Iterative algorithms for the generalized discrete-time periodic Sylvester transpose matrix equations with application in the periodic state observer design of linear systems *

Rui Qi, Caiqin Song†

School of Mathematical Sciences, University of Jinan, Jinan 250022, PR China

Abstract

In this work, four iterative algorithms are provided for solving generalized discrete-time periodic Sylvester transpose matrix equations. Based on the Jacobi iterative algorithm and hierarchical identification principle, the present work provides the full-row rank Jacobi gradient iterative (RRJGI) algorithm, the full-row rank accelerated Jacobi gradient iterative (RRAJGI) algorithm, the full-column rank Jacobi gradient iterative (CRJGI) algorithm and the full-column rank accelerated Jacobi gradient iterative (CRAJGI) algorithm. The convergence of the algorithms are proved, and it is concluded that the proposed iterative methods are convergent under certain conditions for arbitrary initial matrices. Numerical results show the feasibility of the proposed algorithms and its superiority compared with other algorithms. Finally, an application example for the periodic state observer design of linear systems is given.

Keywords: The generalized discrete-time periodic Sylvester transpose matrix equations; iterative algorithm; iterative solutions; Convergence performance

1 Introduction

Periodic matrix equations are closely related to the analysis and synthesis of periodic control systems for various engineering and mechanical problems. The solutions of discrete-time period Sylvester matrix equations play an important role in engineering problems, such as modern control theory, prediction and potential applications in signal processing [1-6]. The reason is that the discrete-time periodic matrix equation is an important part of the analysis and design of linear

*This work is supported by Shandong Natural Science Foundation (No.ZR2020MA052, No.ZR2020MA055 and ZR2017BA025), and the National Natural Science Foundation of China (No.11501246 and No.11801216).

†Corresponding author. E-mail addresses: songcaiqin1983@163.com.(C.Song)

discrete periodic systems, and it has also received extensive attention [7-9]. For instance, the discrete-time period coupled Sylvester matrix equations

$$\begin{cases} A_{1,j}X_j + Y_jB_{1,j} = C_{1,j}, \\ A_{2,j}X_{j+1} + Y_jB_{2,j} = C_{2,j}, \end{cases} \tag{1.1}$$

is encountered in the periodic discrete-time description subsystem [10, 11].

In recent years, many scholars have proposed effective methods for solving periodic matrix equations. For example, in [12], Hajarian based on the bi-conjugate residual algorithm (BCR) proposed a new numerical method for solving discrete-time periodic Sylveste matrix equations

$$A_iX_iB_i + C_iX_{i+1}D_i = E_i, i \in \overline{1,\gamma}. \tag{1.2}$$

In addition, in [13] he also proposed two iterative algorithms based on the LSQR method for solving Eq. (1.2) and Ma et al. in [33] generalized the factor gradient iterative method (FGI) for solving Eq. (1.2). Wang and Song in [14] proposed the Jacobi gradient iterative algorithm (JGI) and the accelerated JGI algorithm for solving the following periodic matrix equations

$$\sum_{s=1}^{p} A_{i,s}X_iB_{i,s} + \sum_{t=1}^{q} E_{i,t}X_{i+1}F_{i,t} = C_i, i \in \overline{1,\gamma}. \tag{1.3}$$

In [31], Hajarian provided four new iterative methods to find the reflexive periodic solutions of the general periodic matrix equations

$$\sum_{s=0}^{\sigma-1} (A_{i,s}X_{i+s}B_{i,s}) + \sum_{t=0}^{\sigma-1} (C_{i,t}Y_{i+t}D_{i,t}) = N_i, i = 1, 2, \cdots, \sigma. \tag{1.4}$$

Ma et al. in [34] proposed a finite iterative algorithm to find the least squares solutions of periodic matrix equations (1.4). In [15], Huang and Ma constructed a finite iterative algorithm to solve the least square solution of the periodic matrix equations. Lv et al. in [16] developed the least square method to give an iterative algorithm for solving the generalized periodic discrete-time coupled Sylvester matrix equations

$$\begin{cases} A_{1,j}X_jB_{1,j} + C_{1,j}Y_jD_{1,j} = E_{1,j}, \\ A_{2,j}X_{j+1}B_{2,j} + C_{2,j}Y_jD_{2,j} = E_{2,j}. \end{cases} \tag{1.5}$$

Chen et al. in [30] constructed a conjugate gradient-based (CGB) method for solving Eq. (1.5). In [17], Ma and Yan established an improved conjugate gradient algorithm to solve the generalized discrete-time period Sylveste matrix equations

$$\sum_{j=1}^{h} (A_{ij}X_iB_{ij} + C_{ij}X_{i+1}D_{ij} + E_{ij}Y_iF_{ij} + G_{ij}Y_{i+1}H_{ij}) = M_i, i = 1, 2, \cdots, T. \tag{1.6}$$

Moreover, in [18], Hajarian proposed the gradient based iterative (GI) algorithm to solve Eq. (1.6). In [19], he also derived the matrix form of the conjugate gradient normal equations residual minimizing (MCGNR) algorithm to find the least squares solution group of discrete-time periodic coupled matrix equations

$$\begin{cases} A_{1,t}X_tB_{1,t} + C_{1,t}X_{t+1}D_{1,t} + E_{1,t}Y_tF_{1,t} = G_{1,t}, \\ A_{2,t}X_tB_{2,t} + C_{2,t}X_{t+1}D_{2,t} + E_{2,t}Y_tF_{2,t} = G_{2,t}. \end{cases} \tag{1.7}$$

There are also many studies on the coupled Sylvestre transpose matrix equation in recent years. For example, in [35] Boonruangkan et al. based on gradients and hierarchical identification principle, built an iterative algorithm for solving the generalized Sylvester-transpose matrix equation

$$\sum_{i=1}^{p} A_i X B_i + \sum_{j=1}^{q} C_j X^T D_j = F, \tag{1.8}$$

Tansri et al. in [32] developed a conjugate-gradient type algorithm to produce approximate least-squares (LS) solution for an inconsistent generalized Sylvester-transpose matrix equation (1.8). Kittisopaporn et al. in [29] established an effective gradient-descent iterative algorithm for solving Eq. (1.8). At present, there have been a lot of research results on iterative algorithms for solving various matrix equations. Now we don't state in detail. Please refer to references [20-27].

This paper focus on the following generalized discrete-time periodic Sylvester transpose matrix equations

$$\sum_{j=1}^{m} (E_{i,j} Y_i F_{i,j} + G_{i,j} Y_{i+1}^T H_{i,j}) = M_i, i \in \overline{1,\xi}, \tag{1.9}$$

where the coefficient matrices $E_{i,j}, G_{i,j} \in R^{m\times m}$, $F_{i,j}, H_{i,j} \in R^{n\times n}$, $M_i \in R^{m\times n}$, and unknown matrices $Y_i \in R^{m\times n}$ are periodic with period ξ, i.e. $E_{i+\xi,j} = E_{i,j}$, $F_{i+\xi,j} = F_{i,j}$, $G_{i+\xi,j} = G_{i,j}$, $H_{i+\xi,j} = H_{i,j}$, $M_{i+\xi} = M_i$, $Y_{i+\xi} = Y_i$, for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$. The problem for solving periodic matrix equations appears in various application fields, but there are few researches on the iterative solutions of periodic matrix equations. In this paper, based on Jacobi iterative algorithm, four iterative algorithms are proposed to solve the generalized discrete-time periodic Sylvester transpose matrix equations (1.9). The main contributions of this paper are as follows.

- In this present work, four iterative algorithms are presented for Eq. (1.9), which are RRJGI algorithm, CRJGI algorithm, RRAJGI algorithm and CRAJGI algorithm. Moreover, four algorithms presented in this paper are not only suitable for solving the above generalized discrete-time periodic Sylvester transpose matrix equations, but also can solve the numerical solutions of the coupled discrete-time periodic matrix equations if we give some small changes.
- Numerical examples show that the proposed algorithms have higher convergence efficiency compared with the GI algorithm [18], the relaxed gradient based iterative (RGI) algorithm [23] and the accelerated gradient based iterative (AGI) algorithm [26] because less cost is used in each iterative step and the data is sufficient to complete an update. And each update uses less data, which can greatly save memory space and improve operation efficiency.
- By applying Algorithm 2 to the linear systems, we obtain Algorithm 5 for solving robust and minimum norm observer design of linear systems and a group of data is given to deduce the gain of the state observer, which shows that the proposed algorithm provides a choice for solving linear systems.

The rest of this article is arranged as follows. In Section 2, we provide several basic notations and related theories. In Section 3, we present RRJGI algorithm, RRAJGI algorithm, CRJGI algorithm and CRAJGI algorithm for solving Eq. (1.9), and also analyze the convergence of four algorithms. In addition, we prove that for any given initial matrices, the iterative solutions obtained by the proposed algorithms will converge to the exact solutions. In Section 4, we give two examples to demonstrate the superiority of the proposed algorithms. The numerical results indicate that these four algorithms are ascendant to the GI algorithm [18], the RGI algorithm [23] and the AGI algorithm [26]. In Section 5, an application example for solving the periodic state observer design of linear discrete system is given. Finally, in Section 6, we give a short conclusion.

2 Preliminaries

Throughout this paper, we use the following notations. Let $R^{s\times t}$ be the set of all matrices of size $s\times t$ over the real number field R. For $A\in R^{s\times t}$, A^T, $rank(A)$, $\rho(A)$, $\lambda(A)$ and $tr\,(A)$ represent the transpose, the rank, the spectral radius, the eigenvalues and the trace of A, respectively. For arbitrary integers p and q with $p\le q$, we denote $\overline{p,q}=\{p,p+1,\cdots,q\}$. For any matrices $A,B\in R^{s\times t}$, $A\otimes B$ represented the Kronecker product of A and B. For $X=(x_1,x_2,\cdots,x_n)\in R^{m\times n}$, $vec\,(X)=\left(x_1^T,x_2^T,\cdots,x_n^T\right)^T$ is represented as the streching operator of X. By combining vector operator with Kronecker product, we get $vec\,(AXB)=\left(B^T\otimes A\right)vec\,(X)$. The real inner product of two matrices $A,B\in R^{s\times t}$ is given by $\langle A,B\rangle=tr\left(A^TB\right)$. $\|A\|$ stands for the Frobenious norm of the matrix A and $||A||_2=\sqrt{\lambda_{max}(A^TA)}$ represents the 2-norm of the matrix A. I stands for the identity matrix of the appropriate dimension.

