Journal of Applied Analysis and Computation
Volume 13, Number 1, February 2023, 249–274

Website:http://www.jaac-online.com
DOI:10.11948/20220032

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

Wenli Wang[1] and Caiqin Song[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.

MSC(2010) 15A24, 15A21, 65F10.

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

$$x(k+1) = A_{r(k)}x(k), \quad x(0) = x_0, \quad r(0) = r_0, \tag{1.1}$$

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

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

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

†The corresponding author. Email: songcaiqin1983@163.com(C. Song)
[1]School of Mathematical Science, University of Jinan, Jinan 250022, China
[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]

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

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]

$$\sum_{l=1}^{s_1}(A_lXB_l)+\sum_{l=1}^{s_2}(C_l\overline{X}D_l)+\sum_{l=1}^{s_3}(G_lX^TH_l)+\sum_{l=1}^{s_4}(M_lX^HN_l)=F, \tag{1.5}$$

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

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

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

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

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 [1–3,5,10,11,13,17,21] 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)

$$\sum_{j=1}^{p}(A_{ij}X_jB_{ij}+C_{ij}\overline{X_j}D_{ij})=F_i, i\in I[1,q], \tag{1.8}$$

where $A_{ij}, C_{ij} \in C^{m_i\times l_j}$, $B_{ij}, D_{ij} \in C^{n_j\times p_i}$, $F_i \in C^{m_i\times p_i}, i \in I[1,q], j \in I[1,p]$ 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

$$\sum_{j=1}^{s_{i1}}A_{i1j}X_1B_{i1j}+\sum_{j=1}^{w_{i1}}C_{i1j}\overline{X_1}D_{i1j}+\cdots+\sum_{j=1}^{s_{ip}}A_{ipj}X_pB_{ipj}+\sum_{j=1}^{w_{ip}}C_{ipj}\overline{X_p}D_{ipj}=F_i, \tag{1.9}$$

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 $A \in \mathcal{C}^{n\times n}$, 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 $\sigma_{max}(A), \sigma_{min}(A), \lambda_{max}(A), \lambda_{min}(A)$ 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 $A\otimes B$ to denote the Kronecker product of two matrices $A \in \mathcal{C}^{n\times m}$ and $B \in \mathcal{C}^{p\times q}$. For a matrix $X = (x_1, x_2, \cdots, x_n) \in \mathcal{C}^{m\times n}$, the vector stretching function $vec(\cdot) : X \to vec(X)$ 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 $\langle A, B\rangle = tr(A^HB)$ with $A, B \in \mathcal{C}^{n\times m}$. The spectral norm of the matrix A is denoted by $\|A\|_2 = \sqrt{\lambda_{max}(A^HA)} = \sigma_{max}(A)$ and the Frobenious norm of the matrix A is denoted by $\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{tr(A^HA)}$. The symbol I_n represents the identity matrix of size $n \times n$ and the symbol $rand(m)$, $diag(A)$, $tril(A)$, $triu(A)$, $eye(A)$ 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 $A \in \mathcal{C}^{m\times n}$ be an arbitrary complex matrix, then A can be uniquely decomposed into the form of $A_1 + A_2 i$ with $A_1, A_2 \in \mathcal{R}^{m\times n}$. Now we are in a position to define the real representation of a complex matrix A as

$$A_\sigma := \begin{bmatrix} A_1 & A_2 \\ A_2 & -A_1 \end{bmatrix} \in \mathcal{R}^{2m\times 2n}. \tag{2.1}$$

Then, let

$$\overline{A}_\sigma := (\overline{A})_\sigma, A_\sigma^T := (A^T)_\sigma, A_\sigma^H := (A^H)_\sigma, \tag{2.2}$$

and

$$P_j := \begin{bmatrix} I_j & 0 \\ 0 & -I_j \end{bmatrix}, Q_j := \begin{bmatrix} 0 & I_j \\ -I_j & 0 \end{bmatrix}, \tag{2.3}$$

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 $A, B \in C^{m\times n}, \alpha \in R$, *then*

$$\begin{cases} (A+B)_\sigma = A_\sigma + B_\sigma, \\ (\alpha A)_\sigma = \alpha A_\sigma, \\ P_m A_\sigma P_n = (\overline{A})_\sigma; \end{cases}$$

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

$$\begin{cases} (AB)_\sigma = A_\sigma P_n B_\sigma = A_\sigma \overline{B}_\sigma P_r, \\ (ABC)_\sigma = A_\sigma \overline{B}_\sigma C_\sigma, \\ Q_m A_\sigma Q_n = A_\sigma; \end{cases}$$

(3) If $A \in \mathcal{C}^{m\times n}$, *then*

$$\begin{cases} (A^T)_\sigma = (A_\sigma)^T, \\ (A^H)_\sigma = P_n (A^T)_\sigma P_m; \end{cases}$$

$$\begin{cases} P_n (A^H)_\sigma = \Big(P_m A_\sigma\Big)^T, \\ (A^H)_\sigma P_m = \Big(A_\sigma P_n\Big)^T. \end{cases}$$

Lemma 2.2 ([26]). *Given a complex matrix* A *with appropriate dimensions, the following relations hold.*

(1) $\|A_\sigma\|_F^2 = 2\|A\|_F^2$;

(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*

$$X(k+1) = X(k) + \mu A^H (F - AX(k)B)B^H, \tag{2.4}$$

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

$$0 \le \mu \le \frac{2}{\|A\|_2^2 \|B\|_2^2}. \tag{2.5}$$

Moreover, the best convergence factor μ_0 is

$$\mu_0 = \frac{2}{\lambda_{\max}(A^H A)\lambda_{\max}(B^H B) + \lambda_{\min}(A^H A)\lambda_{\min}(B^H B)}. \tag{2.6}$$

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

$$\min_{0<\mu<\frac{2}{m_{\max}}} \max_{1\le i\le n} \left|1 - \mu m_i\right| = \frac{m_{\max} - m_{\min}}{m_{\max} + m_{\min}}. \tag{2.7}$$

Moreover, the unique μ_{opt} such that this relation holds is

$$\mu_{opt} = \frac{2}{m_{\max} + m_{\min}}. \tag{2.8}$$

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

$$\Phi_{il} := F_i - \sum_{j=1}^{p}(A_{ij}X_jB_{ij} + C_{ij}\overline{X_j}D_{ij}) + A_{il}X_lB_{il}, i \in I[1,q], l \in I[1,p], \tag{3.1}$$

$$\Omega_{il} := \overline{F_i - \sum_{j=1}^{p}(A_{ij}X_jB_{ij} + C_{ij}\overline{X_j}D_{ij}) + C_{il}\overline{X_l}D_{il}}, i \in I[1,q], l \in I[1,p]. \tag{3.2}$$

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

$$A_{il}X_lB_{il} = \Phi_{il}, i \in I[1,q], l \in I[1,p], \tag{3.3}$$

$$\overline{C_{il}\overline{X_l}D_{il}} = \Omega_{il}, i \in I[1,q], l \in I[1,p]. \tag{3.4}$$

From Lemma 2.3, we can construct the recursive forms as follows,

$$X_l^{1,i}(k+1) = X_l(k) + \mu_i A_{il}^H(\Phi_{il} - A_{il}X_l(k)B_{il})B_{il}^H, i \in I[1,q], l \in I[1,p], \quad (3.5)$$

$$X_l^{2,i}(k+1) = X_l(k) + \mu_i \overline{C_{il}}^H(\Omega_{il} - \overline{C_{il}\overline{X_l(k)}D_{il}})\overline{D_{il}}^H, i \in I[1,q], l \in I[1,p]. \quad (3.6)$$

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

$$\begin{aligned} X_l^{1,i}(k+1) =& X_l(k) + \mu_i A_{il}^H\Bigg[F_i - \sum_{j=1}^{p}(A_{ij}X_jB_{ij} + C_{ij}\overline{X_j}D_{ij}) + A_{il}X_lB_{il} \\ & - A_{il}X_l(k)B_{il}\Bigg]B_{il}^H, i \in I[1,q], l \in I[1,p], \end{aligned} \quad (3.7)$$

$$\begin{aligned} X_l^{2,i}(k+1) =& X_l(k) + \mu_i \overline{C_{il}}^H\Bigg[\overline{F_i - \sum_{j=1}^{p}(A_{ij}X_jB_{ij} + C_{ij}\overline{X_j}D_{ij})} + \overline{C_{il}\overline{X_l}D_{il}} \\ & - \overline{C_{il}\overline{X_l(k)}D_{il}}\Bigg]\overline{D_{il}}^H, i \in I[1,q], l \in I[1,p]. \end{aligned} \quad (3.8)$$

