(k+1) = \frac{X^{1,i}_l(k+1) + X^{2,i}_l(k+1)}{2}
= X_l(k) + \frac{\mu_i}{2} A_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k) B_{ij} + C_{ij} X_j(k) D_{ij}) \right] B_{il}^H
+ \frac{\mu_i}{2} C_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k) B_{ij} + C_{ij} X_j(k) D_{ij}) \right] D_{il}^H, \quad i \in I[1,q], l \in I[1,p].
\end{equation}

(3.11)

Next some suitable positive numbers \(\omega_i, i \in I[1,q]\) called as relaxed factors are introduced and used to update \(X_l(k+1), l \in I[1,p]\). These relaxed factors satisfy \(\sum_{i=1}^{q} \omega_i = 1\) and \(0 < \omega_i < 1\).

\begin{equation}
X_l(k+1) = \omega_1 X^{1}_l(k+1) + \omega_2 X^{2}_l(k+1) + \cdots + \omega_q X^{q}_l(k+1), l \in I[1,p].
\end{equation}

(3.12)
Next, we summarize three existing algorithms for solving CSCMEs (1.8).

**Algorithm 3.1.** (The gradient based iterative (GI) algorithm)

Step 1. Input matrices $A_{ij}, C_{ij} \in C_{m_i \times 1}, B_{ij}, D_{ij} \in C_{n_j \times p_i}, F_i \in C_{m_i \times p_i}, i \in I[1,q], j \in I[1,p]$, give any small positive number $\varepsilon$. Choose the initial matrices $X_j(0), j \in I[1,p]$, set $k := 0$;

Step 2. If $\delta_k = \frac{\sum_{i=1}^{n} \|F_i - \sum_{j=1}^{p} (A_{ij} X_j(k)B_{ij} + C_{ij} X_j(k)D_{ij})\|_F}{\sum_{i=1}^{n} \|F_i\|_F} < \varepsilon$, stop; otherwise, go to Step 3;

Step 3. For $l \in I[1,p]$, update the sequences

$$X_l(k+1) = X_l(k) + \frac{\mu}{2} \sum_{i=1}^{n} A_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k)B_{ij} + C_{ij} X_j(k)D_{ij}) \right] B_{il}^H$$

$$+ \frac{\mu}{2} \sum_{i=1}^{n} C_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k)B_{ij} + C_{ij} X_j(k)D_{ij}) \right] D_{il}^H;$$

Step 4. Set $k := k + 1$, return to Step 2.

**Algorithm 3.2.** (The relaxed gradient based iterative (RGI) algorithm)

Step 1. Input matrices $A_{ij}, C_{ij} \in C_{m_i \times 1}, B_{ij}, D_{ij} \in C_{n_j \times p_i}, F_i \in C_{m_i \times p_i}, i \in I[1,q], j \in I[1,p]$. Give any small positive number $\varepsilon$ and appropriative positive numbers $\omega_l$ such that $0 < \omega_l < 1, l \in I[1,p]$. Choose the initial matrices $X_j(0), j \in I[1,p]$, set $k := 1$;

Step 2. Choose the initial matrices $X_l^{(1)}(0)$ and $X_l^{(2)}(0), l \in I[1,p]$. Compute $X_l(0) = \omega_l X_l^{(1)}(0) + (1 - \omega_l) X_l^{(2)}(0), l \in I[1,p]$, set $k := 1$;

Step 3. If $\delta_{k,l} = \frac{\|F_x - \sum_{j=1}^{p} (A_{ij} X_j(k)B_{ij} + C_{ij} X_j(k)D_{ij})\|_F}{\|F_i\|} < \varepsilon$, stop; otherwise, go to Step 4;

Step 4. For $l \in I[1,p]$, update the sequences

$$X_l^{(1)}(k) = X_l^{(1)}(k-1) + (1 - \omega_l) \mu \sum_{i=1}^{n} A_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k)B_{ij} + C_{ij} X_j(k)D_{ij}) \right] B_{il}^H,$$

$$X_l^{(2)}(k) = X_l^{(2)}(k-1) + \omega_l \mu \sum_{i=1}^{n} C_{il}^H \left[ F_i - \sum_{j=1}^{p} (A_{ij} X_j(k)B_{ij} + C_{ij} X_j(k)D_{ij}) \right] D_{il}^H.$$

Compute $X_l(k) = \omega_l X_l^{(1)}(k) + (1 - \omega_l) X_l^{(2)}(k), l \in I[1,p]$;

Step 5. Set $k := k + 1$, return to Step 3.

**Algorithm 3.3.** (The generalized relaxed gradient based iterative (GRGI) algorithm)

Step 1. Input matrices $A_{ij}, C_{ij} \in C_{m_i \times 1}, B_{ij}, D_{ij} \in C_{n_j \times p_i}, F_i \in C_{m_i \times p_i}, i \in I[1,q], j \in I[1,p]$. Give any small positive number $\varepsilon$ and appropriative positive numbers $\alpha_i$ such that $\alpha_i > 0, i \in I[1,q]$. Choose the initial matrices $X_j(0), j \in I[1,p]$, set $k := 1$;

Step 2. Choose initial matrices $X_l^{(1)}(0)$ and $X_l^{(2)}(0), l \in I[1,p]$. Compute $X_l(0) = \omega_l X_l^{(1)}(0) + (1 - \omega_l) X_l^{(2)}(0), l \in I[1,p], k := 1$;
Step 3. If $\delta_{k,i} = \frac{\|F_i - \sum_{j=1}^{p} (A_{ij}X_j(k)B_{ij} + C_{ij}X_j(k)D_{ij})\|}{\|F_i\|} < \varepsilon$, stop; otherwise, go to Step 4.

Step 4. For $l \in I[1, p]$, update the sequences

$$X^{(1)}_l(k) = X^{(1)}_l(k - 1) + \mu_l \left\{ A_{il}^H \left( F_1 - \sum_{j=1}^{p} (A_{ij}X_j(k - 1)B_{ij} + C_{ij}X_j(k - 1)D_{ij}) \right) \right\}$$

$$\times \overline{B_{il}^H} + C_{il}^TF_1 - \sum_{j=1}^{p} (A_{ij}X_j(k - 1)B_{ij} + C_{ij}X_j(k - 1)D_{ij}) \right\}D_{il}^T,$$

... ...

$$X^{(q)}_l(k) = X^{(q)}_l(k - 1) + \mu_l \left\{ A_{il}^H \left( F_q - \sum_{j=1}^{p} (A_{ij}X_j(k - 1)B_{ij} + C_{ij}X_j(k - 1)D_{ij}) \right) \right\}$$

$$\times \overline{B_{il}^H} + C_{il}^TF_q - \sum_{j=1}^{p} (A_{ij}X_j(k - 1)B_{ij} + C_{ij}X_j(k - 1)D_{ij}) \right\}D_{il}^T.$$
4. Convergence analysis

First, we discuss the convergence properties of the MRGI algorithm. For convenience, we introduce the following notation,

\[ A := \begin{pmatrix} (P_{n1}(B_{11})_\sigma)^T \otimes ((A_{11})_\sigma P_{11}) + (D_{11})_\sigma^T \otimes (C_{11})_\sigma & \cdots & \cdot & \cdot \cdot \\
\vdots & \ddots & \ddots & \ddots \\
(P_{n1}(B_{q1})_\sigma)^T \otimes ((A_{q1})_\sigma P_{11}) + (D_{q1})_\sigma^T \otimes (C_{q1})_\sigma & \cdots & \cdot & \cdot \cdot \\
\cdot & \cdot & \cdot & \cdot \\
(P_{np}(B_{1p})_\sigma)^T \otimes ((A_{1p})_\sigma P_{1p}) + (D_{1p})_\sigma^T \otimes (C_{1p})_\sigma & \cdots & \cdot & \cdot \cdot \\
\cdot & \cdot & \cdot & \cdot \\
(P_{np}(B_{qp})_\sigma)^T \otimes ((A_{qp})_\sigma P_{1p}) + (D_{qp})_\sigma^T \otimes (C_{qp})_\sigma \end{pmatrix}. \] (4.1)

It is easy to see \( A \in \mathbb{R}^{\sum_{i=1}^{q} 4m_i p_i \times \sum_{j=1}^{p} 4n_j l_j} \). Then, we have the following results.