Lemma 2.1. [27] *Consider the matrix equation*

$$A_1XB_1=C, \tag{2.1}$$

where $A_1\in R^{m\times r}, B_1\in R^{s\times n}$ and $C\in R^{m\times n}$ are known matrices, and $X\in R^{r\times s}$ is unknown matrix. Then, the solution of Eq. (2.1) can be obtained by the following algorithm

$$X\,(k+1)=X\,(k)+\mu A_1^T\left(C-A_1X\,(k)\,B_1\right)B_1^T, \tag{2.2}$$

with

$$0<\mu<\frac{2}{\|A_1\|_2^2\,\|B_1\|_2^2}. \tag{2.3}$$

Lemma 2.2. [28] *The unique solution of $Sx=b$ can be given by $x=S^\dagger b$, where $S^\dagger$ is the unique Moore-Penrose inverse of S. Especially, if S is a full-column rank matrix, then the unique solution is given by*

$$x=(S^TS)^{-1}S^Tb, \tag{2.4}$$

if S is a full-row rank matrix, then the unique minimum norm solution is

$$x=S^T(SS^T)^{-1}b. \tag{2.5}$$

Next, we give a lemma by using of Lemma 2.2. In order to convenient expression, the following symbols are defined.

$$\mathcal{E}_j=diag(E_{1,j},E_{2,j},\cdots,E_{\xi,j}), \tag{2.6}$$
$$\mathcal{F}_j=diag(F_{1,j},F_{2,j},\cdots,F_{\xi,j}), \tag{2.7}$$
$$\mathcal{Y}=diag(Y_1,Y_2,\cdots,Y_\xi), \tag{2.8}$$
$$\mathcal{M}=diag(M_1,M_2,\cdots,M_\xi), \tag{2.9}$$

$$\mathcal{G}_j = \begin{pmatrix} 0 & G_{1,j} & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & G_{\xi-1,j} \\ G_{\xi,j} & 0 & \cdots & 0 \end{pmatrix}, \qquad \mathcal{H}_j = \begin{pmatrix} 0 & \cdots & 0 & H_{\xi,j} \\ H_{1,j} & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & & H_{\xi-1,j} & 0 \end{pmatrix}. \tag{2.10}$$

Lemma 2.3. *Let* $A = \sum_{j=1}^{m} [\mathcal{F}_j^T \otimes \mathcal{E}_j + (\mathcal{H}_j^T \otimes \mathcal{G}_j)P]$, *where* P *is the permutation matrix that satisfies* $vec(\mathcal{Y}^T) = Pvec(\mathcal{Y})$, *if* A *is a full-column rank matrix, then the unique solution of Eq. (1.9) is given by*

$$vec(\mathcal{Y}) = (A^T A)^{-1} A^T vec(\mathcal{M}), \tag{2.11}$$

if A *is a full-row rank matrix, then the unique minimum norm solution of Eq. (1.9) is*

$$vec(\mathcal{Y}) = A^T (AA^T)^{-1} vec(\mathcal{M}). \tag{2.12}$$

Proof. Eq. (1.9) can be equivalent to

$$\sum_{j=1}^{m} (\mathcal{E}_j \mathcal{Y} \mathcal{F}_j + \mathcal{G}_j \mathcal{Y}^T \mathcal{H}_j) = \mathcal{M}, \tag{2.13}$$

where $\mathcal{E}_j, \mathcal{F}_j, \mathcal{G}_j, \mathcal{H}_j$ and $\mathcal{Y}, \mathcal{M}$ are defined as (2.6)-(2.10). By the properties of the Kronecker product and the vector function, Eq. (1.9) can be converted to the following form

$$\{\sum_{j=1}^{m} [\mathcal{F}_j^T \otimes \mathcal{E}_j + (\mathcal{H}_j^T \otimes \mathcal{G}_j)P]\} vec(\mathcal{Y}) = vec(\mathcal{M}), \tag{2.14}$$

i.e.

$$Avec(\mathcal{Y}) = vec(\mathcal{M}). \tag{2.15}$$

Therefore, according to Lemma 2.2 if A is a full-column rank matrix, then Eq. (1.9) has a unique solution, and if A is a full-row rank matrix, then Eq. (1.9) has unique minimum norm solutions and completed the proof. □

3 Iterative algorithms and convergence analysis

In this section, by using the Jacobi iterative algorithm we construct four iterative algorithms for solving Eq.(1.9). In Lemma 2.1, when the size of the coefficient matrix is too large, it will require longer running time and more storage space for solving equation (2.1) by (2.2). Therefore, the big coefficient matrix is divided into the corresponding diagonal matrix.

According to the above analysis, based on Jacobi iterative algorithm and hierarchical identification principle, we propose the RRJGI algorithm, RRAJGI algorithm, CRJGI algorithm and

CRAJGI algorithm for computing the numerical solutions of Eq. (1.9). First, the coefficient matrices $E_{i,j}, F_{i,j}, G_{i,j}, H_{i,j}$ are decomposed into the following form:

$$E_{i,j} = D_{i,j}^{(1)} + R_{i,j}^{(1)}, \tag{3.1}$$

$$F_{i,j} = D_{i,j}^{(2)} + R_{i,j}^{(2)}, \tag{3.2}$$

$$G_{i,j} = D_{i,j}^{(3)} + R_{i,j}^{(3)}, \tag{3.3}$$

$$H_{i,j} = D_{i,j}^{(4)} + R_{i,j}^{(4)}, \tag{3.4}$$

where $D_{i,j}^{(1)}, D_{i,j}^{(2)}, D_{i,j}^{(3)}, D_{i,j}^{(4)}$ are the diagonal part of $E_{i,j}, F_{i,j}, G_{i,j}, H_{i,j}$, $i \in \overline{1,\xi}$, $j \in \overline{1,m}$, respectively. Thus, $D_{i,j}^{(1)}, D_{i,j}^{(2)}, D_{i,j}^{(3)}$ and $D_{i,j}^{(4)}$ satisfy the following relations

$$(D_{i,j}^{(1)})^T = D_{i,j}^{(1)}, \qquad (D_{i,j}^{(2)})^T = D_{i,j}^{(2)}, \tag{3.5}$$

$$(D_{i,j}^{(3)})^T = D_{i,j}^{(3)}, \qquad (D_{i,j}^{(4)})^T = D_{i,j}^{(4)}. \tag{3.6}$$

Next, we present two intermediary matrices $b_i^{(1)}, b_i^{(2)}$ as follows:

$$b_i^{(1)} = M_i - \sum_{j=1}^{m} G_{i,j} Y_{i+1}^T H_{i,j}, \tag{3.7}$$

$$b_i^{(2)} = M_i - \sum_{j=1}^{m} E_{i,j} Y_i F_{i,j}. \tag{3.8}$$

Therefore, Eq. (1.9) can be simply written as

$$\sum_{j=1}^{m} E_{i,j} Y_i F_{i,j} = b_i^{(1)}, \tag{3.9}$$

$$\sum_{j=1}^{m} G_{i,j} Y_{i+1}^T H_{i,j} = b_i^{(2)}. \tag{3.10}$$

Substituting (3.1)-(3.4) into (3.9) and (3.10), respectively, we have

$$\sum_{j=1}^{m} (D_{i,j}^{(1)} + R_{i,j}^{(1)}) Y_i (D_{i,j}^{(2)} + R_{i,j}^{(2)}) = b_i^{(1)}, \tag{3.11}$$

$$\sum_{j=1}^{m} (D_{i,j}^{(3)} + R_{i,j}^{(3)}) Y_{i+1}^T (D_{i,j}^{(4)} + R_{i,j}^{(4)}) = b_i^{(2)}, \tag{3.12}$$

that is,

$$\sum_{j=1}^{m} D_{i,j}^{(1)} Y_i D_{i,j}^{(2)} = b_i^{(1)} - \sum_{i=1}^{m} (D_{i,j}^{(1)} Y_i R_{i,j}^{(2)} + R_{i,j}^{(1)} Y_i D_{i,j}^{(2)} + R_{i,j}^{(1)} Y_i R_{i,j}^{(2)}), \tag{3.13}$$

$$\sum_{j=1}^{m} D_{i,j}^{(3)} Y_{i+1}^T D_{i,j}^{(4)} = b_i^{(2)} - \sum_{i=1}^{m} (D_{i,j}^{(3)} Y_{i+1}^T R_{i,j}^{(4)} + R_{i,j}^{(3)} Y_{i+1}^T D_{i,j}^{(4)} + R_{i,j}^{(3)} Y_{i+1}^T R_{i,j}^{(4)}). \tag{3.14}$$

By Lemma 2.1, we can derive the iterative algorithms for solving (3.13) and (3.14)

$$Y_i^{(1)}(k+1) = Y_i^{(1)}(k) + \mu \sum_{j=1}^{m} D_{i,j}^{(1)}[b_i^{(1)} - \sum_{j=1}^{m}(D_{i,j}^{(1)} Y_i R_{i,j}^{(2)} + R_{i,j}^{(1)} Y_i D_{i,j}^{(2)}$$

$$+ R_{i,j}^{(1)} Y_i R_{i,j}^{(2)}) - \sum_{j=1}^{m} D_{i,j}^{(1)} Y_i(k) D_{i,j}^{(2)}] D_{i,j}^{(2)}, \tag{3.15}$$

$$Y_i^{(2)T}(k+1) = Y_i^{(2)T}(k) + \mu \sum_{j=1}^{m} D_{i-1,j}^{(3)}[b_{i-1}^{(2)} - \sum_{j=1}^{m}(D_{i-1,j}^{(3)} Y_i^T R_{i-1,j}^{(4)} + R_{i-1,j}^{(3)} Y_i^T D_{i-1,j}^{(4)}$$

$$+ R_{i-1,j}^{(3)} Y_i^T R_{i-1,j}^{(4)}) - \sum_{j=1}^{m} D_{i-1,j}^{(3)} Y_i^T(k) D_{i-1,j}^{(4)}] D_{i-1,j}^{(4)}. \tag{3.16}$$

Now, we take the transpose of both sides of algorithm (3.16), and we can get

$$Y_i^{(2)}(k+1) = Y_i^{(2)}(k) + \mu \sum_{j=1}^{m} D_{i-1,j}^{(4)}[b_{i-1}^{(2)} - \sum_{j=1}^{m}(D_{i-1,j}^{(3)} Y_i^T R_{i-1,j}^{(4)} + R_{i-1,j}^{(3)} Y_i^T D_{i-1,j}^{(4)}$$

$$+ R_{i-1,j}^{(3)} Y_i^T R_{i-1,j}^{(4)}) - \sum_{j=1}^{m} D_{i-1,j}^{(3)} Y_i^T(k) D_{i-1,j}^{(4)}]^T D_{i-1,j}^{(3)}. \tag{3.17}$$

Substituting (3.7) and (3.8) into (3.15) and (3.17), we can obtain

$$Y_i^{(1)}(k+1) = Y_i^{(1)}(k) + \mu \sum_{i=1}^{m} D_{i,j}^{(1)}[M_i - \sum_{q=1}^{n} G_{i,j} Y_{i+1}^T G_{i,j} - \sum_{j=1}^{m}(D_{i,j}^{(1)} Y_i R_{i,j}^{(2)}$$

$$+ R_{i,j}^{(1)} Y_i D_{i,j}^{(2)} + R_{i,j}^{(1)} Y_i R_{i,j}^{(2)}) - \sum_{j=1}^{m} D_{i,j}^{(1)} Y_i(k) D_{i,j}^{(2)}] D_{i,j}^{(2)}, \tag{3.18}$$

$$Y_i^{(2)}(k+1) = Y_i^{(2)}(k) + \mu \sum_{j=1}^{m} D_{i-1,j}^{(4)}[M_{i-1} - \sum_{j=1}^{m} E_{i-1,j} Y_{i-1} F_{i-1,j} - \sum_{j=1}^{m}(D_{i-1,j}^{(3)} Y_i^T R_{i-1,j}^{(4)}$$

$$+ R_{i-1,j}^{(3)} Y_i^T D_{i-1,j}^{(4)} + R_{i-1,j}^{(3)} Y_i^T R_{i-1,j}^{(4)}) - \sum_{j=1}^{m} D_{i-1,j}^{(3)} Y_i(k)^T D_{i-1,j}^{(4)}]^T D_{i-1,j}^{(3)}. \tag{3.19}$$