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 $X_l(k)$, $l \in I[1,p]$. In this way, one can obtain the following iterative forms for $i \in I[1,q], l \in I[1,p]$,

$$X_l^{1,i}(k+1) = X_l(k) + \mu_i A_{il}^H\left[F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})\right]B_{il}^H, \quad (3.9)$$

$$X_l^{2,i}(k+1) = X_l(k) + \mu_i \overline{C_{il}}^H\left[\overline{F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})}\right]\overline{D_{il}}^H. \quad (3.10)$$

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

$$\begin{aligned} X_l^i(k+1) =& \frac{X_l^{1,i}(k+1) + X_l^{2,i}(k+1)}{2} \\ =& X_l(k) + \frac{\mu_i}{2}A_{il}^H\left[F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})\right]B_{il}^H + \frac{\mu_i}{2}\overline{C_{il}}^H \\ & \times \left[\overline{F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})}\right]\overline{D_{il}}^H, i \in I[1,q], l \in I[1,p]. \end{aligned} \quad (3.11)$$

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 $0 < \omega_i < 1$.

$$X_l(k+1) = \omega_1 X_l^1(k+1) + \omega_2 X_l^2(k+1) + \cdots + \omega_q X_l^q(k+1), l \in I[1,p]. \quad (3.12)$$

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

Algorithm 3.1.([24, The gradient based iterative (GI) algorithm])

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

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

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

$$\begin{aligned} X_l(k+1) = X_l(k) &+ \tfrac{\mu}{2q}\sum_{i=1}^{q} A_{il}^H\left[F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})\right]B_{il}^H \\ &+ \tfrac{\mu}{2q}\sum_{i=1}^{q}\overline{C_{il}}^H\overline{\left[F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})\right]}\overline{D_{il}}^H; \end{aligned}$$

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

Algorithm 3.2. ([15, The relaxed gradient based iterative (RGI) algorithm])

Step 1. Input matrices $A_{ij}, C_{ij} \in C^{m_i \times l_j}$,$B_{ij}, D_{ij} \in C^{n_j \times p_i}$, $F_i \in C^{m_i \times p_i}, i \in I[1,q], j \in I[1,p]$. Give any small positive number ε and appropriative positive numbers ω_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 $\delta_{k,i} = \frac{\|F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})\|}{\|F_i\|} < \varepsilon$, stop; otherwise, go to Step 4;

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

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

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

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. ([15, The generalized relaxed gradient based iterative (GRGI) algorithm)])

Step 1. Input matrices $A_{ij}, C_{ij} \in C^{m_i \times l_j}$,$B_{ij}, D_{ij} \in C^{n_j \times p_i}$, $F_i \in C^{m_i \times p_i}, i \in I[1,q], j \in I[1,p]$. Give any small positive number ε and appropriative positive numbers α_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 $X_l(0) = \omega_l X_l^{(1)}(0) + (1-\omega_l)X_l^{(2)}(0), l \in I[1,p]$, set $k := 1$;

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

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

$$\begin{aligned} X_l^{(1)}(k) = X_l^{(1)}(k-1) + \frac{\mu}{2}\Big\{ & A_{1l}^H\Big[F_1 - \sum_{j=1}^{p}(A_{1j}X_j(k-1)B_{1j} + C_{1j}\overline{X_j(k-1)}D_{1j})\Big] \\ & \times B_{1l}^H + C_{1l}^T \overline{\Big[F_1 - \sum_{j=1}^{p}(A_{1j}X_j(k-1)B_{1j} + C_{1j}\overline{X_j(k-1)}D_{1j})\Big]} D_{1l}^T \Big\}, \end{aligned}$$

$$\vdots \quad \vdots \quad \vdots$$

$$\begin{aligned} X_l^{(q)}(k) = X_l^{(q)}(k-1) + \frac{\mu}{2}\Big\{ & A_{ql}^H\Big[F_q - \sum_{j=1}^{p}(A_{qj}X_j(k-1)B_{qj} + C_{qj}\overline{X_j(k-1)}D_{qj})\Big] \\ & \times B_{ql}^H + C_{ql}^T \overline{\Big[F_q - \sum_{j=1}^{p}(A_{qj}X_j(k-1)B_{qj} + C_{qj}\overline{X_j(k-1)}D_{qj})\Big]} D_{ql}^T \Big\}. \end{aligned}$$

Compute $X_l(k) = \alpha_1 X_l^{(1)}(k) + \alpha_2 X_l^{(2)}(k) + \cdots + \alpha_q X_l^{(q)}(k), l \in I[1,p]$;

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

Summing up Eqs.(3.9)-(3.12), we get the following MRGI algorithm for solving CSCMEs (1.8).

Algorithm 3.4. (The modified relaxed gradient based iterative (MRGI) algorithm)

Step 1. Input matrices $A_{ij}, C_{ij} \in C^{m_i \times l_j}$,$B_{ij}, D_{ij} \in C^{n_j \times p_i}$, $F_i \in C^{m_i \times p_i}, i \in I[1,q], j \in I[1,p]$. Give any small positive number ε and appropriative positive numbers $\omega_i, i = 1,2,\ldots,p$ such that $\sum_{i=1}^{p}\omega_i = 1$. Choose the initial matrices $X_j(0), j \in I[1,p]$, set $k := 1$.

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

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

$$\begin{aligned} X_l^i(k+1) = X_l(k) + \frac{\mu_i}{2}A_{il}^H & \left[F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})\right]B_{il}^H \\ & + \frac{\mu_i}{2}\overline{C_{il}}^H \left[\overline{F_i - \sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j(k)}D_{ij})}\right]\overline{D_{il}}^H, \\ X_l(k+1) = \omega_1 X_l^1(k+1) + & \omega_2 X_l^2(k+1) + \cdots + \omega_q X_l^q(k+1); \end{aligned}$$

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

4. Convergence analysis

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

$$\mathcal{A} := \begin{pmatrix} \left(P_{n1}(B_{11})_\sigma\right)^T \otimes \left((A_{11})_\sigma P_{l1}\right) + \left(D_{11}\right)_\sigma^T \otimes \left(C_{11}\right)_\sigma & \cdots & \cdots & \cdots & \cdots & \left(P_{np}(B_{1p})_\sigma\right)^T \otimes \left((A_{1p})_\sigma P_{lp}\right) + \left(D_{1p}\right)_\sigma^T \otimes \left(C_{1p}\right)_\sigma \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(P_{n1}(B_{q1})_\sigma\right)^T \otimes \left((A_{q1})_\sigma P_{l1}\right) + \left(D_{q1}\right)_\sigma^T \otimes \left(C_{q1}\right)_\sigma & \cdots & \cdots & \cdots & \cdots & \left(P_{np}(B_{qp})_\sigma\right)^T \otimes \left((A_{qp})_\sigma P_{lp}\right) + \left(D_{qp}\right)_\sigma^T \otimes \left(C_{qp}\right)_\sigma \end{pmatrix}. \tag{4.1}$$

It is easy to see $\mathcal{A} \in R^{\sum_{i=1}^{q} 4m_ip_i \times \sum_{j=1}^{p} 4n_jl_j}$. Then, we have the following results.