**Lemma 4.1.** The CSCMEs (1.8) have unique solutions if and only if the matrix \( A \) is nonsingular, the unique solution is given by

\[
\begin{pmatrix} \text{vec}((X_1)_\sigma) \\ \text{vec}((X_2)_\sigma) \\ \vdots \\ \text{vec}((X_p)_\sigma) \end{pmatrix} = A^{-1} \begin{pmatrix} \text{vec}((F_1)_\sigma) \\ \text{vec}((F_2)_\sigma) \\ \vdots \\ \text{vec}((F_p)_\sigma) \end{pmatrix}, \tag{4.2}
\]

and the corresponding homogeneous matrix equations \( \sum_{j=1}^{p} (A_{ij})_\sigma P_{ij}(X_j)_\sigma + (C_{ij})_\sigma (X_j)_\sigma D_{ij} \sigma = (F_i)_\sigma, i \in I[1, q] \), have the unique solutions \( X_1 = X_2 = \cdots = X_p = 0 \).

**Proof.** Apply the real representation of the complex matrix to CSCMEs (1.8), one has

\[
\sum_{j=1}^{p} \left( (A_{ij})_\sigma P_{ij}(X_j)_\sigma + (C_{ij})_\sigma (X_j)_\sigma (D_{ij})_\sigma \right) = (F_i)_\sigma, i \in I[1, q]. \tag{4.3}
\]

By using Kronecker products of matrices and vector stretching operator, the preceding expression can be transformed into \( Ax = f \) with

\[
x = \begin{pmatrix} \text{vec}((X_1)_\sigma) \\ \text{vec}((X_2)_\sigma) \\ \vdots \\ \text{vec}((X_p)_\sigma) \end{pmatrix}, \quad f = \begin{pmatrix} \text{vec}((F_1)_\sigma) \\ \text{vec}((F_2)_\sigma) \\ \vdots \\ \text{vec}((F_p)_\sigma) \end{pmatrix}, \tag{4.4}
\]

where \( A \) is given by (4.1). Therefore, the CSCMEs (1.8) have unique solutions if and only if the matrix \( A \) is nonsingular. The conclusion follows immediately. \( \Box \)
Theorem 4.1. Suppose that the CSCMEs (1.8) have a unique solution \((X_1^*, X_2^*, \ldots, X_p^*)\). If \(\mu_i\) satisfies
\[
0 < \mu_i < \frac{4}{q\omega_i \sum_{l=1}^p (\|A_{il}\|_2^2 \|B_{il}\|_2^2 + \|C_{il}\|_2^2 \|D_{il}\|_2^2)},
\]
then the iterative sequences \(X_i(k), l \in [1, p]\) generated by the Algorithm 3.4 converge to \((X_1^*, X_2^*, \ldots, X_p^*)\), i.e., \(\lim_{k \to \infty} X_i(k) = X_i^*, l \in [1, p]\) or the error matrices \(X_i(k) - X_i^*, l \in [1, p]\) converge to zero for any initial values \(X_i(0), l \in [1, p]\), where \((\cdot)_\sigma\) is the real representation of complex matrix.

Proof. Define the error matrices
\[
\bar{X}_i(k) := X_i(k) - X_i^*, l \in [1, p],
\]
and
\[
\theta_i(k) := F_i - \sum_{j=1}^p (A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij}), i \in [1, q].
\]
From Eq.(4.7), it is easy to derive
\[
\theta_i(k) = -\sum_{j=1}^p (A_{ij}\bar{X}_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij}), i \in [1, q].
\]
It follows from Algorithm 3.4 that
\[
\bar{X}_i(k + 1) = X_i(k + 1) - X_i^* = \sum_{i=1}^q \omega_i X_i^*(k + 1) - X_i^* = \sum_{i=1}^q \omega_i [X_i^*(k + 1) - X_i^*] \\
= \sum_{i=1}^q \omega_i X_i^*(k + 1) - X_i^* + X_i^*(k + 1) - X_i^* \\
= \sum_{i=1}^q \omega_i \frac{2\bar{X}_i(k) + \mu_i (A_{il}^H\theta_i(k)B_{il}^H + C_{il}^H\theta_i(k)D_{il}^H)}{2} \\
= \bar{X}_i(k) + \sum_{i=1}^q \omega_i \mu_i \frac{A_{il}^H\theta_i(k)B_{il}^H + C_{il}^H\theta_i(k)D_{il}^H}{2}.
\]
Thus, by the properties of norm, one has
\[
\|\bar{X}_i(k + 1)\|_F^2 = \|\bar{X}_i(k) + \sum_{i=1}^q \omega_i \mu_i \frac{A_{il}^H\theta_i(k)B_{il}^H + C_{il}^H\theta_i(k)D_{il}^H}{2}\|_F^2 \\
= \|\bar{X}_i(k)\|_F^2 + \left\|\sum_{i=1}^q \omega_i \mu_i \frac{A_{il}^H\theta_i(k)B_{il}^H + C_{il}^H\theta_i(k)D_{il}^H}{2}\right\|_F^2 \\
+ \text{tr} \left[\bar{X}_i(k)^H \sum_{i=1}^q \omega_i \mu_i \frac{A_{il}^H\theta_i(k)B_{il}^H + C_{il}^H\theta_i(k)D_{il}^H}{2}\right].
\]
Moreover, note that the following expression is real,
\[
\text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i \overline{\tilde{X}_i(k)} H C_{il}^{-H} \overline{\theta_i(k) D_{il}^{-H}}\right) + \text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i D_{il} \overline{\theta_i(k)} C_{il}^{-H} \overline{\tilde{X}_i(k)}\right).
\] (4.11)

Therefore, we get
\[
\text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i \overline{\tilde{X}_i(k)} H C_{il}^{-H} \overline{\theta_i(k) D_{il}^{-H}}\right) + \text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i D_{il} \overline{\theta_i(k)} C_{il}^{-H} \overline{\tilde{X}_i(k)}\right)
= \text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i \overline{\tilde{X}_i(k)} H C_{il}^{-H} \overline{\theta_i(k) D_{il}^{-H}}\right) + \text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i D_{il} \overline{\theta_i(k)} C_{il}^{-H} \overline{\tilde{X}_i(k)}\right),
\] (4.12)

and
\[
\text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i \overline{\tilde{X}_i(k)} H C_{il}^{-H} \overline{\theta_i(k) D_{il}^{-H}}\right) + \text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i \overline{\tilde{X}_i(k)} H C_{il}^{-H} \overline{\theta_i(k) D_{il}^{-H}}\right)
+ \text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i D_{il} \overline{\theta_i(k)} C_{il}^{-H} \overline{\tilde{X}_i(k)}\right) + \text{tr}\left(\sum_{i=1}^{q} \omega_i \mu_i D_{il} \overline{\theta_i(k)} C_{il}^{-H} \overline{\tilde{X}_i(k)}\right)
= \text{tr}\left[\sum_{i=1}^{q} \omega_i \mu_i \left(B_{il}^{H} \overline{\tilde{X}_i(k)} H A_{il}^{H} B_{il}^{-H} + D_{il}^{H} \overline{\tilde{X}_i(k)} H C_{il}^{H} D_{il}^{-H}\right) \theta_i(k)\right]
+ \text{tr}\left[\sum_{i=1}^{q} \omega_i \mu_i \theta_i(k) H \left(A_{il} \overline{\tilde{X}_i(k)} B_{il} + C_{il} \overline{\tilde{X}_i(k)} D_{il}\right)\right].
\] (4.13)