To make the iterative procedure operate correctly, we can substitute the unknown matrix Y_i with the iterative solution $Y_i(k)$ acquired at the kth moment, so we can obtain the following algorithms

$$Y_i^{(1)}(k+1) = Y_i(k) + \mu \sum_{j=1}^{m} D_{i,j}^{(1)}[M_i - \sum_{j=1}^{m}(E_{i,j} Y_i(k) F_{i,j} + G_{i,j} Y_{i+1}^T(k) H_{i,j})] D_{i,j}^{(2)}, \tag{3.20}$$

$$Y_i^{(2)}(k+1) = Y_i(k) + \mu \sum_{j=1}^{m} D_{i-1,j}^{(4)}[M_{i-1} - \sum_{j=1}^{m}(E_{i-1,j} Y_{i-1}(k) F_{i-1,j}$$

$$+ G_{i-1,j} Y_i^T(k) H_{i-1,j})]^T D_{i-1,j}^{(3)}. \tag{3.21}$$

3.1 The RRJGI algorithm, the RRAJGI algorithm and convergence analysis

Now, we introduce the full-row rank Jacobi gradient based iterative (RRJGI) algorithm for solving matrix equations (1.9).

Algorithm 1 (The RRJGI algorithm)

Step 1. Given the coefficient matrices $E_{i,j}, G_{i,j} \in R^{m\times m}, F_{i,j}, H_{i,j} \in R^{n\times n}$ and $M_i \in R^{m\times n}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$, choose an appropriate convergence number μ and the initial matrices $K_i(0) \in R^{m\times n}, Y_i(0) = \sum\limits_{j=1}^{m} (D_{i,j}^{(1)} K_i(0) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(0) D_{i,j}^{(3)})$.

Step 2. Set $K_{i+\xi}(0) = K_i(0), Y_{i+\xi}(0) = Y_i(0), E_{i+\xi,j} = E_{i,j}, F_{i+\xi,j} = F_{i,j}, G_{i+\xi,j} = G_{i,j}, H_{i+\xi,j} = H_{i,j}, M_{i+\xi} = M_i, D_{i+\xi,j}^{(1)} = D_{i,j}^{(1)}, D_{i+\xi,j}^{(2)} = D_{i,j}^{(2)}, D_{i+\xi,j}^{(3)} = D_{i,j}^{(3)}$, and $D_{i+\xi,j}^{(4)} = D_{i,j}^{(4)}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$. Let $k := 0$.

Step 3. If $\delta(k) = \frac{\sum\limits_{i=1}^{\xi} ||M_i - \sum_{j=1}^{m}(E_{i,j} Y_i(k) F_{i,j} + G_{i,j} Y_{i+1}^T(k) H_{i,j})||^2}{\sum\limits_{i=1}^{\xi} ||M_i||^2} < \varepsilon$, stop; otherwise, go to Step 4.

Step 4. Compute the following sequences

$$
\begin{aligned}
K_{1,i}(k+1) &= K_i(k) + \mu\{M_i/2 - \sum_{j=1}^{m} E_{i,j}[\sum_{j=1}^{m} (D_{i,j}^{(1)} K_i(k) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(k) D_{i,j}^{(3)})] F_{i,j}\},\\
K_{2,i}(k+1) &= K_i(k) + \mu\{M_{i-1}/2 - \sum_{j=1}^{m} G_{i-1,j}[\sum_{j=1}^{m} (D_{i-1,j}^{(1)} K_{i-1}(k) D_{i-1,j}^{(2)} \\
&\quad + D_{i-1,j}^{(4)} K_i^T(k) D_{i-1,j}^{(3)})]^T H_{i-1,j}\},\\
K_i(k+1) &= \frac{K_{1,i}(k+1) + K_{2,i}(k+1)}{2},\\
K_{i+\xi}(k+1) &= K_i(k+1),\\
Y_i(k+1) &= \sum_{j=1}^{m} (D_{i,j}^{(1)} K_i(k+1) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(k+1) D_{i,j}^{(3)}),\\
Y_{i+\xi}(k+1) &= Y_i(k+1).
\end{aligned}
$$

Step 5. Let $k := k+1$, go to Step 3.

Remark 1. The construction of Algorithm 1 is based on the splitting of matrix equations and by the introduction of matrices $b_i^{(1)}$ and $b_i^{(2)}$, we divide Eq. (1.9) into Eq. (3.9) and Eq. (3.10), where $b_i^{(1)} + b_i^{(2)} = M_i$. It should be noted that the decomposition form of Eq. (1.9) is arbitrary, and the values of $b_i^{(1)}$ and $b_i^{(2)}$ do not affect the progress of the algorithm. For convenience, we take $b_i^{(1)} = b_i^{(2)} = M_i/2$ in Algorithm 1. Then, based on the hierarchical identification principle and Jacobi iterative method, the iterative algorithms $K_{1,i}(k+1)$ and $K_{2,i}(k+1)$ are constructed to solve Eq. (3.9) and Eq. (3.10), respectively. Next combine $K_{1,i}(k+1)$ and $K_{2,i}(k+1)$ to get the iterative value $K_i(k+1)$, and finally the iterative solution $Y_i(k+1)$ of Eq. (1.9) is obtained.

In order to improve the convergence speed and save time, we introduce an appropriate factor

ω_2, $0 < \omega_2 < 1$ on the basis of RRJGI algorithm and propose an full-row rank accelerated Jacobi gradient based iterative (RRAJGI) algorithm for solving Eq. (1.9).

Algorithm 2 (The RRAJGI algorithm)

Step 1. Given the coefficient matrices $E_{i,j}, G_{i,j} \in R^{m\times m}, F_{i,j}, H_{i,j} \in R^{n\times n}$ and $M_i \in R^{m\times n}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$, choose the initial matrices $K_i(0), K_{2,i}(0) \in R^{m\times n}, Y_i(0) = \sum_{j=1}^{m} (D_{i,j}^{(1)} K_i(0) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(0) D_{i,j}^{(3)})$.

Step 2. Set $K_{i+\xi}(0) = K_i(0), Y_{i+\xi}(0) = Y_i(0), E_{i+\xi,j} = E_{i,j}, F_{i+\xi,j} = F_{i,j}, G_{i+\xi,j} = G_{i,j}, H_{i+\xi,j} = H_{i,j}, M_{i+\xi} = M_i, D_{i+\xi,j}^{(1)} = D_{i,j}^{(1)}, D_{i+\xi,j}^{(2)} = D_{i,j}^{(2)}, D_{i+\xi,j}^{(3)} = D_{i,j}^{(3)}$ and $D_{i+\xi,j}^{(4)} = D_{i,j}^{(4)}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$. Let $k := 0$.

Step 3. If $\delta(k) = \frac{\sum_{i=1}^{\xi} ||M_i - \sum_{j=1}^{m} (E_{i,j} Y_i(k) F_{i,j} + G_{i,j} Y_{i+1}^T(k) H_{i,j})||^2}{\sum_{i=1}^{\xi} ||M_i||^2} < \varepsilon$, stop; otherwise, go to Step 4.

Step 4. Compute the following sequences

$$
\begin{aligned}
&K_{1,i}(k+1) = K_i(k) + \mu\omega_2\{M_i/2 - \sum_{j=1}^{m} E_{i,j}[\sum_{j=1}^{m} (D_{i,j}^{(1)} K_i(k) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(k) D_{i,j}^{(3)})]F_{i,j}\},\\
&\widehat{K}_i(k) = (1-\omega_2)K_{1,i}(k+1) + \omega_2 K_{2,i}(k),\\
&\widehat{K}_{i+\xi}(k) = \widehat{K}_i(k),\\
&K_{2,i}(k+1) = \widehat{K}_i(k) + \mu(1-\omega_2)\{M_{i-1}/2 - \sum_{j=1}^{m} G_{i-1,j}[\sum_{j=1}^{m} (D_{i-1,j}^{(1)} \widehat{K}_{i-1}(k) D_{i-1,j}^{(2)}\\
&\qquad + D_{i-1,j}^{(4)} \widehat{K}_i^T(k) D_{i-1,j}^{(3)})]^T H_{i-1,j}\},\\
&K_i(k+1) = (1-\omega_2)K_{1,i}(k+1) + \omega_2 K_{2,i}(k+1),\\
&K_{i+\xi}(k+1) = K_i(k+1),\\
&Y_i(k+1) = \sum_{j=1}^{m} (D_{i,j}^{(1)} K_i(k+1) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(k+1) D_{i,j}^{(3)}),\\
&Y_{i+\xi}(k+1) = Y_i(k+1).
\end{aligned}
$$

Step 5. Let $k := k+1$, go to Step 3.

Theorem 3.1. *Let A be a full-row rank matrix, the iterative solution $K(k) = (K_1(k), K_2(k), \cdots, K_\xi(k))$ given by Algorithm 1 (RRJGI) converges to the unique solutions $K^*(k) = (K_1^*(k), K_2^*(k), \cdots, K_\xi^*(k))$ for arbitrary initial matrices $K(0) = (K_1(0), K_2(0), \cdots, K_\xi(0))$, if μ satisfies*

$$0 < \mu < \frac{2}{\sum_{i=1}^{\xi}\sum_{j=1}^{m}\left(\left\|D_{i,j}^{(1)}\right\|^2\left\|D_{i,j}^{(2)}\right\|^2 + \left\|D_{i,j}^{(3)}\right\|^2\left\|D_{i,j}^{(4)}\right\|^2\right)}. \tag{3.22}$$

Proof. The error matrices are defined as follows

$$\widetilde{K}_i(k) = K_i(k) - K_i^*, \qquad \widetilde{K}_{1,i}(k) = K_{1,i}(k) - K_i^*, \qquad \widetilde{K}_{2,i}(k) = K_{2,i}(k) - K_i^*, \tag{3.23}$$

and

$$\widetilde{\varphi}_i(k) = \sum_{j=1}^{m} (D_{i,j}^{(1)} \widetilde{K}_i(k) D_{i,j}^{(2)} + D_{i,j}^{(4)} \widetilde{K}_{i+1}^T(k) D_{i,j}^{(3)}). \tag{3.24}$$