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

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

and the corresponding homogeneous matrix equations $\sum_{j=1}^{p}(A_{ij}X_jB_{ij}+C_{ij}\overline{X_j}D_{ij}) = 0, i \in I[1,q]$, 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

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

By using Kronecker products of matrices and vector stretching operator, the preceding expression can be transformed into $\mathcal{A}x = f$ with

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

where $\mathcal{A}$ is given by (4.1). Therefore, the CSCMEs (1.8) have unique solutions if and only if the matrix $\mathcal{A}$ is nonsingular. The conclusion follows immediately. $\square$

Theorem 4.1. *Suppose that the CSCMEs* (1.8) *have a unique solution* $(X_1^*, X_2^*, \cdots, X_p^*)$. *If* μ_i *satisfies*

$$0 < \mu_i < \frac{4}{q\omega_i \sum_{l=1}^{p} (\|A_{il}\|_2^2 \|B_{il}\|_2^2 + \|C_{il}\|_2^2 \|D_{il}\|_2^2)}, \tag{4.5}$$

then the iterative sequences $X_l(k), l \in I[1,p]$ *generated by the Algorithm 3.4 converge to* $(X_1^*, X_2^*, \cdots, X_p^*)$, *i.e.,* $\lim_{k\to\infty} X_l(k) = X_l^*, l \in I[1,p]$ *or the error matrices* $X_l(k) - X_l^*, l \in I[1,p]$ *converge to zero for any initial values* $X_l(0), l \in I[1,p]$, *where* $(\cdot)_\sigma$ *is the real representation of complex matrix.*

Proof. Define the error matrices

$$\widetilde{X}_l(k) := X_l(k) - X_l^*, l \in I[1,p], \tag{4.6}$$

and

$$\theta_i(k) := F_i - \sum_{j=1}^{p} (A_{ij} X_j(k) B_{ij} + C_{ij} \overline{X_j(k)} D_{ij}), i \in I[1,q]. \tag{4.7}$$

From Eq.(4.7), it is easy to derive

$$\theta_i(k) = -\sum_{j=1}^{p} (A_{ij} \widetilde{X}_j(k) B_{ij} + C_{ij} \overline{\widetilde{X}_j(k)} D_{ij}), i \in I[1,q]. \tag{4.8}$$

It follows from Algorithm 3.4 that

$$\begin{aligned}
\widetilde{X}_l(k+1) &= X_l(k+1) - X_l^* = \sum_{i=1}^{q} \omega_i X_l^i(k+1) - X_l^* = \sum_{i=1}^{q} \omega_i [X_l^i(k+1) - X_l^*] \\
&= \sum_{i=1}^{q} \omega_i \frac{X_l^{1,i}(k+1) - X_l^* + X_l^{2,i}(k+1) - X_l^*}{2} \\
&= \sum_{i=1}^{q} \omega_i \frac{2\widetilde{X}_l(k) + \mu_i (A_{il}^H \theta_i(k) B_{il}^H + \overline{C_{il}}^H \overline{\theta_i(k)} \overline{D_{il}}^H)}{2} \\
&= \widetilde{X}_l(k) + \sum_{i=1}^{q} \omega_i \mu_i \frac{A_{il}^H \theta_i(k) B_{il}^H + \overline{C_{il}}^H \overline{\theta_i(k)} \overline{D_{il}}^H}{2}.
\end{aligned} \tag{4.9}$$

Thus, by the properties of norm, one has

$$\begin{aligned}
\|\widetilde{X}_l(k+1)\|_F^2 =& \Big\| \widetilde{X}_l(k) + \sum_{i=1}^{q} \omega_i \mu_i \frac{A_{il}^H \theta_i(k) B_{il}^H + \overline{C_{il}}^H \overline{\theta_i(k)} \overline{D_{il}}^H}{2} \Big\|_F^2 \\
=& \|\widetilde{X}_l(k)\|_F^2 + \Big\| \sum_{i=1}^{q} \omega_i \mu_i \frac{A_{il}^H \theta_i(k) B_{il}^H + \overline{C_{il}}^H \overline{\theta_i(k)} \overline{D_{il}}^H}{2} \Big\|_F^2 \\
&+ tr\Big[\widetilde{X}_l(k)^H \sum_{i=1}^{q} (\omega_i \mu_i \frac{A_{il}^H \theta_i(k) B_{il}^H + \overline{C_{il}}^H \overline{\theta_i(k)} \overline{D_{il}}^H}{2}) \Big]
\end{aligned}$$

$$
\begin{aligned}
&+tr\Big[\sum_{i=1}^{q}(\omega_i\mu_i\frac{B_{il}\theta_i(k)^H A_{il}+\overline{D_{il}}\,\overline{\theta_i(k)}^H\overline{C_{il}}}{2})\widetilde{X}_l(k)\Big]\\
=&\|\widetilde{X}_l(k)\|_F^2+\Big\|\sum_{i=1}^{q}\omega_i\mu_i\frac{A_{il}^H\theta_i(k)B_{il}^H+\overline{C_{il}}^H\overline{\theta_i(k)}\,\overline{D_{il}}^H}{2}\Big\|_F^2\\
&+\frac{1}{2}tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\widetilde{X}_l(k)^H A_{il}^H\theta_i(k)B_{il}^H\Big)\\
&+\frac{1}{2}tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\widetilde{X}_l(k)^H\overline{C_{il}}^H\overline{\theta_i(k)}\,\overline{D_{il}}^H\Big)\\
&+\frac{1}{2}tr\Big(\sum_{i=1}^{q}\omega_i\mu_i B_{il}\theta_i(k)^H A_{il}\widetilde{X}_l(k)\Big)\\
&+\frac{1}{2}tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\overline{D_{il}}\,\overline{\theta_i(k)}^H\overline{C_{il}}\widetilde{X}_l(k)\Big).
\end{aligned}
\tag{4.10}
$$

Moreover, note that the following expression is real,

$$
tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\widetilde{X}_l(k)^H\overline{C_{il}}^H\overline{\theta_i(k)}\,\overline{D_{il}}^H\Big)+tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\overline{D_{il}}\,\overline{\theta_i(k)}^H\overline{C_{il}}\widetilde{X}_l(k)\Big).
\tag{4.11}
$$

Therefore, we get

$$
\begin{aligned}
&tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\widetilde{X}_l(k)^H\overline{C_{il}}^H\overline{\theta_i(k)}\,\overline{D_{il}}^H\Big)+tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\overline{D_{il}}\,\overline{\theta_i(k)}^H\overline{C_{il}}\widetilde{X}_l(k)\Big)\\
=&tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\overline{\widetilde{X}_l(k)}^H C_{il}^H\theta_i(k)D_{il}^H\Big)+tr\Big(\sum_{i=1}^{q}\omega_i\mu_i D_{il}\theta_i(k)^H C_{il}\overline{\widetilde{X}_l(k)}\Big),
\end{aligned}
\tag{4.12}
$$

and

$$
\begin{aligned}
&tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\widetilde{X}_l(k)^H A_{il}^H\theta_i(k)B_{il}^H\Big)+tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\widetilde{X}_l(k)^H\overline{C_{il}}^H\overline{\theta_i(k)}\,\overline{D_{il}}^H\Big)\\
&+tr\Big(\sum_{i=1}^{q}\omega_i\mu_i B_{il}\theta_i(k)^H A_{il}\widetilde{X}_l(k)\Big)+tr\Big(\sum_{i=1}^{q}\omega_i\mu_i\overline{D_{il}}\,\overline{\theta_i(k)}^H\overline{C_{il}}\widetilde{X}_l(k)\Big)\\
=&tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\Big(B_{il}^H\widetilde{X}_l(k)^H A_{il}^H+D_{il}^H\overline{\widetilde{X}_l(k)}^H C_{il}^H\Big)\theta_i(k)\Big]\\
&+tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\theta_i(k)^H\Big(A_{il}\widetilde{X}_l(k)B_{il}+C_{il}\overline{\widetilde{X}_l(k)}D_{il}\Big)\Big].
\end{aligned}
\tag{4.13}
$$

So Eq.(4.10) becomes

$$
\begin{aligned}
&\|\widetilde{X}_l(k+1)\|_F^2\\
=&\|\widetilde{X}_l(k)\|_F^2+\Big\|\sum_{i=1}^{q}\omega_i\mu_i\frac{A_{il}^H\theta_i(k)B_{il}^H+\overline{C_{il}}^H\overline{\theta_i(k)}\,\overline{D_{il}}^H}{2}\Big\|_F^2
\end{aligned}
$$

$$+\frac{1}{2}tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\Big(B_{il}^H\widetilde{X}_l(k)^H A_{il}^H + D_{il}^H\overline{\widetilde{X}_l(k)}^H C_{il}^H\Big)\theta_i(k)\Big]$$
$$+\frac{1}{2}tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\theta_i(k)^H\Big(A_{il}\widetilde{X}_l(k)B_{il} + C_{il}\overline{\widetilde{X}_l(k)}D_{il}\Big)\Big]. \tag{4.14}$$