So Eq.(4.10) becomes
\[
\|\tilde{X}_i(k+1)\|_F^2
= \|\tilde{X}_i(k)\|_F^2 + \left\| \sum_{i=1}^{q} \omega_i \mu_i \frac{A_{il}^{H} \overline{\theta_i(k)} B_{il}^{H} + C_{il}^{H} \overline{\theta_i(k)} D_{il}^{H}}{2} \right\|_F^2.
\]
\[ + \frac{1}{2} \text{tr} \left[ \sum_{i=1}^{q} \omega_{i} \mu_{i} \left( B_{il}^{H} \bar{X}_{i}(k) A_{il}^{H} + D_{il}^{H} \bar{X}_{i}(k) C_{il}^{H} \right) \theta_{i}(k) \right] \]

\[ + \frac{1}{2} \text{tr} \left[ \sum_{i=1}^{q} \omega_{i} \mu_{i} \theta_{i}(k) \left( A_{il} \bar{X}_{i}(k) B_{il} + C_{il} \bar{X}_{i}(k) D_{il} \right) \right]. \]  

(4.14)

Adding all \( \| \bar{X}_{i}(k+1) \|_{F}^{2} \), \( i \in I[1, q] \) on both sides of Eq. (4.14), we have

\[ \sum_{i=1}^{p} \| \bar{X}_{i}(k+1) \|_{F}^{2} \]

\[ = \sum_{i=1}^{p} \| \bar{X}_{i}(k) \|_{F}^{2} + \sum_{i=1}^{q} \| \sum_{i=1}^{q} \omega_{i} \mu_{i} A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \|_{F}^{2} \]

\[ + \frac{1}{2} \sum_{i=1}^{p} \text{tr} \left[ \sum_{i=1}^{q} \omega_{i} \mu_{i} \left( B_{il}^{H} \bar{X}_{i}(k) A_{il}^{H} + D_{il}^{H} \bar{X}_{i}(k) C_{il}^{H} \right) \theta_{i}(k) \right] \]

\[ + \frac{1}{2} \sum_{i=1}^{p} \text{tr} \left[ \sum_{i=1}^{q} \omega_{i} \mu_{i} \theta_{i}(k) \left( A_{il} \bar{X}_{i}(k) B_{il} + C_{il} \bar{X}_{i}(k) D_{il} \right) \right]. \]  

(4.15)

It is easy to know that

\[ \left\| \sum_{i=1}^{q} \omega_{i} \mu_{i} A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ = \sum_{i=1}^{q} \frac{\omega_{i}^{2} \mu_{i}^{2}}{4} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ + \sum_{1 \leq i \neq j \leq q} \frac{\omega_{i} \mu_{i} \omega_{j} \mu_{j}}{4} \text{tr} \left[ \left( A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right) \right] ^{H} \]

\[ \times \left( A_{jl}^{H} \theta_{j}(k) B_{jl}^{H} + C_{jl}^{H} \theta_{j}(k) D_{jl}^{H} \right) \]

\[ = \sum_{i=1}^{q} \frac{\omega_{i}^{2} \mu_{i}^{2}}{4} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ + \sum_{1 \leq i \neq j \leq q} \frac{\omega_{i} \mu_{i} \omega_{j} \mu_{j}}{4} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ \leq \sum_{i=1}^{q} \frac{\omega_{i}^{2} \mu_{i}^{2}}{4} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ + \sum_{1 \leq i \neq j \leq q} \frac{\omega_{i} \mu_{i} \omega_{j} \mu_{j}}{4} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ \times \left\| A_{jl}^{H} \theta_{j}(k) B_{jl}^{H} + C_{jl}^{H} \theta_{j}(k) D_{jl}^{H} \right\|_{F} \]

\[ \leq \sum_{i=1}^{q} \frac{\omega_{i}^{2} \mu_{i}^{2}}{4} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ + \sum_{1 \leq i \neq j \leq q} \frac{\omega_{i} \mu_{i} \omega_{j} \mu_{j}}{4} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} + C_{il}^{H} \theta_{i}(k) D_{il}^{H} \right\|_{F}^{2} \]

\[ \times \left\| A_{jl}^{H} \theta_{j}(k) B_{jl}^{H} + C_{jl}^{H} \theta_{j}(k) D_{jl}^{H} \right\|_{F} \]

\[ \leq \sum_{i=1}^{q} \frac{\omega_{i}^{2} \mu_{i}^{2}}{8} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} \right\|_{F}^{2} + \sum_{1 \leq i \neq j \leq q} \frac{\omega_{i} \mu_{i} \omega_{j} \mu_{j}}{8} \left\| A_{il}^{H} \theta_{i}(k) B_{il}^{H} \right\|_{F}^{2} \]
A novel iterative method for solving the CSCMEs

\begin{align}
&+ \frac{C_{il}^H \theta_i(k) D_{il}^H}{2} \left[ \frac{\omega_i^2 \mu_i^2}{2} \left| A_{il}^H \theta_i(k) B_{il}^H + C_{il}^H \theta_i(k) D_{il}^H \right|^2 \right] \\
&= -q \sum_{i=1}^{q} \omega_i^2 \mu_i^2 \left| A_{il}^H \theta_i(k) B_{il}^H + C_{il}^H \theta_i(k) D_{il}^H \right|^2. 
\end{align}

Moreover, we have

\begin{align}
&\left| A_{il}^H \theta_i(k) B_{il}^H + C_{il}^H \theta_i(k) D_{il}^H \right|^2 \\
&= \frac{1}{2} \left\| \left( A_{il}^H \theta_i(k) B_{il}^H + C_{il}^H \theta_i(k) D_{il}^H \right) \right\|^2 \\
&= \frac{1}{2} \left\| (A_{il}^H)_{\sigma} P_m ((\theta_i(k))_{\sigma}) P_{ni} (B_{il}^H)_{\sigma} + (C_{il}^H)_{\sigma} (\theta_i(k))_{\sigma} (D_{il}^H)_{\sigma} \right\|^2 \\
&\leq \frac{1}{2} \left( \left\| (A_{il}^H)_{\sigma} \left\| P_m ((\theta_i(k))_{\sigma}) P_{ni} (B_{il}^H)_{\sigma} \right\|_F^2 + \left\| (C_{il}^H)_{\sigma} \left\| (\theta_i(k))_{\sigma} (D_{il}^H)_{\sigma} \right\|_F^2 \right) \\
&= \frac{1}{2} \left( 2 \left\| A_{il} \right\|^2 \left\| B_{il} \right\|^2 + 2 \left\| C_{il} \right\|^2 \left\| D_{il} \right\|^2 \right) \left\| \theta_i(k) \right\|^2.
\end{align}