From (3.23)-(3.26), Algorithm 1 and Remark 1, it is obvious that

$$\begin{aligned}
&\widetilde{K}_{1,i}(k+1)\\
&= K_{1,i}(k+1) - K_i^*\\
&= K_i(k) - K_i^* + \mu\{M_i/2 - \sum_{j=1}^{m} E_{i,j}[\sum_{j=1}^{m}(D_{i,j}^{(1)} K_i(k) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(k) D_{i,j}^{(3)})] F_{i,j}\}\\
&= \widetilde{K}_i(k) + \mu\{\sum_{j=1}^{m} E_{i,j}[\sum_{j=1}^{m}(D_{i,j}^{(1)} K_i^* D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^{*T} D_{i,j}^{(3)})] F_{i,j}\}\\
&\quad - \mu\{\sum_{j=1}^{m} E_{i,j}[\sum_{j=1}^{m}(D_{i,j}^{(1)} K_i(k) D_{i,j}^{(2)} + D_{i,j}^{(4)} K_{i+1}^T(k) D_{i,j}^{(3)})] F_{i,j}\}\\
&= \widetilde{K}_i(k) + \mu\{\sum_{j=1}^{m} E_{i,j}[\sum_{j=1}^{m}(D_{i,j}^{(1)}(K_i^* - K_i(k)) D_{i,j}^{(2)} + D_{i,j}^{(4)}(K_{i+1}^{*T} - K_{i+1}^T(k)) D_{i,j}^{(3)})] F_{i,j}\}\\
&= \widetilde{K}_i(k) - \mu\{\sum_{j=1}^{m} E_{i,j}[\sum_{j=1}^{m}(D_{i,j}^{(1)} \widetilde{K}_i(k) D_{i,j}^{(2)} + D_{i,j}^{(4)} \widetilde{K}_{i+1}^T(k) D_{i,j}^{(3)})] F_{i,j}\}\\
&= \widetilde{K}_i(k) - \mu \sum_{j=1}^{m} E_{i,j} \widetilde{\varphi}_i(k) F_{i,j},
\end{aligned} \tag{3.25}$$

and

$$\begin{aligned}
&\widetilde{K}_{2,i}(k+1)\\
&= K_{2,i}(k) - K_i^*\\
&= K_i(k) - K_i^* + \mu\{M_{i-1}/2 - \sum_{j=1}^{m} G_{i-1,j}[\sum_{j=1}^{m}(D_{i-1,j}^{(1)} K_{i-1}(k) D_{i-1,j}^{(2)}\\
&\quad + D_{i-1,j}^{(4)} K_i^T(k) D_{i-1,j}^{(3)})]^T H_{i-1,j}\}\\
&= \widetilde{K}_i(k) + \mu\{\sum_{j=1}^{m} G_{i-1,j}[\sum_{j=1}^{m}(D_{i-1,j}^{(1)}(K_i^* - K_{i-1}(k)) D_{i-1,j}^{(2)}\\
&\quad + D_{i-1,j}^{(4)}(K_i^{*T} - K_i^T(k)) D_{i-1,j}^{(3)})]^T H_{i-1,j}\}\\
&= \widetilde{K}_i(k) - \mu\{\sum_{j=1}^{m} G_{i-1,j}[\sum_{j=1}^{m}(D_{i-1,j}^{(1)} \widetilde{K}_{i-1}(k) D_{i-1,j}^{(2)} + D_{i-1,j}^{(4)} \widetilde{K}_i^T(k) D_{i-1,j}^{(3)})]^T H_{i-1,j}\}\\
&= \widetilde{K}_i(k) - \mu \sum_{j=1}^{m} G_{i-1,j} \widetilde{\varphi}_{i-1}^T(k) H_{i-1,j}.
\end{aligned} \tag{3.26}$$

Taking the square of the norm on both sides of (3.25) and (3.26), it can be derived

$$\left\|\widetilde{K}_{1,i}(k+1)\right\|^2 = \left\|\widetilde{K}_i(k)\right\|^2 - 2\mu tr(\widetilde{K}_i^T(k)\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}) + \mu^2\left\|\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\right\|^2, \quad (3.27)$$

$$\begin{aligned}\left\|\widetilde{K}_{2,i}(k+1)\right\|^2 = &\left\|\widetilde{K}_i(k)\right\|^2 - 2\mu tr(\widetilde{K}_i^T(k)\sum_{j=1}^{m} G_{i-1,j}\widetilde{\varphi}_{i-1}^T(k)H_{i-1,j}) \\ &+ \mu^2\left\|\sum_{j=1}^{m} G_{i-1,j}\widetilde{\varphi}_{i-1}^T(k)H_{i-1,j}\right\|^2. \end{aligned} \quad (3.28)$$

The function $W(k)$ is defined as

$$W(k) = \sum_{i=1}^{\xi}\left\|\widetilde{K}_i(k)\right\|^2. \quad (3.29)$$

Thus, through (3.27)-(3.29) and Algorithm 1, we have

$$\begin{aligned}
&W(k+1) \\
=&\sum_{i=1}^{\xi}\left\|\widetilde{K}_i(k+1)\right\|^2 \\
=&\sum_{i=1}^{\xi}\left\|\frac{\widetilde{K}_{1,i}(k+1)+\widetilde{K}_{2,i}(k+1)}{2}\right\|^2 \\
\leq&\sum_{i=1}^{\xi}\left(\frac{1}{2}\left\|\widetilde{K}_{1,i}(k+1)\right\|^2 + \frac{1}{2}\left\|\widetilde{K}_{2,i}(k+1)\right\|^2\right) \\
=&\sum_{i=1}^{\xi}\Big[\left\|\widetilde{K}_i(k)\right\|^2 - \mu tr(\widetilde{K}_i^T(k)\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}) + \frac{1}{2}\mu^2\left\|\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\right\|^2 \\
&- \mu tr(\widetilde{K}_i^T(k)\sum_{j=1}^{m} G_{i-1,j}\widetilde{\varphi}_{i-1}^T(k)H_{i-1,j}) + \frac{1}{2}\mu^2\left\|\sum_{j=1}^{m} G_{i-1,j}\widetilde{\varphi}_{i-1}^T(k)H_{i-1,j}\right\|^2\Big] \\
=&\sum_{i=1}^{\xi}\Big[\left\|\widetilde{K}_i(k)\right\|^2 - \mu tr(\widetilde{K}_i^T(k)\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}) + \frac{1}{2}\mu^2\left\|\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\right\|^2 \\
&- \mu tr(\widetilde{K}_{i+1}^T(k)\sum_{j=1}^{m} G_{i,j}\widetilde{\varphi}_i^T(k)H_{i,j}) + \frac{1}{2}\mu^2\left\|\sum_{j=1}^{m} G_{i,j}\widetilde{\varphi}_i^T(k)H_{i,j}\right\|^2\Big] \\
=&\sum_{i=1}^{\xi}\Big[\left\|\widetilde{K}_i(k)\right\|^2 - \mu tr(\widetilde{\varphi}_i(k)\sum_{j=1}^{m} E_{i,j}\widetilde{K}_i^T(k)F_{i,j} + \widetilde{\varphi}_i^T(k)\sum_{j=1}^{m} G_{i,j}\widetilde{K}_{i+1}^T(k)H_{i,j}) \\
&+ \frac{1}{2}\mu^2\left\|\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\right\|^2 + \frac{1}{2}\mu^2\left\|\sum_{j=1}^{m} G_{i,j}\widetilde{\varphi}_i^T(k)H_{i,j}\right\|^2\Big]
\end{aligned}$$

$$\leq W(k) - \mu \sum_{i=1}^{\xi} \|\widetilde{\varphi}_i(k)\|^2 + \frac{1}{2}\mu^2 \sum_{i=1}^{\xi} [\sum_{j=1}^{m} (\|E_{i,j}\|^2 \|F_{i,j}\|^2 + \|G_{i,j}\|^2 \|H_{i,j}\|^2)] \|\widetilde{\varphi}_i(k)\|^2$$

$$= W(k) - \frac{1}{2}\mu\{2 - \mu \sum_{i=1}^{\xi} \Big[\sum_{j=1}^{m} (\|E_{i,j}\|^2 \|F_{i,j}\|^2 + \|G_{i,j}\|^2 \|H_{i,j}\|^2)\Big]\} \sum_{i=1}^{\xi} \|\widetilde{\varphi}_i(k)\|^2$$

$$\leq W(0) - \frac{1}{2}\mu\{2 - \mu \sum_{i=1}^{\xi} \Big[\sum_{j=1}^{m} (\|E_{i,j}\|^2 \|F_{i,j}\|^2 + \|G_{i,j}\|^2 \|H_{i,j}\|^2)\Big]\} \sum_{t=0}^{k} \sum_{i=1}^{\xi} \|\widetilde{\varphi}_i(t)\|^2.$$

Furthermore, if the convergence number μ satisfies (3.22), it can be obtained

$$\sum_{t=0}^{k} \sum_{i=1}^{\xi} \|\widetilde{\varphi}_i(t)\|^2 < \infty. \tag{3.30}$$

Because of the conditions that the series converges, when $t \to \infty$, it has

$$\sum_{i=1}^{\xi} \|\widetilde{\varphi}_i(t)\|^2 \to 0. \tag{3.31}$$

Then it follows from (3.25) and (3.31) that

$$\lim_{t\to\infty} \widetilde{\varphi}_i(t) = 0, \tag{3.32}$$

or

$$\lim_{t\to\infty} \Big[\sum_{j=1}^{m} (D_{i,j}^{(1)} \widetilde{K}_i(t) D_{i,j}^{(2)} + D_{i,j}^{(4)} \widetilde{K}_{i+1}^T(t) D_{i,j}^{(3)})\Big] = 0. \tag{3.33}$$

Now, due to Lemma 2.3, it gets

$$\lim_{t\to\infty} \widetilde{K}_i(t) = 0. \tag{3.34}$$

This completes the proof of Theorem 3.1. □

To prove the following Theorem 3.2, we define several symbols

$$\mathcal{M} = diag\Big(\sum_{j=1}^{m} (F_{1,j}^T \otimes E_{1,j}), \sum_{j=1}^{m} (F_{2,j}^T \otimes E_{2,j}), \cdots, \sum_{j=1}^{m} (F_{\xi,j}^T \otimes E_{\xi,j})\Big), \tag{3.35}$$

$$\mathcal{N} = diag\Big(\sum_{j=1}^{m} (D_{1,j}^{(2)} \otimes D_{1,j}^{(1)}), \sum_{j=1}^{m} (D_{2,j}^{(2)} \otimes D_{2,j}^{(1)}), \cdots, \sum_{j=1}^{m} (D_{\xi,j}^{(2)} \otimes D_{\xi,j}^{(1)})\Big), \tag{3.36}$$

$$\mathcal{P} = diag\Big(\sum_{j=1}^{m} (H_{\xi,j}^T \otimes G_{\xi,j})P, \sum_{j=1}^{m} (H_{1,j}^T \otimes G_{1,j})P, \cdots, \sum_{j=1}^{m} (H_{\xi-1,j}^T \otimes G_{\xi-1,j})P\Big), \tag{3.37}$$

$$\mathcal{Q} = diag\Big(\sum_{j=1}^{m} (D_{\xi,j}^{(3)} \otimes D_{\xi,j}^{(4)})P, \sum_{j=1}^{m} (D_{1,j}^{(3)} \otimes D_{1,j}^{(4)})P, \cdots, \sum_{j=1}^{m} (D_{\xi-1,j}^{(3)} \otimes D_{\xi-1,j}^{(4)})P\Big), \tag{3.38}$$

$$\mathcal{R} = \begin{pmatrix} 0 & \sum\limits_{j=1}^{m}(D_{1,j}^{(3)} \otimes D_{1,j}^{(4)})P & & 0 \\ & & \ddots & \\ 0 & 0 & & \sum\limits_{j=1}^{m}(D_{\xi-1,j}^{(3)} \otimes D_{\xi-1,j}^{(4)})P \\ \sum\limits_{j=1}^{m}(D_{\xi,j}^{(3)} \otimes D_{\xi,j}^{(4)})P & 0 & \cdots & 0 \end{pmatrix}, \tag{3.39}$$

$$\mathcal{V} = \begin{pmatrix} 0 & \cdots & 0 & \sum\limits_{j=1}^{m}(D_{\xi,j}^{(2)} \otimes D_{\xi,j}^{(1)}) \\ \sum\limits_{j=1}^{m}(D_{1,j}^{(2)} \otimes D_{1,j}^{(1)}) & & 0 & 0 \\ & \ddots & & \\ 0 & & \sum\limits_{j=1}^{m}(D_{\xi-1,j}^{(2)} \otimes D_{\xi-1,j}^{(1)}) & 0 \end{pmatrix}. \tag{3.40}$$