Adding all $||\widetilde{X}_l(k+1)||_F^2, l \in I[1,q]$ on both sides of Eq.(4.14), we have

$$\sum_{l=1}^{p}\|\widetilde{X}_l(k+1)\|_F^2$$
$$=\sum_{l=1}^{p}\|\widetilde{X}_l(k)\|_F^2 + \sum_{l=1}^{p}\|\sum_{i=1}^{q}\omega_i\mu_i\frac{A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H}{2}\|_F^2$$
$$+\frac{1}{2}\sum_{l=1}^{p}tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\Big(B_{il}^H\widetilde{X}_l(k)^H A_{il}^H + D_{il}^H\overline{\widetilde{X}_l(k)}^H C_{il}^H\Big)\theta_i(k)\Big]$$
$$+\frac{1}{2}\sum_{l=1}^{p}tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\theta_i(k)^H\Big(A_{il}\widetilde{X}_l(k)B_{il} + C_{il}\overline{\widetilde{X}_l(k)}D_{il}\Big)\Big]. \tag{4.15}$$

It is easy to know that

$$\Big\|\sum_{i=1}^{q}\omega_i\mu_i\frac{A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H}{2}\Big\|_F^2$$
$$=\sum_{i=1}^{q}\frac{\omega_i^2\mu_i^2}{4}\Big\|A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F^2$$
$$+\sum_{1\le i\ne j\le q}\frac{\omega_i\mu_i\omega_j\mu_j}{4}tr\Big[\Big(A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big)^H$$
$$\times(A_{jl}^H\theta_j(k)B_{jl}^H + \overline{C_{jl}}^H\overline{\theta_j(k)}\overline{D_{jl}}^H)\Big]$$
$$=\sum_{i=1}^{q}\frac{\omega_i^2\mu_i^2}{4}\Big\|A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F^2$$
$$+\sum_{1\le i\ne j\le q}\frac{\omega_i\mu_i\omega_j\mu_j}{4}\langle A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H,$$
$$A_{jl}^H\theta_j(k)B_{jl}^H + \overline{C_{jl}}^H\overline{\theta_j(k)}\overline{D_{jl}}^H\rangle$$
$$\le\sum_{i=1}^{q}\frac{\omega_i^2\mu_i^2}{4}\Big\|A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F^2$$
$$+\sum_{1\le i\ne j\le q}\frac{\omega_i\mu_i\omega_j\mu_j}{4}\Big\|A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F$$
$$\times\Big\|A_{jl}^H\theta_j(k)B_{jl}^H + \overline{C_{jl}}^H\overline{\theta_j(k)}\overline{D_{jl}}^H\Big\|_F$$
$$\le\sum_{i=1}^{q}\frac{\omega_i^2\mu_i^2}{4}\Big\|A_{il}^H\theta_i(k)B_{il}^H + \overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F^2 + \sum_{1\le i\ne j\le q}\Big[\frac{\omega_i^2\mu_i^2}{8}\Big\|A_{il}^H\theta_i(k)B_{il}^H$$

$$+\overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F^2+\frac{\omega_j^2\mu_j^2}{8}\Big\|A_{jl}^H\theta_j(k)B_{jl}^H+\overline{C_{jl}}^H\overline{\theta_j(k)}\overline{D_{jl}}^H\Big\|_F^2\Big]$$

$$=\frac{q}{4}\sum_{i=1}^{q}\omega_i^2\mu_i^2\Big\|A_{il}^H\theta_i(k)B_{il}^H+\overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F^2. \tag{4.16}$$

Moreover, we have

$$\begin{aligned}
&\Big\|A_{il}^H\theta_i(k)B_{il}^H+\overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H\Big\|_F^2\\
=&\frac{1}{2}\Big\|(A_{il}^H\theta_i(k)B_{il}^H+\overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H)_\sigma\Big\|_F^2\\
=&\frac{1}{2}\Big\|(A_{il}^H)_\sigma P_{mi}(\theta_i(k))_\sigma P_{pi}(B_{il}^H)_\sigma+(\overline{C_{il}}^H)_\sigma(\theta_i(k))_\sigma(\overline{D_{il}}^H)_\sigma\Big\|_F^2\\
\le&\frac{1}{2}\Big(\|(A_{il}^H)_\sigma\|_2^2\|P_{mi}\|_2^2\|(\theta_i(k))_\sigma\|_F^2\|P_{pi}\|_2^2\|(B_{il}^H)_\sigma\|_2^2\\
&+\|(\overline{C_{il}}^H)_\sigma\|_2^2\|(\theta_i(k))_\sigma\|_F^2\|(\overline{D_{il}}^H)_\sigma\|_2^2\Big)\\
=&\frac{1}{2}\Big(2\|A_{il}^H\|_2^2\|\theta_i(k)\|_F^2\|B_{il}^H\|_2^2+2\|\overline{C_{il}}^H\|_2^2\|\theta_i(k)\|_F^2\|\overline{D_{il}}^H\|_2^2\Big)\\
=&\Big(\|A_{il}\|_2^2\|B_{il}\|_2^2+\|C_{il}\|_2^2\|D_{il}\|_2^2\Big)\|\theta_i(k)\|_F^2.
\end{aligned} \tag{4.17}$$

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

$$\begin{aligned}
&\Big\|\sum_{i=1}^{q}\omega_i\mu_i\frac{A_{il}^H\theta_i(k)B_{il}^H+\overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H}{2}\Big\|_F^2\\
\le&\frac{q}{4}\sum_{i=1}^{q}\omega_i^2\mu_i^2\Big(\|A_{il}\|_2^2\|B_{il}\|_2^2+\|C_{il}\|_2^2\|D_{il}\|_2^2\Big)\|\theta_i(k)\|_F^2.
\end{aligned} \tag{4.18}$$

From Eq.(4.18), one has

$$\begin{aligned}
&\sum_{l=1}^{p}\Big\|\sum_{i=1}^{q}\omega_i\mu_i\frac{A_{il}^H\theta_i(k)B_{il}^H+\overline{C_{il}}^H\overline{\theta_i(k)}\overline{D_{il}}^H}{2}\Big\|_F^2\\
\le&\frac{q}{4}\sum_{l=1}^{p}\sum_{i=1}^{q}\omega_i^2\mu_i^2\Big(\|A_{il}\|_2^2\|B_{il}\|_2^2+\|C_{il}\|_2^2\|D_{il}\|_2^2\Big)\|\theta_i(k)\|_F^2.
\end{aligned} \tag{4.19}$$

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

$$\begin{aligned}
&\sum_{l=1}^{p}\Big\|\widetilde{X}_l(k+1)\Big\|_F^2\\
\le&\sum_{l=1}^{p}\|\widetilde{X}_l(k)\|_F^2+\frac{q}{4}\sum_{l=1}^{p}\sum_{i=1}^{q}\omega_i^2\mu_i^2\Big(\|A_{il}\|_2^2\|B_{il}\|_2^2+\|C_{il}\|_2^2\|D_{il}\|_2^2\Big)\|\theta_i(k)\|_F^2\\
&+\frac{1}{2}\sum_{l=1}^{p}tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\Big(B_{il}^H\widetilde{X}_l(k)^HA_{il}^H+D_{il}^H\overline{\widetilde{X}_l(k)}^HC_{il}^H\Big)\theta_i(k)\Big]\\
&+\frac{1}{2}\sum_{l=1}^{p}tr\Big[\sum_{i=1}^{q}\omega_i\mu_i\theta_i(k)^H\Big(A_{il}\widetilde{X}_l(k)B_{il}+C_{il}\overline{\widetilde{X}_l(k)}D_{il}\Big)\Big]
\end{aligned}$$

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

Thus, if the convergence factors are chosen to satisfy Eq.(4.5), then for any initial values $X_l(0), l \in I[1,p]$, 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

$$\lim_{k\to\infty}\sum_{j=1}^{p}(A_{ij}X_j(k)B_{ij} + C_{ij}\overline{X_j}(k)D_{ij}) = F_i, i \in I[1,q]. \tag{4.22}$$

It follows that

$$\lim_{k\to\infty}X_l(k) = X_l^*, l \in I[1,p]. \tag{4.23}$$

So the desire result follows. □

In the following, we study the necessary and sufficient conditions of convergence for Algorithm 3.1 and consider the optimal μ 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*

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

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

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

where the matrix $\mathcal{A}$ *is defined in Eq.*(4.1).