Substituting Eq. (4.17) into Eq. (4.16) we get

\begin{align}
&\left\| \sum_{i=1}^{q} \omega_i \mu_i \frac{A_{il}^H \theta_i(k) B_{il}^H + C_{il}^H \theta_i(k) D_{il}^H}{2} \right\|^2 \\
&\leq \frac{1}{4} \sum_{i=1}^{q} \omega_i^2 \mu_i^2 \left( \left\| A_{il} \right\|^2 \left\| B_{il} \right\|^2 + \left\| C_{il} \right\|^2 \left\| D_{il} \right\|^2 \right) \left\| \theta_i(k) \right\|^2.
\end{align}

From Eq. (4.18), one has

\begin{align}
&\sum_{i=1}^{p} \left\| \sum_{i=1}^{q} \omega_i \mu_i \frac{A_{il}^H \theta_i(k) B_{il}^H + C_{il}^H \theta_i(k) D_{il}^H}{2} \right\|^2 \\
&\leq \frac{1}{4} \sum_{i=1}^{q} \omega_i^2 \mu_i^2 \left( \left\| A_{il} \right\|^2 \left\| B_{il} \right\|^2 + \left\| C_{il} \right\|^2 \left\| D_{il} \right\|^2 \right) \left\| \theta_i(k) \right\|^2.
\end{align}

By the relations Eqs. (4.8), (4.15) and (4.19), one can get

\begin{align}
&\sum_{i=1}^{p} \left\| \Xi_i(k + 1) \right\|^2 \\
&\leq \sum_{i=1}^{p} \left\| \Xi_i(k) \right\|^2 + \frac{1}{4} \sum_{i=1}^{q} \sum_{i=1}^{q} \omega_i^2 \mu_i^2 \left( \left\| A_{il} \right\|^2 \left\| B_{il} \right\|^2 + \left\| C_{il} \right\|^2 \left\| D_{il} \right\|^2 \right) \left\| \theta_i(k) \right\|^2 \\
&+ \frac{1}{2} \sum_{i=1}^{p} tr \left[ \sum_{i=1}^{q} \omega_i \mu_i \left( B_{il}^H \Xi_i(k) A_{il}^H + D_{il}^H \Xi_i(k) C_{il}^H \right) \theta_i(k) \right] \\
&+ \frac{1}{2} \sum_{i=1}^{p} tr \left[ \sum_{i=1}^{q} \omega_i \mu_i \theta_i(k) \right] \left( A_{il}^H \Xi_i(k) B_{il} + C_{il} \Xi_i(k) D_{il} \right)
\end{align}
\[= \sum_{l=1}^{p} \|\overline{X}_l(k)\|_F^2 + \frac{q}{4} \sum_{l=1}^{p} \sum_{i=1}^{q} \omega_i^2 \mu_i^2 \left(\|A_{il}\|_2^2 \|B_{il}\|_2^2 + \|C_{il}\|_2^2 \|D_{il}\|_2^2\right) \|\theta_i(k)\|_F^2 \]
\[+ \frac{1}{2} \sum_{i=1}^{q} \omega_i \mu_i \text{tr} \left[ \sum_{l=1}^{p} \left( B_{il}^H \overline{X}_l(k) A_{il}^H + D_{il}^H \overline{X}_l(k) C_{il}^H \right) \theta_i(k) \right] \]
\[+ \frac{1}{2} \sum_{i=1}^{q} \omega_i \mu_i \text{tr} \left[ \sum_{l=1}^{p} \left( B_{il} \overline{X}_l(k) B_{il} + C_{il} \overline{X}_l(k) D_{il} \right) \right] \]
\[= \sum_{l=1}^{p} \|\overline{X}_l(k)\|_F^2 + \frac{q}{4} \sum_{l=1}^{p} \sum_{i=1}^{q} \omega_i^2 \mu_i^2 \left(\|A_{il}\|_2^2 \|B_{il}\|_2^2 + \|C_{il}\|_2^2 \|D_{il}\|_2^2\right) \|\theta_i(k)\|_F^2 \]
\[+ \frac{1}{2} \sum_{i=1}^{q} \omega_i \mu_i \|\theta_i(k)\|_F^2 \]
\[= \sum_{l=1}^{p} \|\overline{X}_l(0)\|_F^2 - \sum_{i=1}^{q} \omega_i \mu_i \left[1 - \frac{q}{4} \omega_i \mu_i \sum_{l=1}^{p} \left(\|A_{il}\|_2^2 \|B_{il}\|_2^2 + \|C_{il}\|_2^2 \|D_{il}\|_2^2\right) \right] \times \|\theta_i(k)\|_F^2 \]
\[= \sum_{l=1}^{p} \|\overline{X}_l(0)\|_F^2 - \sum_{j=1}^{k} \omega_i \mu_i \left[1 - \frac{q}{4} \omega_i \mu_i \sum_{l=1}^{p} \left(\|A_{il}\|_2^2 \|B_{il}\|_2^2 + \|C_{il}\|_2^2 \|D_{il}\|_2^2\right) \right] \times \|\theta_i(j)\|_F^2 . \]

Thus, if the convergence factors are chosen to satisfy Eq.(4.5), then for any initial values \(X_i(0), i \in I[1, p]\), one has
\[\sum_{j=1}^{\infty} \sum_{i=1}^{q} \|\theta_i(j)\|_F^2 < \infty. \tag{4.20}\]

This implies that
\[\lim_{k \to \infty} \theta_i(k) = 0, i \in I[1, q]. \tag{4.21}\]

Therefore, we obtain
\[\lim_{k \to \infty} \sum_{j=1}^{p} \left( A_{ij} \overline{X}_j(k) B_{ij} + C_{ij} \overline{X}_j(k) D_{ij} \right) = F_i, i \in I[1, q]. \tag{4.22}\]

It follows that
\[\lim_{k \to \infty} X_i(k) = X_i^*, i \in I[1, p]. \tag{4.23}\]

So the desired result follows.
In the following, we study the necessary and sufficient conditions of convergence for Algorithm 3.1 and consider the optimal $\mu$ value to make it converge in the maximum convergence rate. This result can be stated as Theorem 4.2. The proof of this theorem has the same line as Theorem 3.1 and 4.1 in Huang and Ma [14]. However, it uses a different real representation matrix as a tool in this paper with that in [14]. For convenience, we rewrite the proof of this theorem.

**Theorem 4.2.** Suppose that the CSCMEs (1.8) have a unique solution $(X(1)^*, X(2)^*, \cdots, X(p)^*)$, then the GI algorithm yields $\lim_{k \to \infty} X_l(k) = X_l^*$, $l \in I[1, p]$ for any initial matrices $X_l(0)$, $l \in I[1, p]$ if and only if

$$0 < \mu < \frac{4q}{\sigma_{\text{max}}^2(A)}. \quad (4.24)$$

Moreover, the $F$-convergence rate of the GI algorithm is maximized if

$$\mu = \mu_{\text{opt}} = \frac{4q}{\sigma_{\text{max}}^2(A) + \sigma_{\text{min}}^2(A)}, \quad (4.25)$$

where the matrix $A$ is defined in Eq.(4.1).