Theorem 3.2. *If A is a full-row rank matrix, then the iterative solution $K(k) = (K_1(k), K_2(k), \cdots, K_\xi(k))$ given by Algorithm 1 (RRJGI) converges to the unique solution $K^* = (K_1^*, K_2^*, \cdots, K_\xi^*)$ for arbitrary initial matrices $K(0) = (K_1(0), K_2(0), \cdots, K_\xi(0))$, if and only if*

$$0 < \mu < \frac{4}{\lambda_{\max}(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})}. \tag{3.41}$$

Proof. From Algorithm 1 (RRJGI), (3.25) and (3.26), we have

$$\begin{aligned} \widetilde{K}_i(k+1) &= \frac{\widetilde{K}_{1,i}(k+1) + \widetilde{K}_{2,i}(k+1)}{2} \\ &= \widetilde{K}_i(k) - \frac{\mu}{2}\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j} - \frac{\mu}{2}\sum_{j=1}^{m} G_{i-1,j}\widetilde{\varphi}_{i-1}^T(k)H_{i-1,j}. \end{aligned} \tag{3.42}$$

Taking the *vec* operator on both sides of (3.24) and (3.42), we can obtain

$$vec(\widetilde{\varphi}_i(k)) = \sum_{j=1}^{m}(D_{i,j}^{(2)} \otimes D_{i,j}^{(1)})vec(\widetilde{K}_i(k)) + \sum_{j=1}^{m}(D_{i,j}^{(3)} \otimes D_{i,j}^{(4)})Pvec(\widetilde{K}_{i+1}(k)), \tag{3.43}$$

and

$$\begin{aligned} vec(\widetilde{K}_i(k+1)) &= vec(\widetilde{K}_i(k)) - \frac{\mu}{2}\sum_{j=1}^{m}(F_{i,j}^T \otimes E_{i,j})vec(\widetilde{\varphi}_i(k)) \\ &\quad - \frac{\mu}{2}\sum_{j=1}^{m}(H_{i-1,j}^T \otimes G_{i-1,j})Pvec(\widetilde{\varphi}_{i-1}(k)) \\ &= vec(\widetilde{K}_i(k)) - \frac{\mu}{2}\sum_{j=1}^{m}(F_{i,j}^T \otimes E_{i,j})\sum_{j=1}^{m}(D_{i,j}^{(2)} \otimes D_{i,j}^{(1)})vec(\widetilde{K}_i(k)) \\ &\quad - \frac{\mu}{2}\sum_{j=1}^{m}(F_{i,j}^T \otimes E_{i,j})\sum_{j=1}^{m}(D_{i,j}^{(3)} \otimes D_{i,j}^{(4)})Pvec(\widetilde{K}_{i+1}(k)) \end{aligned}$$

$$- \frac{\mu}{2} \sum_{j=1}^{m} (H_{i-1,j}^T \otimes G_{i-1,j}) P \sum_{j=1}^{m} (D_{i-1,j}^{(2)} \otimes D_{i-1,j}^{(1)}) vec(\widetilde{K}_{i-1}(k))$$

$$- \frac{\mu}{2} \sum_{j=1}^{m} (H_{i-1,j}^T \otimes G_{i-1,j}) P \sum_{j=1}^{m} (D_{i-1,j}^{(3)} \otimes D_{i-1,j}^{(4)}) P vec(\widetilde{K}_i(k)). \tag{3.44}$$

It follows from (3.35)-(3.40) and (3.44) that

$$\begin{pmatrix} vec(\widetilde{K}_1(k+1)) \\ vec(\widetilde{K}_2(k+1)) \\ \vdots \\ vec(\widetilde{K}_\xi(k+1)) \end{pmatrix} = [I - \frac{\mu}{2}(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})] \begin{pmatrix} vec(\widetilde{K}_1(k)) \\ vec(\widetilde{K}_2(k)) \\ \vdots \\ vec(\widetilde{K}_\xi(k)) \end{pmatrix}. \tag{3.45}$$

Eq. (3.45) shows that Algorithm 1 is convergent if and only if

$$\rho[I - \frac{\mu}{2}(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})] < 1. \tag{3.46}$$

Then, it is obvious that

$$\begin{aligned} &\lambda[I - \frac{\mu}{2}(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})] \\ &= \{1 - \frac{\mu}{2}\lambda_s(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ}), s = 1, 2, \cdots, r\}, \end{aligned} \tag{3.47}$$

where $r = rank(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})$. Since $\rho[I - \frac{\mu}{2}(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})] < 1$, it can be derived

$$-1 < 1 - \frac{\mu}{2}\lambda_s(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})] < 1, \tag{3.48}$$

i.e.

$$0 < \mu < \frac{4}{\lambda_s(\mathcal{MN} + \mathcal{MR} + \mathcal{PV} + \mathcal{PQ})}, s = 1, 2, \cdots, r, \tag{3.49}$$

and using the yields from the intersection (3.41). Thus, the proof of the conclusion is complete. □

Theorem 3.3. *Let A be a full-row rank matrix, the iterative solution $K(k) = (K_1(k), K_2(k), \cdots, K_\xi(k))$ given by Algorithm 2(RRAJGI) converges to the unique solution $K^*(k) = (K_1^*(k), K_2^*(k), \cdots, K_\xi^*(k))$ for arbitrary initial matrices $K(0) = (K_1(0), K_2(0), \cdots, K_\xi(0))$, if μ satisfies*

$$0 < \mu < \min \left\{ \frac{2}{\omega_2 \sum_{i=1}^{\xi} \sum_{j=1}^{m} \left\| D_{i,j}^{(1)} \right\|^2 \left\| D_{i,j}^{(2)} \right\|^2}, \frac{2}{(1-\omega_2) \sum_{i=1}^{\xi} \sum_{j=1}^{m} \left\| D_{i,j}^{(3)} \right\|^2 \left\| D_{i,j}^{(4)} \right\|^2} \right\}. \tag{3.50}$$

Proof. The error matrices are defined as

$$\widetilde{K}_i(k) = K_i(k) - K_i^*, \qquad \widetilde{\widehat{K}}_i(k) = \widehat{K}_i(k) - K_i^*, \tag{3.51}$$

$$\widetilde{K}_{1,i}(k) = K_{1,i}(k) - K_i^*, \qquad \widetilde{K}_{2,i}(k) = K_{2,i}(k) - K_i^*, \tag{3.52}$$

and

$$\widetilde{\varphi}_i(k)=\sum_{j=1}^{m}(D_{i,j}^{(1)}\widetilde{K}_i(k)D_{i,j}^{(2)}+D_{i,j}^{(4)}\widetilde{K}_{i+1}^{T}(k)D_{i,j}^{(3)}), \tag{3.53}$$

$$\widetilde{\delta}_i(k)=\sum_{j=1}^{m}(D_{i,j}^{(1)}\widetilde{\widehat{K}}_i(k)D_{i,j}^{(2)}+D_{i,j}^{(4)}\widetilde{\widehat{K}}_{i+1}^{T}(k)D_{i,j}^{(3)}). \tag{3.54}$$

From (3.51)-(3.54) and Algorithm 2, we can get the following relations

$$\widetilde{K}_{1,i}(k+1)=\widetilde{K}_i(k)-\mu\omega_2\sum_{j=1}^{m}E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}, \tag{3.55}$$

$$\widetilde{K}_{2,i}(k+1)=\widetilde{\widehat{K}}_i(k)-\mu(1-\omega_2)\sum_{j=1}^{m}G_{i-1,j}\widetilde{\delta}_{i-1}^{T}(k)H_{i-1,j}. \tag{3.56}$$

Taking the square of the norm on both sides of (3.55) and (3.56), it has

$$\left\|\widetilde{K}_{1,i}(k+1)\right\|^2=\left\|\widetilde{K}_i(k)\right\|^2-2\mu\omega_2 tr\Big(\widetilde{K}_i^T(k)\sum_{j=1}^{m}E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\Big)+\mu^2\omega_2^2\left\|\sum_{j=1}^{m}E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\right\|^2, \tag{3.57}$$

$$\begin{aligned}\left\|\widetilde{K}_{2,i}(k+1)\right\|^2=&\left\|\widetilde{\widehat{K}}_i(k)\right\|^2-2\mu(1-\omega_2)tr\Big(\widetilde{\widehat{K}}_i^T(k)\sum_{j=1}^{m}G_{i-1,j}\widetilde{\delta}_{i-1}^{T}(k)H_{i-1,j}\Big)\\&+\mu^2(1-\omega_2)^2\left\|\sum_{j=1}^{m}G_{i-1,j}\widetilde{\delta}_{i-1}^{T}(k)H_{i-1,j}\right\|^2.\end{aligned} \tag{3.58}$$