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

$$\begin{aligned}\widetilde{X}_l(k+1) &= \widetilde{X}_l(k) - \frac{\mu}{2q}\sum_{i=1}^{q}\Big[A_{il}^H \sum_{j=1}^{p}\Big(A_{ij}\widetilde{X}_j(k)B_{ij} + C_{ij}\overline{\widetilde{X}_j(k)}D_{ij}\Big)B_{il}^H \\ &\quad + C_{il}^T\sum_{j=1}^{p}\overline{\Big(A_{ij}\widetilde{X}_j(k)B_{ij} + C_{ij}\overline{\widetilde{X}_j(k)}D_{ij}\Big)}D_{il}^T\Big] \\ &= \widetilde{X}_l(k) - \frac{\mu}{2q}\sum_{i=1}^{q}\sum_{j=1}^{p}\Big[A_{il}^H\Big(A_{ij}\widetilde{X}_j(k)B_{ij} + C_{ij}\overline{\widetilde{X}_j(k)}D_{ij}\Big)B_{il}^H \\ &\quad + C_{il}^T\overline{\Big(A_{ij}\widetilde{X}_j(k)B_{ij} + C_{ij}\overline{\widetilde{X}_j(k)}D_{ij}\Big)}D_{il}^T\Big],\end{aligned} \tag{4.26}$$

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

$$\begin{aligned}&\big(\widetilde{X}_l(k+1)\big)_\sigma \\ &= \big(\widetilde{X}_l(k)\big)_\sigma - \frac{\mu}{2q}\sum_{i=1}^{q}\sum_{j=1}^{p}\Big[(A_{il}^H)_\sigma P_{mi}\Big(A_{ij}\widetilde{X}_j(k)B_{ij} + C_{ij}\overline{\widetilde{X}_j(k)}D_{ij}\Big)_\sigma \\ &\quad \times P_{pi}(B_{il}^H)_\sigma + (C_{il}^T)_\sigma\Big(A_{ij}\widetilde{X}_j(k)B_{ij} + C_{ij}\overline{\widetilde{X}_j(k)}D_{ij}\Big)_\sigma (D_{il}^T)_\sigma\Big].\end{aligned} \tag{4.27}$$

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

$$\begin{aligned}&vec\Big[\big(\widetilde{X}_l(k+1)\big)_\sigma\Big] \\ &= vec\Big[\big(\widetilde{X}_l(k)\big)_\sigma\Big] - \sum_{i=1}^{q}\sum_{j=1}^{p}\frac{\mu}{2q}\Big[\big(P_{pi}(B_{il}^H)_\sigma\big)^T \otimes \big((A_{il}^H)_\sigma P_{mi}\big) \\ &\quad + \big((D_{il}^T)_\sigma\big)^T \otimes \big(C_{il}^T\big)_\sigma\Big] vec\Big[\Big(A_{ij}\widetilde{X}_j(k)B_{ij} + C_{ij}\overline{\widetilde{X}_j(k)}D_{ij}\Big)_\sigma\Big]\end{aligned}$$

$$= vec\Big[\big(\widetilde{X}_l(k)\big)_\sigma\Big] - \sum_{i=1}^{q}\sum_{j=1}^{p}\frac{\mu}{2q}\Big[\big(P_{nj}(B_{il})_\sigma\big)\otimes\big((A_{il})_\sigma P_{lj}\big)^T + (D_{il})_\sigma\otimes\big(C_{il}\big)_\sigma^T\Big]$$

$$\times\, vec\Big[(A_{ij})_\sigma P_{lj}(\widetilde{X}_j(k))_\sigma P_{nj}(B_{ij})_\sigma + (C_{ij})_\sigma(\widetilde{X}_j(k))_\sigma(D_{ij})_\sigma\Big]$$

$$= vec\Big[\big(\widetilde{X}_l(k)\big)_\sigma\Big] - \sum_{i=1}^{q}\sum_{j=1}^{p}\frac{\mu}{2q}\Big[\big(P_{nj}(B_{il})_\sigma\otimes\big((A_{il})_\sigma P_{lj}\big)^T + (D_{il})_\sigma\otimes\big(C_{il}\big)_\sigma^T\Big]$$

$$\times\Big[\big(P_{nj}(B_{ij})_\sigma\big)^T\otimes\big((A_{ij})_\sigma P_{lj}\big) + \big(D_{ij}\big)_\sigma^T\otimes\big(C_{ij}\big)_\sigma\Big] vec\Big[\big(\widetilde{X}_j(k)\big)_\sigma\Big]$$

$$= vec\Big[\big(\widetilde{X}_l(k)\big)_\sigma\Big] - \sum_{i=1}^{q}\sum_{j=1}^{p}\frac{\mu}{2q}\Big[\big(P_{nj}(B_{il})_\sigma\big)^T\otimes\big((A_{il})_\sigma P_{lj}\big) + \big(D_{il}\big)_\sigma^T\otimes\big(C_{il}\big)_\sigma\Big]^T$$

$$\times\Big[\big(P_{nj}(B_{ij})_\sigma\big)^T\otimes\big((A_{ij})_\sigma P_{lj}\big) + \big(D_{ij}\big)_\sigma^T\otimes\big(C_{ij}\big)_\sigma\Big] vec\Big[\big(\widetilde{X}_j(k)\big)_\sigma\Big]. \tag{4.28}$$

So we have

$$\begin{aligned} vec\Big[\widetilde{X}_\sigma(k+1)\Big] &= vec\Big[\widetilde{X}_\sigma(k)\Big] - \frac{\mu}{2q}\mathcal{A}^T\mathcal{A}vec\Big[\widetilde{X}_\sigma(k)\Big] \\ &= (I - \frac{\mu}{2q}\mathcal{A}^T\mathcal{A})vec\Big[\widetilde{X}_\sigma(k)\Big], \end{aligned} \tag{4.29}$$

where $vec\big[\widetilde{X}_\sigma(k)\big] = \Big[\Big[vec((\widetilde{X}_1(k))_\sigma)\Big]^T, \Big[vec((\widetilde{X}_2(k))_\sigma)\Big]^T, \cdots, \Big[vec((\widetilde{X}_p(k))_\sigma)\Big]^T\Big]^T$.

This equation is a linear matrix equation with the coefficient matrix $\Upsilon := I - \frac{\mu}{2q}\mathcal{A}^T\mathcal{A}$. Thus, the gradient based iterative algorithm converges for any initial matrices $X_l(0), l \in I[1,p]$ if and only if $\rho(\Upsilon) < 1$. Since $\mathcal{A}^T\mathcal{A}$ is a symmetric matrix, we have

$$\rho(\Upsilon) = \max_{1\le i\le\sum_{j=1}^{p}4n_jl_j}|\lambda_i(\Upsilon)| = \max_{1\le i\le\sum_{j=1}^{p}4n_jl_j}|1 - \frac{\mu}{2q}\lambda_i(\mathcal{A}^T\mathcal{A})| < 1. \tag{4.30}$$