**Proof.** Subtract $X_l^*$, $l \in I[1, p]$ on both sides of Algorithm 3.1, we get

$$\tilde{X}_l(k+1) = \tilde{X}_l(k) - \frac{\mu}{2} \sum_{i=1}^{q} \sum_{j=1}^{p} \begin{bmatrix} A_{il}^H \sum_{j=1}^{p} (A_{ij} \tilde{X}_j(k)B_{ij} + C_{ij} \tilde{X}_j(k)D_{ij})B_{il}^H \\
+ C_{il}^T \sum_{j=1}^{p} (A_{ij} \tilde{X}_j(k)B_{ij} + C_{ij} \tilde{X}_j(k)D_{ij})D_{il}^T \end{bmatrix}$$

$$= \tilde{X}_l(k) - \frac{\mu}{2} \sum_{i=1}^{q} \sum_{j=1}^{p} \begin{bmatrix} A_{il}^H (A_{ij} \tilde{X}_j(k)B_{ij} + C_{ij} \tilde{X}_j(k)D_{ij})B_{il}^H \\
+ C_{il}^T (A_{ij} \tilde{X}_j(k)B_{ij} + C_{ij} \tilde{X}_j(k)D_{ij})D_{il}^T \end{bmatrix}, \quad (4.26)$$

where $\tilde{X}_l(k) = X_l(k) - X_l^*$, $l \in I[1, p]$. Combining this relation with the real representation of complex matrix, one has

$$(\tilde{X}_l(k+1))_{\sigma}$$

$$= (\tilde{X}_l(k))_{\sigma} - \frac{\mu}{2} \sum_{i=1}^{q} \sum_{j=1}^{p} \begin{bmatrix} (A_{il}^H)_{\sigma} P_{mi} (A_{ij} \tilde{X}_j(k)B_{ij} + C_{ij} \tilde{X}_j(k)D_{ij})_{\sigma} \\
\times P_{pi} (B_{il}^H)_{\sigma} + (C_{il}^T)^T \sigma (A_{ij} \tilde{X}_j(k)B_{ij} + C_{ij} \tilde{X}_j(k)D_{ij})_{\sigma} (D_{il}^T)_{\sigma} \end{bmatrix}. \quad (4.27)$$

Taking the vec-operator on both sides of above relation, we can obtain

$$\text{vec}\left[(\tilde{X}_l(k+1))_{\sigma}\right]$$

$$= \text{vec}\left[(\tilde{X}_l(k))_{\sigma}\right] - \frac{q}{2} \sum_{i=1}^{q} \sum_{j=1}^{p} \frac{\mu}{2} \begin{bmatrix} (P_{pi} (B_{il}^H)_{\sigma})^T \otimes ((A_{il}^H)_{\sigma} P_{mi}) \\
+ ((D_{il}^T)_{\sigma})^T \otimes (C_{il}^T)_{\sigma} \text{vec}\left[(A_{ij} \tilde{X}_j(k)B_{ij} + C_{ij} \tilde{X}_j(k)D_{ij})_{\sigma}\right] \end{bmatrix}$$
So we have
\[
\text{vec}
\left[
\tilde{X}_\sigma(k+1)
\right] = \text{vec}
\left[
\tilde{X}_\sigma(k)
\right] - \frac{\mu}{2q} A^T A \text{vec}
\left[
\tilde{X}_\sigma(k)
\right]
\]
\[
= \left(I - \frac{\mu}{2q} A^T A\right) \text{vec}
\left[
\tilde{X}_\sigma(k)
\right],
\]
(4.29)
where \(\text{vec}
\left[
\tilde{X}_\sigma(k)
\right] = \left[
\text{vec}(\tilde{X}_1(k))_\sigma
\right]^T, \left[
\text{vec}(\tilde{X}_2(k))_\sigma
\right]^T, \ldots, \left[
\text{vec}(\tilde{X}_p(k))_\sigma
\right]^T, T\).

This equation is a linear matrix equation with the coefficient matrix \(\Upsilon := I - \frac{\mu}{2q} A^T A\). Thus, the gradient based iterative algorithm converges for any initial matrices \(X_i(0), i \in [1,p]\) if and only if \(\rho(\Upsilon) < 1\). Since \(A^T A\) is a symmetric matrix, we have
\[
\rho(\Upsilon) = \max_{1 \leq i \leq \sum_{j=1}^{4n q_{t_1}} |\lambda_i(\Upsilon)| = \max_{1 \leq i \leq \sum_{j=1}^{4n q_{t_1}} |1 - \frac{\mu}{2q} \lambda_i(A^T A)| < 1.\)
(4.30)
Therefore, we have \(0 < \mu < \frac{4q}{\sigma_{\text{max}}(A)}\). According to Lemma 2.4, if
\[
\mu = \mu_{opt} = \frac{4q}{\sigma_{\text{max}}^2(A) + \sigma_{\text{min}}^2(A)},
\]
(4.31)
then the maximal convergence rate of Algorithm 3.1 can be reached. \(\square\)

5. A more general case

In this section, we extend the idea of Algorithm 3.4 to solve a more general coupled Sylvester conjugate matrix equations
\[
\sum_{j=1}^{s_{i_1}} A_{i_1j} X_1 B_{i_1j} + \sum_{j=1}^{w_{i_1}} C_{i_1j} \overline{X}_1 D_{i_1j} + \cdots + \sum_{j=1}^{s_{i_p}} A_{i_pj} X_p B_{i_pj} + \sum_{j=1}^{w_{i_p}} C_{i_pj} \overline{X}_p D_{i_pj} = F_i,
\]
(5.1)
where \(A_{ij}, C_{ij} \in C^{m_{i} \times n_{i}}, B_{ij}, D_{ij} \in C^{s_{ij} \times n_{i}}, i \in [1,N], \eta \in I[1,p]\) are the given matrices, and \(X_{\eta} \in C^{r_{\eta} \times s_{\eta}}, \eta \in I[i,p]\) are the unknown matrices to be determined.
In the same way, the iterative algorithm of the equations (5.1) is constructed as follows,

\[
X_{1}^{(k+1)l} = X_{1}^{(kl)} + \mu_{i} \sum_{j=1}^{s_{il}} A_{ilj}^{H} \left[ F_{i} - \left( \sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} A_{iut} X_{u}^{(k)} B_{iut} \right) \right] + \sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} C_{iut} X_{u}^{(k)} D_{iut}^{H},
\]

(5.2)

\[
X_{2}^{(k+1)l} = X_{1}^{(kl)} + \mu_{i} \sum_{j=1}^{w_{il}} C_{ilj}^{H} \left[ F_{i} - \left( \sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} A_{iut} X_{u}^{(k)} B_{iut} \right) \right] + \sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} C_{iut} X_{u}^{(k)} D_{iut}^{H},
\]

(5.3)

\[
X_{l}^{(k+1)} = \frac{X_{1}^{(k+1)l} + X_{2}^{(k+1)l}}{2}
\]

\[
X_{l}(k+1) = \omega_{1} X_{1}^{(k+1)l} + \omega_{2} X_{2}^{(k+1)l} + \cdots + \omega_{q} X_{q}^{(k+1)l}, l \in I[1,p] .
\]

(5.4)

(5.5)

Similar to Theorem 4.1, we have the following Theorem 5.1 and its proof is omitted here.

**Theorem 5.1.** If the equations (5.1) have a unique solution \((X_{1*}, X_{2*}, \cdots, X_{p*})\), then the iterative solution \(X_{l}(k), l \in I[1,p]\) given by algorithm (5.2)-(5.5) converge to \(X_{l*}, l \in I[1,p]\) for arbitrary initial values \(X_{l}(0), l \in I[1,p]\) if

\[
0 < \mu_{i} < \frac{4}{q \omega_{j} \sum_{l=1}^{p} \left( \sum_{j=1}^{s_{ij}} \| A_{ilj} \|_{F}^{2} \| B_{ilj} \|_{F}^{2} + \sum_{j=1}^{w_{il}} \| C_{ilj} \|_{F}^{2} \| D_{ilj} \|_{F}^{2} \right)^{2}}.
\]

(5.6)

6. Numerical examples

In this section, two numerical examples are presented to show the effectiveness of the MRGI Method. All the computations are performed on Intel Pentium(R) Dual-Core CPU T4300 XP system by using MATLAB 7.0.