The function $W(k)$ is defined as

$$W(k)=\sum_{i=1}^{\xi}\left\|\widetilde{K}_i(k)\right\|^2. \tag{3.59}$$

Thus, through (3.57)-(3.59) and Algorithm 2, we have

$$\begin{aligned}&W(k+1)\\=&\sum_{i=1}^{\xi}\left\|\widetilde{K}_i(k+1)\right\|^2\\=&\sum_{i=1}^{\xi}\left\|(1-\omega_2)\widetilde{K}_{1,i}(k+1)+\omega_2\widetilde{K}_{2,i}(k+1)\right\|^2\\\leq&\sum_{i=1}^{\xi}\Big[(1-\omega_2)^2\left\|\widetilde{K}_{1,i}(k+1)\right\|^2+\omega_2^2\left\|\widetilde{K}_{2,i}(k+1)\right\|^2\Big]\\=&\sum_{i=1}^{\xi}\Big[(1-\omega_2)^2\left\|\widetilde{K}_i(k)\right\|^2+\omega_2^2\left\|\widetilde{\widehat{K}}_i(k)\right\|^2-2\mu\omega_2(1-\omega_2)^2 tr(\widetilde{K}_i^T(k)\sum_{j=1}^{m}E_{i,j}\widetilde{\varphi}_i(k)F_{i,j})\end{aligned}$$

$$
\begin{aligned}
&+ \mu^2\omega_2^2(1-\omega_2)^2\left\|\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\right\|^2 - 2\mu\omega_2^2(1-\omega_2)tr(\widetilde{\widehat{K}}_i^T(k)\sum_{j=1}^{m} G_{i-1,j}\widetilde{\delta}_{i-1}^T(k)H_{i-1,j})\\
&+ \mu^2\omega_2^2(1-\omega_2)^2\left\|\sum_{j=1}^{m} G_{i-1,j}\widetilde{\delta}_{i-1}^T(k)H_{i-1,j}\right\|^2 \Big]\\
=&\sum_{i=1}^{\xi}\Big[(1-\omega_2)^2\left\|\widetilde{K}_i(k)\right\|^2 + \omega_2^2\left\|\widetilde{\widehat{K}}_i(k)\right\|^2 - 2\mu\omega_2(1-\omega_2)^2 tr(\widetilde{\varphi}_i(k)\sum_{j=1}^{m} E_{i,j}\widetilde{K}_i^T(k)F_{i,j})\\
&+ \mu^2\omega_2^2(1-\omega_2)^2\left\|\sum_{j=1}^{m} E_{i,j}\widetilde{\varphi}_i(k)F_{i,j}\right\|^2 - 2\mu\omega_2^2(1-\omega_2)tr(\widetilde{\delta}_i^T(k)\sum_{j=1}^{m} G_{i,j}\widetilde{\widehat{K}}_{i+1}^T(k)H_{i,j})\\
&+ \mu^2\omega_2^2(1-\omega_2)^2\left\|\sum_{j=1}^{m} G_{i,j}\widetilde{\delta}_i^T(k)H_{i,j}\right\|^2 \Big]
\end{aligned}
$$

$$
\begin{aligned}
\leq&\sum_{i=1}^{\xi}\Big[(1-\omega_2)^2\left\|\widetilde{K}_i(k)\right\|^2 + \omega_2^2\left\|\widetilde{\widehat{K}}_i(k)\right\|^2 - 2\mu\omega_2(1-\omega_2)^2\|\widetilde{\varphi}_i(k)\|^2\\
&+ \mu^2\omega_2^2(1-\omega_2)^2\sum_{j=1}^{m}\|E_{i,j}\|^2\|F_{i,j}\|^2\|\widetilde{\varphi}_i(k)\|^2\\
&- 2\mu\omega_2^2(1-\omega_2)\left\|\widetilde{\delta}_i^T(k)\right\|^2 + \mu^2\omega_2^2(1-\omega_2)^2\sum_{j=1}^{m}\|G_{i,j}\|^2\|H_{i,j}\|^2\left\|\widetilde{\delta}_i^T(k)\right\|^2\Big]\\
=&(1-\omega_2)^2W(k) + \omega_2^2\sum_{i=1}^{\xi}\left\|\widetilde{\widehat{K}}_i(k)\right\|^2 - \mu\omega_2(1-\omega_2)^2\Big[2-\mu\omega_2\sum_{i=1}^{\xi}\sum_{j=1}^{m}\|E_{i,j}\|^2\|F_{i,j}\|^2\Big]\sum_{i=1}^{\xi}\|\widetilde{\varphi}_i(k)\|^2\\
&- \mu\omega_2^2(1-\omega_2)\Big[2-\mu(1-\omega_2)\sum_{i=1}^{\xi}\sum_{j=1}^{m}\|G_{i,j}\|^2\|H_{i,j}\|^2\Big]\sum_{i=1}^{\xi}\left\|\widetilde{\delta}_i^T(k)\right\|^2\\
\leq&(1-\omega_2)^2W(0) + \omega_2^2\sum_{i=1}^{\xi}\sum_{t=0}^{k}\left\|\widetilde{\widehat{K}}_i(t)\right\|^2\\
&- \mu\omega(1-\omega_2)^2\Big[2-\mu\omega_2\sum_{i=1}^{\xi}\sum_{j=1}^{m}\|E_{i,j}\|^2\|F_{i,j}\|^2\Big]\sum_{i=1}^{\xi}\sum_{t=0}^{k}\|\widetilde{\varphi}_i(t)\|^2\\
&- \mu\omega_2^2(1-\omega_2)\Big[2-\mu(1-\omega_2)\sum_{i=1}^{\xi}\sum_{j=1}^{m}\|G_{i,j}\|^2\|H_{i,j}\|^2\Big]\sum_{i=1}^{\xi}\sum_{t=0}^{k}\left\|\widetilde{\delta}_i^T(t)\right\|^2.
\end{aligned}
$$

Furthermore, if the convergence number μ satisfies (3.50), it can be obtained

$$
\sum_{t=0}^{k}\sum_{i=1}^{\xi}\|\widetilde{\varphi}_i(t)\|^2 < \infty, \qquad \sum_{t=0}^{k}\sum_{i=1}^{\xi}\left\|\widetilde{\delta}_i^T(t)\right\|^2 < \infty. \tag{3.60}
$$

Because of the conditions that the series converges, when $t \to \infty$, it is derived

$$\sum_{i=1}^{\xi} \|\widetilde{\varphi}_i(t)\|^2 \to 0, \qquad \sum_{i=1}^{\xi} \left\|\widetilde{\delta}_i^T(t)\right\|^2 \to 0. \tag{3.61}$$

It follows from (3.53), (3.54) and (3.61) that

$$\lim_{t\to\infty} \widetilde{\varphi}_i(k-t) = 0, \qquad \lim_{t\to\infty} \widetilde{\delta}_i^T(k-t) = 0. \tag{3.62}$$

or

$$\lim_{t\to\infty} \Big[\sum_{j=1}^{m} (D_{i,j}^{(1)} \widetilde{K}_i(t) D_{i,j}^{(2)} + D_{i,j}^{(4)} \widetilde{K}_{i+1}^T(t) D_{i,j}^{(3)})\Big] = 0, \tag{3.63}$$

$$\lim_{t\to\infty} \Big[\sum_{j=1}^{m} (D_{i,j}^{(1)} \widetilde{\widetilde{K}}_i(t) D_{i,j}^{(2)} + D_{i,j}^{(4)} \widetilde{\widetilde{K}}_{i+1}^T(t) D_{i,j}^{(3)})\Big]^T = 0. \tag{3.64}$$

Now, from Lemma 2.3, it gets

$$\lim_{t\to\infty} \widetilde{K}_i(t) = 0. \tag{3.65}$$

This completes the proof of Theorem 3.3. □

3.2 The CRJGI algorithm, the CRAJGI algorithm and convergence analysis

In the following, first we introduce the full-column rank Jacobi gradient based iterative (CRJGI) algorithm and the full-collumn rank accelerated Jacobi gradient based iterative algorithm (CRAJGI) for solving Eq. (1.9). Then we give the convergence analysis on these two iterative algorithms.

Algorithm 3 (The CRJGI algorithm)

Step 1. Given the coefficient matrices $E_{i,j}, G_{i,j} \in R^{m\times m}, F_{i,j}, H_{i,j} \in R^{n\times n}$ and $M_i \in R^{m\times n}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$, choose an appropriate convergence number μ and the initial matrices $Y_i(0) \in R^{m\times n}$.

Step 2. Set $Y_{i+\xi}(0) = Y_i(0), E_{i+\xi,j} = E_{i,j}, F_{i+\xi,j} = F_{i,j}, G_{i+\xi,j} = G_{i,j}, H_{i+\xi,j} = H_{i,j}, M_{i+\xi} = M_i, D_{i+\xi,j}^{(1)} = D_{i,j}^{(1)}, D_{i+\xi,j}^{(2)} = D_{i,j}^{(2)}, D_{i+\xi,j}^{(3)} = D_{i,j}^{(3)}$, and $D_{i+\xi,j}^{(4)} = D_{i,j}^{(4)}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$. Let $k := 0$.

Step 3. If $\delta(k) = \frac{\sum_{i=1}^{\xi} ||M_i - \sum_{j=1}^{m} (E_{i,j}Y_i(k)F_{i,j} + G_{i,j}Y_{i+1}^T(k)H_{i,j})||^2}{\sum_{i=1}^{\xi} ||M_i||^2} < \varepsilon$, stop; otherwise, go to Step 4.

Step 4. Compute the following sequences

$$Y_{1,i}(k+1) = Y_i(k) + \mu \sum_{j=1}^{m} D_{i,j}^{(1)}[M_i - \sum_{j=1}^{m} (E_{i,j}Y_i(k)F_{i,j} + G_{i,j}Y_{i+1}^T(k)H_{i,j})]D_{i,j}^{(2)},$$

$$\begin{aligned} Y_{2,i}(k+1) = Y_i(k) + \mu \sum_{j=1}^{m} D_{i-1,j}^{(4)}[M_{i-1} - \sum_{j=1}^{m} (&E_{i-1,j}Y_{i-1}(k)F_{i-1,j} \\ &+ G_{i-1,j}Y_i^T(k)H_{i-1,j})]^T D_{i-1,j}^{(3)}, \end{aligned}$$

$$Y_i(k+1) = \frac{Y_{1,i}(k+1) + Y_{2,i}(k+1)}{2},$$
$$Y_{i+\xi}(k+1) = Y_i(k+1).$$

Step 5. Let $k := k+1$, go to Step 3.

Similar to the construction of RRAJGI algorithm, we introduce an appropriate factor ω_1, $0 < \omega_1 < 1$ on the basis of CRJGI algorithm, and propose the following full-column rank accelerated Jacobi gradient based iterative (CRAJGI) algorithm for solving Eq. (1.9).

Algorithm 4 (The CRAJGI algorithm)

Step 1. Given the coefficient matrices $E_{i,j}, G_{i,j} \in R^{m\times m}, F_{i,j}, H_{i,j} \in R^{n\times n}$ and $M_i \in R^{m\times n}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$, choose the initial matrices $Y_i(0), Y_{2,i}(0) \in R^{m\times n}$.

Step 2. Set $Y_{i+\xi}(0) = Y_i(0), E_{i+\xi,j} = E_{i,j}, F_{i+\xi,j} = F_{i,j}, G_{i+\xi,j} = G_{i,j}, H_{i+\xi,j} = H_{i,j}, M_{i+\xi} = M_i, D^{(1)}_{i+\xi,j} = D^{(1)}_{i,j}, D^{(2)}_{i+\xi,j} = D^{(2)}_{i,j}, D^{(3)}_{i+\xi,j} = D^{(3)}_{i,j}$, and $D^{(4)}_{i+\xi,j} = D^{(4)}_{i,j}$ for $i \in \overline{1,\xi}$, $j \in \overline{1,m}$. Let $k := 0$.

Step 3. If $\delta(k) = \frac{\sum_{i=1}^{\xi} ||M_i - \sum_{j=1}^{m}(E_{i,j}Y_i(k)F_{i,j} + G_{i,j}Y^T_{i+1}(k)H_{i,j})||^2}{\sum_{i=1}^{\xi}||M_i||^2} < \varepsilon$, stop; otherwise, go to Step 4.