Therefore, we have $0 < \mu < \frac{4q}{\sigma^2_{max}(\mathcal{A})}$. According to Lemma 2.4, if

$$\mu = \mu_{opt} = \frac{4q}{\sigma^2_{max}(\mathcal{A}) + \sigma^2_{min}(\mathcal{A})}, \tag{4.31}$$

then the maximal convergence rate of Algorithm 3.1 can be reached. □

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

$$\sum_{j=1}^{s_{i1}}A_{i1j}X_1B_{i1j} + \sum_{j=1}^{w_{i1}}C_{i1j}\overline{X_1}D_{i1j} + \cdots + \sum_{j=1}^{s_{ip}}A_{ipj}X_pB_{ipj} + \sum_{j=1}^{w_{ip}}C_{ipj}\overline{X_p}D_{ipj} = F_i, \tag{5.1}$$

where $A_{i\eta j}, C_{i\eta j} \in C^{m_i\times r_\eta}$, $B_{i\eta j}, D_{i\eta j} \in C^{s_\eta\times n_i}$, $i \in I[1,N], \eta \in I[1,p]$ 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,

$$
\begin{aligned}
X_l^{1,i}(k+1) =& X_l(k) + \mu_i \sum_{j=1}^{s_{il}} A_{ilj}^H \Big[F_i - \Big(\sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} A_{iut} X_u(k) B_{iut} \\
& + \sum_{u=1}^{p} \sum_{t=1}^{w_{iu}} C_{iut} \overline{X_u(k)} D_{iut} \Big) \Big] B_{ilj}^H, \qquad (5.2)\\
X_l^{2,i}(k+1) =& X_l(k) + \mu_i \sum_{j=1}^{w_{il}} \overline{C_{ilj}}^H \overline{\Big[F_i - \Big(\sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} A_{iut} X_u(k) B_{iut}} \\
& \overline{+ \sum_{u=1}^{p} \sum_{t=1}^{w_{iu}} C_{iut} \overline{X_u(k)} D_{iut} \Big) \Big]} \overline{D_{ilj}}^H, \qquad (5.3)\\
X_l^{i}(k+1) =& \frac{X_l^{1,i}(k+1) + X_l^{2,i}(k+1)}{2} \\
=& X_l(k) + \frac{\mu_i}{2} \sum_{j=1}^{s_{il}} A_{ilj}^H \Big[F_i - \Big(\sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} A_{iut} X_u(k) B_{iut} \\
& + \sum_{u=1}^{p} \sum_{t=1}^{w_{iu}} C_{iut} \overline{X_u(k)} D_{iut} \Big) \Big] B_{ilj}^H \\
& + \frac{\mu_i}{2} \sum_{j=1}^{w_{il}} \overline{C_{ilj}}^H \overline{\Big[F_i - \Big(\sum_{u=1}^{p} \sum_{t=1}^{s_{iu}} A_{iut} X_u(k) B_{iut}} \\
& \overline{+ \sum_{u=1}^{p} \sum_{t=1}^{w_{iu}} C_{iut} \overline{X_u(k)} D_{iut} \Big) \Big]} \overline{D_{ilj}}^H, \qquad (5.4)\\
X_l(k+1) =& \omega_1 X_l^1(k+1) + \omega_2 X_l^2(k+1) + \cdots + \omega_q X_l^q(k+1), l \in I[1,p]. \qquad (5.5)
\end{aligned}
$$

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*

$$
0 < \mu_i < \frac{4}{q\omega_i \sum_{l=1}^{p} \Big(\sum_{j=1}^{s_{il}} \|A_{ilj}\|_F^2 \|B_{ilj}\|_F^2 + \sum_{j=1}^{w_{il}} \|C_{ilj}\|_F^2 \|D_{ilj}\|_F^2 \Big)}. \qquad (5.6)
$$

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-conjugate matrix equations

$$\begin{cases} A_{11}X_1B_{11} + C_{11}\overline{X_1}D_{11} + A_{12}X_2B_{12} + C_{12}\overline{X_2}D12 = F_1, \\ A_{21}X_1B_{21} + C_{21}\overline{X_1}D_{21} + A_{22}X_2B_{22} = F_2, \end{cases} \tag{6.1}$$

with the following coefficient matrices:

$$A_{11} = \begin{bmatrix} 2-2i & 2i \\ 8+i & 2+3i \end{bmatrix},\ B_{11} = \begin{bmatrix} 0.5-i & -1+3i \\ -1.5+2i & 1-2i \end{bmatrix},\ C_{11} = \begin{bmatrix} 4i & 2+2i \\ 3+2.5i & i \end{bmatrix},$$

$$C_{12} = \begin{bmatrix} 1+3i & 4-i \\ 2-3i & 1+2i \end{bmatrix}, D_{12} = \begin{bmatrix} 1+2.5i & -3+i \\ -1-0.5i & -1+2i \end{bmatrix}, D_{21} = \begin{bmatrix} 4-i & 1.5-i \\ i & -2+2i \end{bmatrix},$$

$$D_{11} = \begin{bmatrix} -2+i & 3i \\ -1 & 3+i \end{bmatrix},\ A_{12} = \begin{bmatrix} 1-1.5i & 3i \\ -2+3i & 4 \end{bmatrix},\ B_{12} = \begin{bmatrix} 1-2i & -1+4i \\ -1+3i & 1 \end{bmatrix},$$

$$B_{21} = \begin{bmatrix} -1-i & -3i \\ 5 & 1+2i \end{bmatrix},\ C_{21} = \begin{bmatrix} -1-i & 2-i \\ -2+3i & 1+2i \end{bmatrix},\ B_{22} = \begin{bmatrix} 3+i & 2+3i \\ 3 & 1-7i \end{bmatrix},$$

$$A_{22} = \begin{bmatrix} 1-4i & 1+i \\ -1+3i & 2 \end{bmatrix},\ A_{21} = \begin{bmatrix} -1+0.5i & 0.5 \\ 1-2i & -2.5+1.5i \end{bmatrix},$$

$$F_1 = \begin{bmatrix} 74+52.5i & -124+38.5i \\ 23+44.5i & -134+83i \end{bmatrix},\ F_2 = \begin{bmatrix} 18-6i & -23-17i \\ -21.5-27.5i & 103+22.5i \end{bmatrix}.$$

These matrix equations have a unique solution

$$X_1 = \begin{bmatrix} 1+i & 2-3i \\ -1+2i & -2+3i \end{bmatrix},\ X_2 = \begin{bmatrix} 2+i & 3+i \\ 3+2i & 1+2i \end{bmatrix}.$$

Choose $X_1(0) = X_2(0) = X_1^i(0) = X_2^i(0) = 10^{-6} \times I_2$ for $i = 1, 2$ as the initial iterative matrices and define the relative iterative error is

$$f(k) = \sqrt{\frac{||X_1(k) - X_1||^2 + ||X_2(k) - X_2||^2}{||X_1||^2 + ||X_2||^2}}, \tag{6.2}$$

where $X_1(k)$ and $X_2(k)$ are the kth 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 convergence accuracy and efficiency by comparing these results.

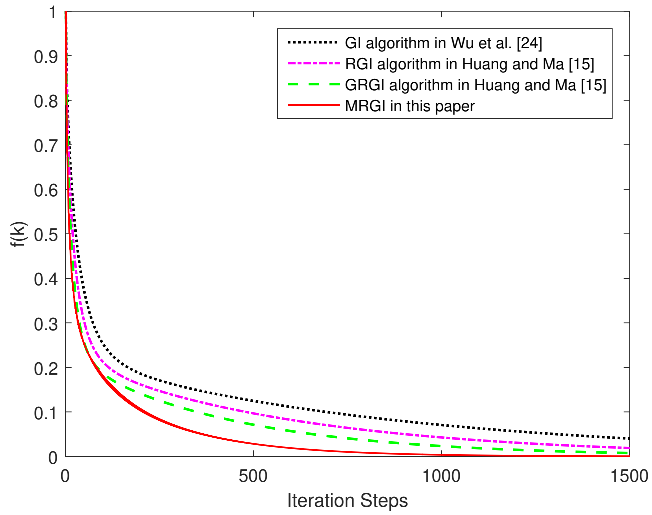

Figure 1. Comparison of convergence curve

Table 1. The iterative solution for the MRGI algorithm with $\mu1 = 2.2 \times 10^{-3}, \mu2 = 1.8 \times 10^{-3}$ and $\omega1 = 0.4, \omega2 = 0.6$