**Example 6.1.** In this example, we consider the generalized coupled Sylvester-
given in Table 2 and 3. We can see the advantages of the proposed algorithm in Huang and Ma [115x213] where

conjugate matrix equations

\[
\begin{aligned}
A_{11}X_{1}B_{11} + C_{11}X_{1}D_{11} + A_{12}X_{2}B_{12} + C_{12}X_{2}D_{12} &= F_{1}, \\
A_{21}X_{1}B_{21} + C_{21}X_{1}D_{21} + A_{22}X_{2}B_{22} &= F_{2},
\end{aligned}
\]

with the following coefficient matrices:

\[
\begin{align*}
A_{11} &= \begin{bmatrix} 2 - 2i & 2i \\ 8 + i & 2 + 3i \end{bmatrix}, & B_{11} &= \begin{bmatrix} 0.5 - i & -1 + 3i \\ -1.5 + 2i & 1 - 2i \end{bmatrix}, & C_{11} &= \begin{bmatrix} 4i & 2 + 2i \\ 3 + 2.5i & i \end{bmatrix}, \\
C_{12} &= \begin{bmatrix} 1 + 3i & 4 - i \\ 2 - 3i & 1 + 2i \end{bmatrix}, & D_{12} &= \begin{bmatrix} 1 + 2.5i & -3 + i \\ -1.05i & 1 + 2i \end{bmatrix}, & D_{21} &= \begin{bmatrix} 4 - i & 1.5 - i \\ i & -2 + 2i \end{bmatrix}, \\
D_{11} &= \begin{bmatrix} -2 + i & 3i \\ -1 & 3 + i \end{bmatrix}, & A_{12} &= \begin{bmatrix} 1 - 1.5i & 3i \\ -2 + 3i & 4 \end{bmatrix}, & B_{12} &= \begin{bmatrix} 1 - 2i & -1 + 4i \\ -1 + 3i & 1 \end{bmatrix}, \\
B_{21} &= \begin{bmatrix} -1 - i & -3i \\ 5 & 1 + 2i \end{bmatrix}, & C_{21} &= \begin{bmatrix} -1 - i & 2 - i \\ -2 + 3i & 1 + 2i \end{bmatrix}, & B_{22} &= \begin{bmatrix} 3 + i & 2 + 3i \\ 3 & 1 - 7i \end{bmatrix}, \\
A_{22} &= \begin{bmatrix} 1 - 4i & 1 + i \\ -1 + 3i & 2 \end{bmatrix}, & A_{21} &= \begin{bmatrix} -1 + 0.5i & 0.5 \\ 1 - 2i & -2.5 + 1.5i \end{bmatrix}, \\
F_{1} &= \begin{bmatrix} 74 + 52.5i & -124 + 38.5i \\ 23 + 44.5i & -134 + 83i \end{bmatrix}, & F_{2} &= \begin{bmatrix} 18 - 6i & -23 - 17i \\ -21.5 - 27.5i & 103 + 22.5i \end{bmatrix}.
\end{align*}
\]

These matrix equations have a unique solution

\[
X_{1} = \begin{bmatrix} 1 + i & 2 - 3i \\ -1 + 2i & -2 + 3i \end{bmatrix}, \quad X_{2} = \begin{bmatrix} 2 + i & 3 + i \\ 3 + 2i & 1 + 2i \end{bmatrix}.
\]

Choose \(X_{1}(0) = X_{2}(0) = X_{1}^{(0)} = X_{2}^{(0)} = 10^{-6} \times I_{2}\) for \(i = 1, 2\) as the initial iterative matrices and define the relative iterative error is

\[
f(k) = \sqrt{\frac{|X_{1}(k) - X_{1}|^2 + |X_{2}(k) - X_{2}|^2}{|X_{1}|^2 + |X_{2}|^2}},
\]

where \(X_{1}(k)\) and \(X_{2}(k)\) are the \(k\)th solution of the corresponding matrix equations.

The relative residual of these algorithms is illustrated in Figure 1, in which it can be concluded that the efficiency of the MRGI algorithm is faster than the GI algorithm in Wu et al. [24], the RGI algorithm in Huang and Ma [15] and the GRGI algorithm in Huang and Ma [15].

In Table 1, it is clear that the iterative solution obtained by the MRGI algorithm converges to the exact solution with the increase of iteration number \(k\). In addition, the iterative steps, relative residual and computational time results are given in Table 2 and 3. We can seen the advantages of the proposed algorithm in
A novel iterative method for solving the CSCMEs convergence accuracy and efficiency by comparing these results.

![Comparison of convergence curve](image)

**Figure 1.** Comparison of convergence curve

**Table 1.** The iterative solution for the MRGI algorithm with $\mu_1 = 2.2 \times 10^{-3}$, $\mu_2 = 1.8 \times 10^{-3}$ and $\omega_1 = 0.4$, $\omega_2 = 0.6$

<table>
<thead>
<tr>
<th>k</th>
<th>$X_1$</th>
<th>$X_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>$0.7453 + 1.0247i$ $2.1434 - 2.9552i$</td>
<td>$2.0322 + 0.6542i$ $2.9387 + 0.8960i$</td>
</tr>
<tr>
<td></td>
<td>$-0.7792 + 1.6749i$ $-1.5885 + 3.2316i$</td>
<td>$3.2106 + 1.9661i$ $0.9123 + 1.8146i$</td>
</tr>
<tr>
<td></td>
<td>$0.8835 + 1.0113i$ $2.0664 - 2.9755i$</td>
<td>$2.0165 + 0.8406i$ $2.9730 + 0.9507i$</td>
</tr>
<tr>
<td>400</td>
<td>$-0.8953 + 1.8465i$ $-1.8154 + 3.1092i$</td>
<td>$3.0941 + 1.9933i$ $0.9600 + 1.9137i$</td>
</tr>
<tr>
<td></td>
<td>$0.9479 + 1.0053i$ $2.0325 - 2.9984i$</td>
<td>$2.0077 + 0.9298i$ $2.9876 + 0.9733i$</td>
</tr>
<tr>
<td>600</td>
<td>$-0.9513 + 1.9291i$ $-1.9157 + 3.0504i$</td>
<td>$3.0427 + 1.9978i$ $0.9813 + 1.9602i$</td>
</tr>
<tr>
<td>800</td>
<td>$0.9755 + 1.0025i$ $2.0150 - 2.9940i$</td>
<td>$2.0035 + 0.9678i$ $2.9943 + 0.9895i$</td>
</tr>
<tr>
<td>1000</td>
<td>$-0.9776 + 1.9674i$ $-1.9613 + 3.0232i$</td>
<td>$3.0195 + 1.9991i$ $0.9914 + 1.9817i$</td>
</tr>
<tr>
<td></td>
<td>$0.9888 + 1.0011i$ $2.0069 - 2.9975i$</td>
<td>$2.0016 + 0.9852i$ $2.9974 + 0.9952i$</td>
</tr>
<tr>
<td>1000</td>
<td>$-0.9897 + 1.9850i$ $-1.9823 + 3.0106i$</td>
<td>$3.0090 + 1.9996i$ $0.9960 + 1.9916i$</td>
</tr>
<tr>
<td>Solution</td>
<td>$1 + i$ $2 - 3i$</td>
<td>$2 + i$ $3 + i$</td>
</tr>
<tr>
<td></td>
<td>$-1 + 2i$ $-2 + 3i$</td>
<td>$3 + 2i$ $1 + 2i$</td>
</tr>
</tbody>
</table>