Step 4. Compute the following sequences

$$Y_{1,i}(k+1) = Y_i(k) + \mu\omega_1 \sum_{j=1}^{m} D^{(1)}_{i,j}[M_i - \sum_{j=1}^{m}(E_{i,j}Y_i(k)F_{i,j} + G_{i,j}Y^T_{i+1}(k)H_{i,j})]D^{(2)}_{i,j},$$
$$\widehat{Y}_i(k) = (1-\omega_1)Y_{1,i}(k+1) + \omega_1 Y_{2,i}(k),$$
$$\widehat{Y}_{i+\xi}(k) = \widehat{Y}_i(k),$$
$$Y_{2,i}(k+1) = \widehat{Y}_i(k) + \mu(1-\omega_1)\sum_{j=1}^{m} D^{(4)}_{i-1,j}[M_{i-1} - \sum_{j=1}^{m}(E_{i-1,j}\widehat{Y}_{i-1}(k)F_{i-1,j}$$
$$+ G_{i-1,j}\widehat{Y}^T_i(k)H_{i-1,j})]^T D^{(3)}_{i-1,j},$$
$$Y_i(k+1) = (1-\omega_1)Y_{1,i}(k+1) + \omega_1 Y_{2,i}(k+1),$$
$$Y_{i+\xi}(k+1) = Y_i(k+1).$$

Step 5. Let $k := k+1$, go to Step 3.

Theorem 3.4. *Assumed that A be a full-column rank matrix, then the iterative solution $Y(k) = (Y_1(k), Y_2(k), \cdots, Y_\xi(k))$ given by Algorithm 3 (CRJGI) converges to the unique solution $Y^*(k) = (Y^*_1(k), Y^*_2(k), \cdots, Y^*_\xi(k))$ of Eq. (1.9) for arbitrary initial matrix group $Y(0) = (Y_1(0), Y_2(0), \cdots, Y_\xi(0))$, if*

$$0 < \mu < \frac{2}{\sum_{i=1}^{\xi}\sum_{j=1}^{m}\left(\left\|D^{(1)}_{i,j}\right\|^2\left\|D^{(2)}_{i,j}\right\|^2 + \left\|D^{(3)}_{i,j}\right\|^2\left\|D^{(4)}_{i,j}\right\|^2\right)}. \tag{3.66}$$

Proof. We can prove this result by using the same line as Theorem 3.1. Hence, it has been omitted here. □

Similarly, in order to prove the following Theorem 3.5, we define the following notations

$$\mathcal{B}=\begin{pmatrix} 0 & \sum\limits_{j=1}^{m}(H_{1,j}^{T}\otimes G_{1,j})P & & 0 \\ & & \ddots & \\ 0 & 0 & & \sum\limits_{j=1}^{m}(H_{\xi-1,j}^{T}\otimes G_{\xi-1,j})P \\ \sum\limits_{j=1}^{m}(H_{\xi,j}^{T}\otimes G_{\xi,j})P & 0 & \cdots & 0 \end{pmatrix}, \tag{3.67}$$

$$\mathcal{C}=\begin{pmatrix} 0 & \cdots & 0 & \sum\limits_{j=1}^{m}(F_{\xi,j}^{T}\otimes E_{\xi,j}) \\ \sum\limits_{j=1}^{m}(F_{1,j}^{T}\otimes E_{1,j}) & & 0 & 0 \\ & \ddots & & \\ 0 & \sum\limits_{j=1}^{m}(F_{\xi-1,j}^{T}\otimes E_{\xi-1,j}) & & 0 \end{pmatrix}. \tag{3.68}$$

Theorem 3.5. *If A is a full-column rank matrix, $Y^*(k)=(Y_1^*(k),Y_2^*(k),\cdots,Y_\xi^*(k))$ is the unique solution group of Eq. (1.9), then the Algorithm 3 (CRJGI) obtains $\lim_{k\to\infty}Y_i(k)=Y_i^*(k), i\in\overline{1,\xi}$, for arbitrary initial matrix group $Y(0)=(Y_1(0),Y_2(0),\cdots,Y_\xi(0))$, if and only if*

$$0<\mu<\frac{4}{\lambda_{\max}(\mathcal{N}\mathcal{M}+\mathcal{N}\mathcal{B}+\mathcal{Q}\mathcal{C}+\mathcal{Q}\mathcal{P})}. \tag{3.69}$$

Proof. We can use the same line as Theorem 3.2 to demonstrate this result. So, it is not included here. □

Theorem 3.6. *Supposed that A be a full-column rank matrix, then the iterative solution $Y(k)=(Y_1(k),Y_2(k),\cdots,Y_\xi(k))$ given by Algorithm 4 (CRAJGI) converges to the unique solution $Y^*(k)=(Y_1^*(k),Y_2^*(k),\cdots,Y_\xi^*(k))$ of Eq. (1.9) for arbitrary initial matrix group $Y(0)=(Y_1(0),Y_2(0),\cdots,Y_\xi(0))$, if*

$$0<\mu<\min\left\{\frac{2}{\omega_1\sum\limits_{i=1}^{\xi}\sum\limits_{j=1}^{m}\left\|D_{i,j}^{(1)}\right\|^2\left\|D_{i,j}^{(2)}\right\|^2},\frac{2}{(1-\omega_1)\sum\limits_{i=1}^{\xi}\sum\limits_{j=1}^{m}\left\|D_{i,j}^{(3)}\right\|^2\left\|D_{i,j}^{(4)}\right\|^2}\right\}. \tag{3.70}$$

Proof. We can prove this result by using the same line as Theorem 3.3. Therefore, it has been omitted here. □

4 Numerical experiments

In this section, we give two examples to illustrate the performance of the proposed algorithms. All algorithms are calculated using MATLAB R2020a.

Example 4.1. Consider the discrete-time periodic Sylvester transpose matrix equations

$$A_iX_iB_i + C_iX_{i+1}^TE_i = G_i, i = 1, 2,$$

where the coefficient matrices are given by

$$\begin{aligned}
A1 &= tril\left(rand\left(m\right), m\right) + diag\left(1.5 + diag\left(rand\left(m\right)\right)\right),\\
B1 &= -triu\left(rand\left(m\right), m\right) - b \times diag\left(2.6 + diag\left(rand\left(m\right)\right)\right),\\
C1 &= tril\left(rand\left(m\right), m\right) - diag\left(1 + diag\left(rand\left(m\right)\right)\right),\\
E1 &= -triu\left(rand\left(m\right), m\right) + diag\left(2 + diag\left(rand\left(m\right)\right)\right),\\
G1 &= rand\left(m\right) - eye\left(m\right) \times b,\\
A2 &= triu\left(rand\left(m\right), m\right) + diag\left(1.8 + diag\left(rand\left(m\right)\right)\right),\\
B2 &= -tril\left(rand\left(m\right), m\right) + b \times diag\left(4.4 + diag\left(rand\left(m\right)\right)\right),\\
C2 &= triu\left(rand\left(m\right), m\right) + diag\left(2.8 + diag\left(rand\left(m\right)\right)\right),\\
E2 &= -tril\left(rand\left(m\right), m\right) + diag\left(3.4 + diag\left(rand\left(m\right)\right)\right),\\
G2 &= -rand\left(m\right) - eye\left(m\right) \times b.
\end{aligned}$$

The initial matrices are chosen as $X_i(0) = zeros(m), i = 1, 2$ and the iterative residual is defined as

$$r_i(k) := \log_{10}\left\|G_i - A_iX_i(k)B_i - C_iX_{i+1}^T(k)E_i\right\|, i = 1, 2,$$

where $X_i(k)$ are the kth iterative solutions of the Algorithms 1-4. In this example, let $m = 4, b = 4$.

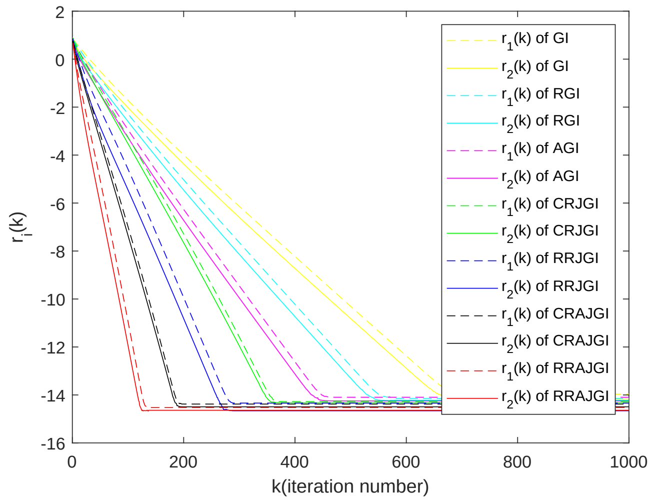

Fig. 1: Comparison of the convergence curves for Example 4.1.

Fig. 1 compares the convergence curves of GI algorithm, RGI algorithm, AGI algorithm, CRJGI algorithm, RRJGI algorithm, CRAJGI algorithm, and RRAJGI algorithm. It can be seen from Fig. 1 that the four algorithms proposed in this paper are effective for solving matrix equations,

Table 1: Iterative steps, residual and CPU time of Fig.1

Method	Step	$r_1(k)$	$r_2(k)$	Time
GI algorithm	631	−13.0101	−13.5179	0.0140
RGI algorithm	554	−14.0144	−14.2214	0.0126
AGI algorithm	448	−14.0020	−14.2179	0.0121
CRJGI algorithm	354	−14.0002	−14.2403	0.0104
RRJGI algorithm	275	−14.0050	−14.6148	0.0100
CRAJGI algorithm	185	−14.0513	−14.3357	0.0095
RRAJGI algorithm	126	−14.0956	−14.6313	0.0092

which shows the correctness of the algorithms and the convergence speed of these four algorithms are obviously faster than other algorithms which also shows the superiority of the algorithms.

Table 1 compares the number of iterative steps, iterative residual and computation time of several algorithms. It is showed from the Table 1 that the four algorithms proposed in this paper have more advantages than other algorithms in terms of convergence efficiency, convergence accuracy and calculation time. Especially, the advantage of RRAJGI algorithm is more obvious, which shows that this algorithm can save a lot of computing time and storage space.