k	X1	X2
200	$\begin{bmatrix} 0.7453 + 1.0247i & 2.1434 - 2.9552i \\ -0.7792 + 1.6749i & -1.5885 + 3.2316i \end{bmatrix}$	$\begin{bmatrix} 2.0322 + 0.6542i & 2.9387 + 0.8960i \\ 3.2106 + 1.9661i & 0.9123 + 1.8146i \end{bmatrix}$
400	$\begin{bmatrix} 0.8835 + 1.0113i & 2.0696 - 2.9755i \\ -0.8953 + 1.8465i & -1.8154 + 3.1092i \end{bmatrix}$	$\begin{bmatrix} 2.0165 + 0.8460i & 2.9730 + 0.9507i \\ 3.0941 + 1.9933i & 0.9600 + 1.9137i \end{bmatrix}$
600	$\begin{bmatrix} 0.9467 + 1.0053i & 2.0325 - 2.9884i \\ -0.9513 + 1.9291i & -1.9157 + 3.0504i \end{bmatrix}$	$\begin{bmatrix} 2.0077 + 0.9298i & 2.9876 + 0.9773i \\ 3.0427 + 1.9978i & 0.9813 + 1.9602i \end{bmatrix}$
800	$\begin{bmatrix} 0.9755 + 1.0025i & 2.0150 - 2.9946i \\ -0.9776 + 1.9674i & -1.9613 + 3.0232i \end{bmatrix}$	$\begin{bmatrix} 2.0035 + 0.9678i & 2.9943 + 0.9895i \\ 3.0195 + 1.9991i & 0.9914 + 1.9817i \end{bmatrix}$
1000	$\begin{bmatrix} 0.9888 + 1.0011i & 2.0069 - 2.9975i \\ -0.9897 + 1.9850i & -1.9823 + 3.0106i \end{bmatrix}$	$\begin{bmatrix} 2.0016 + 0.9852i & 2.9974 + 0.9952i \\ 3.0090 + 1.9996i & 0.9960 + 1.9916i \end{bmatrix}$
Solution	$\begin{bmatrix} 1 + i & 2 - 3i \\ -1 + 2i & -2 + 3i \end{bmatrix}$	$\begin{bmatrix} 2 + i & 3 + i \\ 3 + 2i & 1 + 2i \end{bmatrix}$

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

Method	Steps	$f(k)$	Time (s)
GI algorithm in Wu et al. [24]	4819	9.5817×10^{-4}	2.5238
RGI algorithm in in Huang and Ma [15]	3332	9.5753×10^{-4}	1.7718
GRGI algorithm in in Huang and Ma [15]	2425	9.5582×10^{-4}	0.7904
MRGI algorithm in this paper	1327	9.5180×10^{-4}	0.7798

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

f(k)	k			
	GI algorithm	RGI algorithm	GRGI algorithm	MRGI algorithm
0.1	693	479	349	210
0.01	2731	1888	1374	753
0.001	4738	3276	2384	1304

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

$$\begin{cases} A_{11}X_1B_{11} + C_{11}\overline{X_1}D_{11} + A_{12}X_2B_{12} + C_{12}\overline{X_2}D_{12} = F_1, \\ A_{21}X_1B_{21} + C_{21}\overline{X_1}D_{21} + A_{22}X_2B_{22} + C_{22}\overline{X_2}D_{22} = F_2. \end{cases} \tag{6.3}$$

with the coefficient matrices:

$$\begin{aligned}
&A_{11} = diag(a + diag(rand(m))) - tril(rand(m), m)i, \quad B_{11} = eye(m) + eye(m)i, \\
&C_{11} = eye(m) - tril(rand(m), m)i, \quad D_{11} = rand(m) + triu(rand(m), m)i, \\
&A_{12} = diag(a + diag(rand(m))) - tril(rand(m), m)i, \\
&B_{12} = rand(m) \times a + triu(rand(m), m)i, \\
&C_{12} = eye(m) \times a + eye(m)i, \quad D_{12} = eye(m) \times a + tril(rand(m), m)i, \\
&A_{21} = diag(a + diag(rand(m))) \times a - triu(rand(m), m)i, \\
&B_{21} = diag(a + diag(rand(m))) + tril(rand(m), m)i, \\
&C_{21} = -eye(m) \times a - triu(rand(m), m)i, \quad D_{21} = eye(m) + eye(m)i, \\
&A_{22} = -rand(m) + eye(m)i, \quad B_{22} = eye(m) \times a + tril(rand(m), m)i, \\
&C_{22} = -eye(m) - +tril(rand(m), m)i, \quad D_{22} = rand(m) - triu(rand(m), m)i,
\end{aligned}$$

let

$$\begin{aligned}
X_1 =& diag(a + diag(rand(m))) + eye(m) \times a \\
&+ \Big(tril(rand(m), -1) \times a - (tril(rand(m), -1)^T)\Big)i, \\
X_2 =& \Big(tril(rand(m), -1) + triu(rand(m)^T, 0)\Big) \times a \\
&+ \Big(tril(rand(m), -1) - tril(rand(m), m)^T\Big)i,
\end{aligned}$$

then we have

$$\begin{aligned}
F_1 = A_{11}X_1B_{11} + C_{11}\overline{X_1}D_{11} + A_{12}X_2B_{12} + C_{12}\overline{X_2}D_{12}, \\
F_2 = A_{21}X_1B_{21} + C_{21}\overline{X_1}D_{21} + A_{22}X_2B_{22} + C_{22}\overline{X_2}D_{22}.
\end{aligned}$$

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 $X_1(0) = X_2(0) = X_1^i(0) = X_2^i(0) = 10^{-6} \times I_m$, $i = 1, 2$ 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.

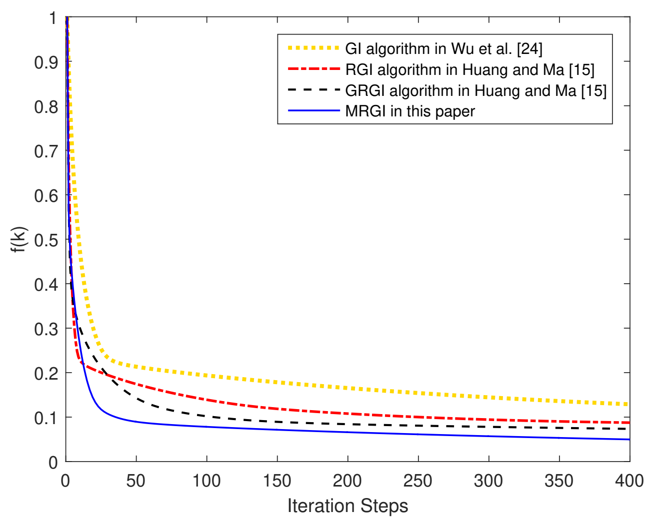

Figure 2. Comparison of convergence curves

Table 4. Iterative steps, relative residual and computational time results

Method	Steps	$f(k)$	Time (s)
GI algorithm in Wu et al. [24]	15822	3.6900×10^{-2}	13.1977
RGI algorithm in in Huang and Ma [15]	11948	1.2800×10^{-2}	9.9247
GRGI algorithm in in Huang and Ma [15]	7865	4.6000×10^{-3}	4.8349
MRGI algorithm in this paper	4207	9.4651×10^{-4}	3.6684

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 μ_1 and μ_2 in Figure 3. This situation illustrates that the convergence range of the convergence factors μ_1 and μ_2 calculated by Theorem 4.1 is a little conservative. How to reduce or remove this conservatism is our future work.