**Table 2.** Iterative steps, relative residual and computational time results

<table>
<thead>
<tr>
<th>Method</th>
<th>Steps</th>
<th>$f(k)$</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GI algorithm in Wu et al. [24]</td>
<td>4819</td>
<td>$9.5817 \times 10^{-4}$</td>
<td>2.5238</td>
</tr>
<tr>
<td>RGI algorithm in Huang and Ma [15]</td>
<td>3332</td>
<td>$9.5753 \times 10^{-4}$</td>
<td>1.7718</td>
</tr>
<tr>
<td>GRGI algorithm in Huang and Ma [15]</td>
<td>2425</td>
<td>$9.5582 \times 10^{-4}$</td>
<td>0.7904</td>
</tr>
<tr>
<td>MRGI algorithm in this paper</td>
<td>1327</td>
<td>$9.5180 \times 10^{-4}$</td>
<td>0.7798</td>
</tr>
</tbody>
</table>
Table 3. The relative iterative error versus the number of iterations $k$ for the GI algorithm in Wu et al. [24], the RGI algorithm in Huang and Ma [15], the GRGI algorithm in Huang and Ma [15] and the MRGI algorithm in this paper.

<table>
<thead>
<tr>
<th>$f(k)$</th>
<th>GI algorithm</th>
<th>RGI algorithm</th>
<th>GRGI algorithm</th>
<th>MRGI algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>693</td>
<td>479</td>
<td>349</td>
<td>210</td>
</tr>
<tr>
<td>0.01</td>
<td>2731</td>
<td>1888</td>
<td>1374</td>
<td>753</td>
</tr>
<tr>
<td>0.001</td>
<td>4738</td>
<td>3276</td>
<td>2384</td>
<td>1304</td>
</tr>
</tbody>
</table>

Example 6.2. We consider the following coupled Sylvester conjugate matrix equations

$$
\begin{align*}
& A_{11}X_1B_{11} + C_{11}X_1D_{11} + A_{12}X_2B_{12} + C_{12}X_2D_{12} = F_1, \\
& A_{21}X_1B_{21} + C_{21}X_1D_{21} + A_{22}X_2B_{22} + C_{22}X_2D_{22} = F_2.
\end{align*}
$$

(6.3)

with the coefficient matrices:

$A_{11} = \text{diag}(a + \text{diag}(\text{rand}(m))) - \text{tril}(\text{rand}(m), m)i$, $B_{11} = \text{eye}(m) + \text{eye}(m)i$,

$C_{11} = \text{eye}(m) - \text{tril}(\text{rand}(m), m)i$, $D_{11} = \text{rand}(m) + \text{triu}(\text{rand}(m), m)i$,

$A_{12} = \text{diag}(a + \text{diag}(\text{rand}(m))) - \text{tril}(\text{rand}(m), m)i$,

$B_{12} = \text{rand}(m) \times a + \text{triu}(\text{rand}(m), m)i$,

$C_{12} = \text{eye}(m) \times a + \text{eye}(m)i$, $D_{12} = \text{eye}(m) \times a + \text{tril}(\text{rand}(m), m)i$,

$A_{21} = \text{diag}(a + \text{diag}(\text{rand}(m))) \times a - \text{triu}(\text{rand}(m), m)i$,

$B_{21} = \text{diag}(a + \text{diag}(\text{rand}(m))) + \text{tril}(\text{rand}(m), m)i$,

$C_{21} = -\text{eye}(m) \times a - \text{triu}(\text{rand}(m), m)i$, $D_{21} = \text{eye}(m) + \text{eye}(m)i$,

$A_{22} = -\text{rand}(m) + \text{eye}(m)i$, $B_{22} = \text{eye}(m) \times a + \text{tril}(\text{rand}(m), m)i$,

$C_{22} = -\text{eye}(m) - \text{tril}(\text{rand}(m), m)i$, $D_{22} = \text{rand}(m) - \text{triu}(\text{rand}(m), m)i$.

Let

$$
X_1 = \text{diag}(a + \text{diag}(\text{rand}(m))) + \text{eye}(m) \times a \\
+ \left(\text{tril}(\text{rand}(m), -1) \times a - \text{tril}(\text{rand}(m), -1)^T\right)i,
$$

$$
X_2 = \left(\text{tril}(\text{rand}(m), -1) + \text{triu}(\text{rand}(m)^T, 0)\right) \times a \\
+ \left(\text{tril}(\text{rand}(m), -1) - \text{tril}(\text{rand}(m), m)^T\right)i,
$$

then we have

$$
F_1 = A_{11}X_1B_{11} + C_{11}X_1D_{11} + A_{12}X_2B_{12} + C_{12}X_2D_{12},
$$

$$
F_2 = A_{21}X_1B_{21} + C_{21}X_1D_{21} + A_{22}X_2B_{22} + C_{22}X_2D_{22}.
$$

Therefore, the Eqs.(6.3) have a unique solution group $\{X_1, X_2\}$. In this example, we let $m = 10$ and $a = 10$. 
In addition, we choose \( X_1(0) = X_2(0) = X_1^2(0) = X_2^2(0) = 10^{-6} \times I_m, i = 1, 2 \) as the initial matrices and define the relative error as in Eq. (6.2).

In Figure 2, we compare the MRGI algorithm with the other methods including the GI algorithm in Wu et al. [24], the RGI algorithm in Huang and Ma [15] and the GRGI algorithm in Huang and Ma [15]. The relative error becomes smaller and smaller with the increase of iterative number \( k \), which indicates that the iterative solution is gradually approaching to the accurate solution and these algorithms are all effective. Meanwhile, it is illustrated that the MRGI algorithm is superior to the other three algorithms in convergence performance from Figure 2.

In order to evaluate the convergence performance of these algorithms, we compare iterative steps, relative residual and computational time results. The detailed information is given in Table 4. It can be clearly seen that the MRGI algorithm requires much fewer iteration steps and computational time than the GI algorithm in Wu et al. [24], the RGI algorithm in Huang and Ma [15] and the GRGI algorithm in Huang and Ma [15] to obtain iterative solution with smaller iterative error.

![Figure 2. Comparison of convergence curves](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>Steps</th>
<th>( f(k) )</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GI algorithm in Wu et al. [24]</td>
<td>15822</td>
<td>( 3.6900 \times 10^{-2} )</td>
<td>13.1977</td>
</tr>
<tr>
<td>RGI algorithm in Huang and Ma [15]</td>
<td>11948</td>
<td>( 1.2800 \times 10^{-2} )</td>
<td>9.9247</td>
</tr>
<tr>
<td>GRGI algorithm in Huang and Ma [15]</td>
<td>7865</td>
<td>( 4.6000 \times 10^{-3} )</td>
<td>4.8349</td>
</tr>
<tr>
<td>MRGI algorithm in this paper</td>
<td>4207</td>
<td>( 9.4651 \times 10^{-4} )</td>
<td>3.6684</td>
</tr>
</tbody>
</table>

According to Theorem 4.1, the MRGI algorithm is convergent if we choose \( 0 < \mu_1 < 1.1090 \times 10^{-5}, 0 < \mu_2 < 1.8842 \times 10^{-6} \) and \( \omega_1 = 0.4, \omega_2 = 0.6 \). However, through continuously attempting and testing, we get that the MRGI algorithm is also convergent if we choose \( 0 < \mu_1 < 1.7327 \times 10^{-5}, 0 < \mu_2 < 2.9676 \times 10^{-6} \) when...
we take $\omega_1 = 0.4, \omega_2 = 0.6$. Besides, it is clear to get that the convergence speed of MRGI algorithm becomes faster if we choose the larger convergence factors $\mu_1$ and $\mu_2$ in Figure 3. This situation illustrates that the convergence range of the convergence factors $\mu_1$ and $\mu_2$ calculated by Theorem 4.1 is a little conservative. How to reduce or remove this conservatism is our future work.

![Figure 3. The convergence performance of MRGI for different convergence factors $\mu_1$ and $\mu_2$ ($\omega_1 = 0.4, \omega_2 = 0.6$)](image)

### 7. Application in antilinear system

Consider the following discrete-time antilinear system

\[
\begin{align*}
\dot{x} &= Ax + Bu + Pw \\
\dot{w} &= Fw \\
e &=Cx + Qw,
\end{align*}
\]

(7.1)

where $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times r}$, $F \in \mathbb{C}^{p \times p}$, $P \in \mathbb{C}^{n \times p}$, $C \in \mathbb{C}^{m \times n}$ and $Q \in \mathbb{C}^{m \times p}$ are constant matrices, $x \in \mathbb{C}^n$, $u \in \mathbb{C}^r$ and $e \in \mathbb{C}^m$ are the state, the control input and the measurable error output, respectively. The symbol $w \in \mathbb{C}^p$ is the exogenous input that includes “reference signals to be tracked” and/or “disturbances to be rejected”. If we assume that $(A, B)$ is controllable, then $F$ is critical stable. If the full information feedback $u = -Kx + Lw$ is applied on the system, then the closed-loop system can result in

\[
\begin{align*}
\dot{x} &= (A - BK)x + (P + BL)w \\
\dot{w} &= Fw \\
e &=Cx + Qw.
\end{align*}
\]

(7.2)
The aim of the output regulation problem is to find two matrices $K$ and $L$ such that the matrix $A - B\overline{K}$ is stable and

$$\lim_{t \to \infty} e(t) = \lim_{t \to \infty} (C_x(t) + Qw(t)) = 0,$$

(7.3)

in which $(x(t), w(t))$ is arbitrary and $(x(0), w(0)) \in \mathbb{R}^n \times \mathbb{R}^p$. It has been shown that such a problem is solvable if and only if there exist two matrices $X$ and $Y$ such that

$$\begin{cases}
A\overline{X} - XF = BY + R \\
CX + Q = 0.
\end{cases}$$

(7.4)

If the parameter matrices are given as follows

$$A = \begin{bmatrix}
1 - i & 3 + 2i & -1 + i & 5 - i \\
2 + 2i & 1 & 1 - 8i & -3 + 8i \\
i & 2 - i & 5 + i & -6 - 4i \\
2 - 6i & 3 + i & 7 - i & -6 + i
\end{bmatrix},
B = \begin{bmatrix}
-1 + i & 1 - 4i & 2 + i & 2 - 5i \\
1 - 2i & -3 + i & 6 - i & -1 \\
1 + 2i & 2 - 3i & 4 - 7i & 3 + 2i \\
-3 - i & 5 + i & -9 + 2i & -1 - 6i
\end{bmatrix},
$$

$$Q = \begin{bmatrix}
212.63 + 65.46i & 92.9 + 400.77i & 86.42 - 415.58i & 44.51 - 69.96i \\
-23.93 - 17.08i & 95.27 - 165.11i & -126 + 227.46i & -12.23 + 25i \\
193.17 - 17.85i & 140.69 - 203.12i & -96.65 + 181.17i & 32.52 - 75.38i \\
60.04 + 40.57i & 220.23 - 15.59i & -182.93 + 159.68i & 2.23 + 52.4i
\end{bmatrix},
$$

$$C = \begin{bmatrix}
1.32 - 9.05i & -2.08 + 9.53i & -7.25 + 6.93i & -9.37 - 1.04i \\
-3.35 + 2.78i & 2.07 - 7.89i & 3.98 - 0.54i & 2.37 + 2.96i \\
-2.76 + 5.37i & 0.31 - 5.39i & 0.17 + 6.39i & 6.27i \\
-6.32 - 0.82i & 2.95 - 9.02i & 3.64 + 1.03i & -2.75 + 1.94i
\end{bmatrix},
$$

$$F = \begin{bmatrix}
2 - i & 9 + 3i & -5 - 2i & -7 \\
4 + 3i & -1 + 5i & 2 - 7i & 2 + i \\
-7 + 3i & -4 + 5i & 3 - 8i & -2 - 5i \\
5 & -8 + i & 5 - 4i & -7i
\end{bmatrix},
$$

$$R = \begin{bmatrix}
-69.57 - 95.78i & 74.83 - 187.11i & 125.03 + 222.47i & 2.18 - 49.19i \\
-242.29 + 73.82i & 105.29 - 26.16i & -126.56 + 7.04i & 144.05 + 142.32i \\
-173.98 - 120.99i & 90.91 - 59.56i & -114.88 - 21.57i & 54.15 + 27i \\
-15.39 - 378.98i & 2.9 - 60.71i & 89.17 + 365.53i & 252.51 - 46.41i
\end{bmatrix}.$$
By the Algorithm 3.4, the solutions to the matrix equation above are

\[
X = \begin{bmatrix}
1.2564 + 10.3357i & 23.9801 - 7.3301i \\
10.2329 - 2.0192i & -1.2398 - 0.8694i \\
-2.3726 + 23.0946i & 0.1973 + 2.1073i \\
17.0087 - 3.4275i & 7.3491 + 15.2093i \\
-14.2399 + 0.1275i & 1.7629 + 3.6928i \\
21.3982 + 0.7638i & 2.6382 - 1.9804i \\
2.0633 - 0.2743i & 9.0853 - 2.7734i \\
-0.3807 - 7.2063i & 5.3204 + 2.7493i
\end{bmatrix}
\]

\[
Y = \begin{bmatrix}
5.3502 - 1.2194i & -12.8920 - 3.7504i \\
-1.2957 + 0.3721i & 0.2975 - 13.7636i \\
17.9502 + 4.8529i & 3.8704 - 0.7697i \\
-13.9402 - 7.4028i & 0.3967 - 5.6428i \\
2.5307 + 27.3651i & 24.5803 - 5.7294i \\
-2.5497 + 4.0927i & -1.3529 - 9.4325i \\
-2.7846 + 16.3397i & 0.4819 - 3.9052i \\
4.7829 - 7.0438i & -7.2906 - 4.3951i
\end{bmatrix}
\]

8. Conclusion remarks

This work has constructed a modified relaxed gradient based iterative (MRGI) algorithm for solving the coupled Sylvester conjugate matrix equations (CSCMEs). Combining the real representation and the vec-operator of complex matrix, the convergence analysis of MRGI algorithm is analyzed. The numerical experiments are offered to illustrate that the method presented in this paper has better convergence performance and requires less storage capacity than the other existing iterative methods \[15, 24\]. The method adopted in this paper can be applied to study the general or constraint solutions of discrete-time periodic matrix equations \[8, 9\].

References


A novel iterative method for solving the CSCMEs