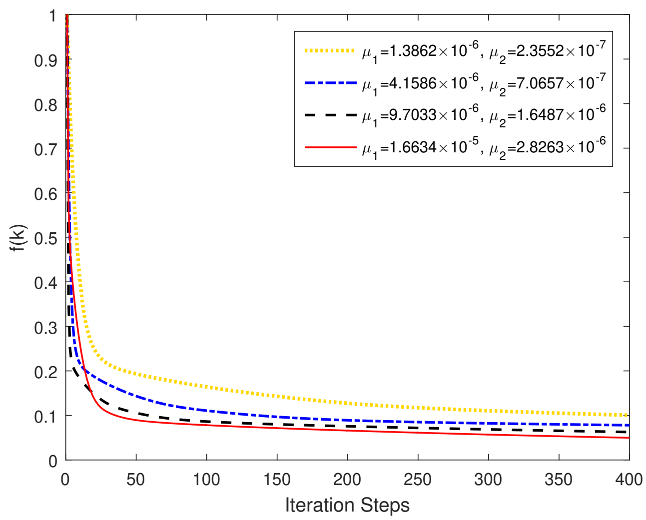

Figure 3. The convergence performance of MRGI for different convergence factors μ_1 and μ_2 ($\omega_1 = 0.4, \omega_2 = 0.6$)

7. Application in antilinear system

Consider the following discrete-time antilinear system

$$\begin{cases} \dot{x} = A\overline{x} + B\overline{u} + Pw \\ \dot{w} = Fw \\ e = Cx + Qw, \end{cases} \tag{7.1}$$

where $A \in \mathcal{C}^{n\times n}$, $B \in \mathcal{C}^{n\times r}$, $F \in \mathcal{C}^{p\times p}$, $P \in \mathcal{C}^{n\times p}$, $C \in \mathcal{C}^{m\times n}$ and $Q \in \mathcal{C}^{m\times p}$ are constant matrices, $x \in \mathcal{C}^n$, $u \in \mathcal{C}^r$ and $e \in \mathcal{C}^m$ are the state, the control input and the measurable error output, respectively. The symbol $w \in \mathcal{C}^p$ 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 $u = -Kx + \overline{L}\overline{w}$ is applied on the system, then the closed-loop system can result in

$$\begin{cases} \dot{x} = (A - B\overline{K})\overline{x} + (P + BL)w \\ \dot{w} = Fw \\ e = Cx + Qw. \end{cases} \tag{7.2}$$

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

$$\lim_{t\to\infty} e(t) = \lim_{t\to\infty} (Cx(t) + Qw(t)) = 0, \tag{7.3}$$

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

$$\begin{cases} A\overline{X} - XF = BY + R \\ CX + Q = 0. \end{cases} \tag{7.4}$$

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

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

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

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

[1] K. Adisorn and C. Pattrawut, *Approximated least-squares solutions of a generalized Sylvester-transpose matrix equation via gradient-descent iterative algorithm*, Advances in Difference Equations, 2021, 2021, 266.

[2] K. Adison, C. Pattrawut and L. Wicharn, *Convergence analysis of gradient-based iterative algorithms for a class of rectangular Sylvester matrix equations based on Banach contraction principle*, Advances in Difference Equations, 2021, 2021, 17.

[3] R. Agarwal, S. Hristova, D. O'Regan and K. Stefanova, *Iterative Algorithm for Solving Scalar Fractional Differential Equations with Riemann-Liouville Derivative and Supremum*, Algorithms, 2020, 13(8), 184.

[4] M. Dehghan and M. Hajarian, *The generalised Sylvester matrix equations over the generalised bisymmetric and skew-symmetric matrices*, International Journal of Systems Science, 2012, 43(8), 1580–1590.

[5] M. Dehghan and R. Mohammadi-Arani, *Generalized product-type methods based on bi-conjugate gradient (GPBiCG) for solving shifted linear systems*, Comput. Appl. Math., 2016, 36(4), 1–16.

[6] F. Ding, X. Liu and J. Ding, *Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle*, Appl. Math. Comput., 2008, 197(1), 41–50.

[7] F. Ding, F. Wang, L. Xu and M. Wu, *Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear ARMA systems using the data filtering*, J. Frankl. Inst., 2017, 354(3), 1321–1339.

[8] M. Hajarian, *Solving the general Sylvester discrete-time periodic matrix equations via the gradient based iterative method*, Appl. Math. Lett., 2016, 52, 87–95.

[9] M. Hajarian, *Gradient based iterative algorithm to solve general coupled discretetime periodic matrix equations over generalized reflexive matrices*, Math. Model. Anal., 2016, 21, 533–549.

[10] M. Hajarian, *New finite algorithm for solving the generalized nonhomogeneous Yakubovich-transpose matrix equation*, Asian J. Control, 2017, 19, 164–172.

[11] M. Hajarian, *Convergence of HS version of BCR algorithm to solve the generalized Sylvester matrix equation over generalized reflexive matrices*, J. Frankl. Inst., 2017, 354, 2340–2357.

[12] M. Hajarian, *Reflexive periodic solutions of general periodic matrix equations*, Mathematical Methods in the Applied Sciences, 2019, 42(10), 3527–3548.

[13] M. Hajarian, *Three types of biconjugate residual method for general periodic matrix equations over generalized bisymmetric periodic matrices*, Transactions of the Institute of Measurement and Control, 2019, 41(10), 2708–2725.

[14] B. Huang and C. Ma, *Gradient-based iterative algorithms for generalized coupled Sylvester-conjugate matrix equations*, Comput. Math. Appl., 2018, 75, 2295–2310.

[15] B. Huang and C. Ma, *The relaxed gradient-based iterative algorithms for a class of generalized coupled Sylvester-conjugate matrix equations*, J. Frankl. Inst., 2018, 355, 3168–3195.

[16] Y. Ji, H. J. Chizeck, X. Feng and K. A. Loparo, *Stability and control of discrete-time jump linear systems*, Control Theory Adv. Technol., 1991, 7(2), 247–270.

[17] T. Jiang and M. Wei, *On solutions of the matrix equations $X - AXB = C$ and $X - A\overline{X}B = C$*, Linear Algebra and its Applications, 2003, 367, 225–233.

[18] Q. Niu, X. Wang and L. Lu, *A relaxed gradient based algorithm for solving Sylvester equations*, Asian J. Control, 2011, 13, 461–464.

[19] X. Sheng, *A relaxed gradient based algorithm for solving generalized coupled Sylvester matrix equations*, J. Frankl. Inst., 2018, 355, 4282–4297.

[20] C. Song, G. Chen and L. Zhao, *Iterative solutions to coupled Sylvester-transpose matrix equations*, Appl. Math. Model., 2011, 35(10), 4675–4683.

[21] C. Vanhille, *A note on the convergence of the irrational Halley¡¯s iterative algorithm for solving nonlinear equations*, International Journal of Computer Mathematics, 2020, 97(9), 1840–1848.

[22] X. Wang, L. Dai and D. Liao, *A modified gradient based algorithm for solving Sylvester equations*, Appl. Math. Comput., 2012, 218, 5620–5628.

[23] X. Wang and D. Liao, *The optimal convergence factor of the gradient based iterative algorithm for linear matrix equations*, Filomat, 2012, 26, 607–613.

[24] A. Wu, G. Feng, G. Duan and W. Wu, *Iterative solutions to coupled Sylvester-conjugate matrix equations*, Comput. Math. Appl., 2010, 60(1), 54–66.

[25] A. Wu, L. Lv and G. Duan, *Iterative algorithms for solving a class of complex conjugate and transpose matrix equations*, Appl. Math. Comput., 2011, 217(21), 8343–8353.

[26] A. Wu, X. Zeng, G. Duan and W. Wu, *Iterative solutions to the extended Sylvester-conjugate matrix equatinos*, Appl. Math. Comput., 2010, 217(1), 130–142.

[27] Y. Xie and C. Ma, *The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation*, Appl. Math. Comput., 2016, 273, 1257–1269.

[28] T. Yan and C. Ma, *The BCR algorithms for solving the reflexive or anti-reflexive solutions of generalized coupled Sylvester matrix equations*, Journal of the Franklin Institute, 2020, 357, 12787–12807.

[29] L. Zhang, B. Huang and J. Lam, *H_∞ model reduction of Markovian jump linear systems*, Syst. Control Lett., 2003, 50, 103–118.

[30] H. Zhang, *Reduced-rank gradient-based algorithms for generalized coupled Sylvester matrix equations and its applications*, Comput. Math. Appl., 2015, 70, 2049–2062.

[31] H. Zhang and F. Ding, *A property of the eigenvalues of the symmetric positive definite matrix and the iterative algorithm for coupled Sylvester matrix equations*, J. Frankl. Inst., 2014, 351, 340–357.

[32] H. Zhang and H. Yin, *New proof of the gradient-based iterative algorithm for a complex conjugate and transpose matrix equation*, J. Frankl. Inst., 2017, 354, 7585–7603.

[33] B. Zhou and G. Duan, *Periodic Lyapunov equation based approaches to the stabilization of continuous-time periodic linear systems*, IEEE Trans. Autom. Control, 2012, 57, 2139–2146.

[34] B. Zhou, J. Lam and G. Duan, *Gradient-based maximal convergence rate iterative method for solving linear matrix equations*, International Journal of Computer Mathematics, 2010, 87(3), 515–527.