Example 4.2. Consider the discrete-time periodic Sylvester transpose matrix equations

$$\begin{cases} A_{11}X_1B_{11} + C_{11}X_2^TD_{11} + A_{12}X_1B_{12} + C_{12}X_2^TD_{12} = M_1, \\ A_{21}X_2B_{21} + C_{21}X_1^TD_{21} + A_{22}X_2B_{22} + C_{22}X_1^TD_{22} = M_2, \end{cases}$$

where the coefficient matrices are given by

$$\begin{aligned}
&A_{11} = tril\left(rand\left(m\right), m\right) + diag\left(2.5 + diag\left(rand\left(m\right)\right)\right), \\
&B_{11} = -triu\left(rand\left(m\right), m\right) - a * diag\left(1.6 + diag\left(rand\left(m\right)\right)\right), \\
&C_{11} = tril\left(rand\left(m\right), m\right) - diag\left(3.1 + diag\left(rand\left(m\right)\right)\right), \\
&D_{11} = -triu\left(rand\left(m\right), m\right) + diag\left(1 + diag\left(rand\left(m\right)\right)\right), \\
&A_{12} = tril\left(rand\left(m\right), m\right) + diag\left(1.5 + diag\left(rand\left(m\right)\right)\right), \\
&B_{12} = -triu\left(rand\left(m\right), m\right) - a * diag\left(0.6 + diag\left(rand\left(m\right)\right)\right), \\
&C_{12} = tril\left(rand\left(m\right), m\right) - diag\left(1 + diag\left(rand\left(m\right)\right)\right), \\
&D_{12} = -triu\left(rand\left(m\right), m\right) + diag\left(2.2 + diag\left(rand\left(m\right)\right)\right), \\
&M_1 = rand\left(m\right) - eye\left(m\right) * a, \\
&A_{21} = tril\left(rand\left(m\right), m\right) + diag\left(0.5 + diag\left(rand\left(m\right)\right)\right), \\
&B_{21} = -triu\left(rand\left(m\right), m\right) - a * diag\left(1.6 + diag\left(rand\left(m\right)\right)\right), \\
&C_{21} = tril\left(rand\left(m\right), m\right) - diag\left(1 + diag\left(rand\left(m\right)\right)\right), \\
&D_{21} = -triu\left(rand\left(m\right), m\right) + diag\left(2 + diag\left(rand\left(m\right)\right)\right), \\
&A_{22} = tril\left(rand\left(m\right), m\right) + diag\left(1.5 + diag\left(rand\left(m\right)\right)\right), \\
&B_{22} = -triu\left(rand\left(m\right), m\right) - a * diag\left(2.6 + diag\left(rand\left(m\right)\right)\right), \\
&C_{22} = tril\left(rand\left(m\right), m\right) - diag\left(1 + diag\left(rand\left(m\right)\right)\right),
\end{aligned}$$

$$D_{22} = -triu\left(rand\left(m\right), m\right) + diag\left(2 + diag\left(rand\left(m\right)\right)\right),$$
$$M_2 = rand\left(m\right) - eye\left(m\right) * a.$$

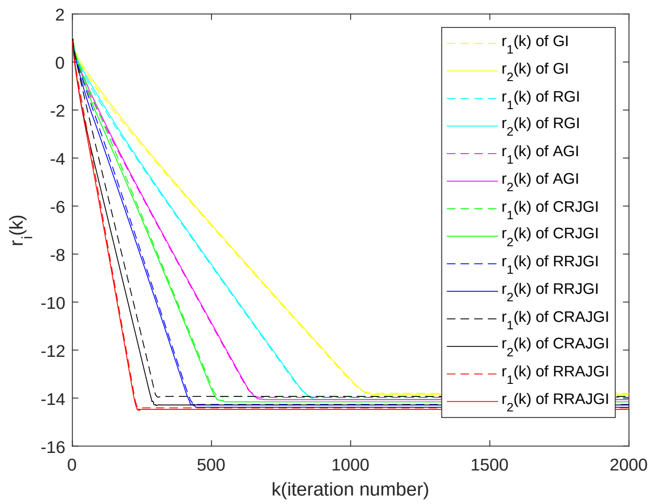

Fig. 2: Comparison of the convergence curves for Example 4.2.

The initial matrices are chosen as $X_i(0) = zeros(m), i = 1, 2$ and the iterative residual are defined as

$$r_1(k) := \log_{10} \left\| M_1 - A_{11}X_1(k)B_{11} - C_{11}X_2^T(k)D_{11} - A_{12}X_1(k)B_{12} - C_{12}X_2^T(k)D_{12} \right\|,$$
$$r_2(k) := \log_{10} \left\| M_2 - A_{21}X_2(k)B_{21} - C_{21}X_1^T(k)D_{21} - A_{22}X_2(k)B_{22} - C_{22}X_1^T(k)D_{22} \right\|,$$

where $X_i(k)$ are the kth iterative solutions of the Algorithms 1-4. In this example, let $m = 7, a = 5$.

Table 2: Iterative steps, residual and CPU time of Fig.2

Method	Step	$r_1(k)$	$r_2(k)$	Time
GI algorithm	1098	-13.7999	-13.8522	0.0686
RGI algorithm	854	-13.9414	-13.9744	0.0556
AGI algorithm	665	-13.9424	-14.0249	0.0518
CRJGI algorithm	541	-13.9487	-14.0915	0.0445
RRJGI algorithm	456	-14.2657	-14.3814	0.0375
CRAJGI algorithm	308	-13.9626	-14.2871	0.0293
RRAJGI algorithm	237	-14.4063	-14.5005	0.0260

Fig. 2 shows the comparison of the convergence curves of several algorithms. From Fig. 2 we can conclude that the iterative residual decreases gradually as the number of iterative steps increases, which indicates that these four algorithms are given by this paper can obtain the exact solutions of the equations under a limited number of iterative steps, and the algorithm proposed in this paper is better than other algorithms in terms of both the running time and the iterative residual.

Table 2 compares the number of iterative steps, iterative residuals and calculation time of several algorithms. Table 2 shows that RRJGI algorithm, RRAJGI algorithm, CRJGI algorithm and CRAJGI algorithm have significantly better computational efficiency than other algorithms. In addition, it can be obtained that the RRAJGI algorithm needs the minimum number of iterative steps and the shortest calculation time when convergence is achieved.

5 Application in state observer design of periodic linear systems

Example 5.1. We consider the following linear discrete-time periodic system

$$q_{k+1} = A_k q_k + B_k u_k, \tag{5.1}$$

where $A_k \in R^{n\times n}$ is the state matrix, $B_k \in R^{n\times r}$ is the input matrix, and both A_k, B_k are matrices with period T.

Periodic state observer based on state error feedback is the most widely used, which can be expressed as follows

$$\hat{x}_{t+1} = A_t\hat{x}_t + B_t u_t + L_t(y_t - \hat{y}_t), \tag{5.2}$$

where $\hat{x}_t \in R^n$ is the observer state, $\hat{y}_t = C_t\hat{x}_t$ is the observer output and $L_t \in R^{n\times m}$ is observer gain.

Obviously, the system (5.2) is equivalent to the following periodic closed-loop system

$$\hat{x}_{t+1} = (A_t - L_tC_t)\hat{x}_t + B_t u_t + L_t y_t, \tag{5.3}$$

its univalued matrix is written as

$$\Phi_A = \tilde{A}_{T-1}\tilde{A}_{T-2}\cdots\tilde{A}_0,$$

where $\tilde{A}_t = A_t - L_tC_t, t \in \overline{0, T-1}$. Then the problem of state-observer design for first-order linear periodic discrete system (5.1) can be described as follows.

Consider a fully observable first-order linear periodic discrete system (5.1) and find a periodic matrix $L_t \in R^{n\times m}$ so that the observer system (5.2) can give a asymptotic approximation to the state x_t of the system (5.1).

Next, we solve the periodic matrix L_t according to Algorithm 2 proposed in this paper, which is summarized as follows.

Algorithm 5 (The periodic state observer design in linear systems)

Step 1. Choose the appropriate matrices $F_t \in R^{n\times n}$ with periodic T that satisfies $\Lambda(\Phi_F) = \Gamma$ and $\Lambda(\Phi_F)\cap\Lambda(\Phi_A) = \emptyset$, and choose the periodic matrices $G_t \in R^{r\times n}$ satisfying that the matrix pair $(\tilde{A}_t, G_t)$ is fully observable, give an appropriate convergence factor μ and an appropriate relaxation factor ω, where $0 < \omega < 1$, $D_t^{(1)}$ and $D_t^{(2)}$ are the diagonal matrices corresponding to A_t^T and $-F_t$, respectively.

Step 2. Set tolerance error ε, the initial matrices are chosen as $Y_t(0), Y_{2,t}(0) \in R^{n\times n}, X_t(0) = D_t^{(1)}Y_t(0) + Y_{t+1}(0)D_t^{(2)}$, compute

$$R_t(0) = C_t^T G_t - A_t^T X_t(0) + X_{t+1}(0)F_t,$$
$$k := 0.$$

Step 3. When $||R_t(k)|| \leq \varepsilon$, compute

$$Y_{1,t}(k+1) = Y_t(k) + \mu\omega\{(C_t^T G_t)\Big/2 - A_t^T[D_t^{(1)}Y_t(k) + Y_{t+1}(k)D_t^{(2)}]\},$$
$$\widehat{Y}_t(k) = (1-\omega)Y_{1,t}(k+1) + \omega Y_{2,t}(k),$$
$$\widehat{Y}_{t+T}(k) = \widehat{Y}_t(k),$$
$$Y_{2,t}(k+1) = \widehat{Y}_t(k) + \mu(1-\omega)\{(C_{t-1}^T G_{t-1})\Big/2 - [D_{t-1}^{(1)}\widehat{Y}_{t-1}(k) + \widehat{Y}_t(k)D_{t-1}^{(2)}](-F_{t-1})\},$$
$$Y(k+1) = (1-\omega)Y_{1,t}(k+1) + \omega Y_{2,t}(k+1),$$
$$Y_{t+T}(k+1) = Y_t(k),$$
$$X_t(k+1) = D_t^{(1)}Y_t(k+1) + Y_{t+1}(k+1)D_t^{(2)},$$
$$X_{t+T}(k+1) = X_t(k),$$
$$k = k+1.$$

Step 4. Let $X_t^* = X_t(k)$, calculate the periodic state observer gain L_t

$$L_t = (G_t(X_t^*)^{-1})^T.$$

We know that the solution matrix X_t^* generated by Algorithm 5 are the solutions of the following Sylvester matrix equations

$$A_t^T X_t - X_{t+1}F_t = C_t^T G_t, t \in \overline{1,T}. \tag{5.4}$$

If we choose $T = 2$ and

$$A_1 = \begin{pmatrix} 3.3856 & 0.8913 & 0.8214 & 0.9218 \\ 0.2311 & 2.6150 & 0.4447 & 0.7382 \\ 0.6068 & 0.4565 & 2.3142 & 0.1763 \\ 0.4860 & 0.0185 & 0.7919 & 2.6525 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} -2.0185 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & -2.9667 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -4.2146 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -3.3796 \end{pmatrix},$$

$$F_1 = \begin{pmatrix} -1.0768 & 0.3412 & 0.8385 & 0.5466 \\ 0.6602 & -1.5457 & 0.5681 & 0.4449 \\ 0.3420 & 0.7271 & -0.8661 & 0.6946 \\ 0.2897 & 0.3093 & 0.7027 & -1.1401 \end{pmatrix},$$

$$F_2 = \begin{pmatrix} 2.7833 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 2.0592 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 2.8744 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 2.7889 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} -3.1951 & 0.3200 & 0.7446 & 0.6833 \\ 0.4983 & -3.1150 & 0.2679 & 0.2126 \\ 0.2140 & 0.7266 & -3.0729 & 0.8392 \\ 0.6435 & 0.4120 & 0.9334 & -3.4836 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} 3.8078 & 0.5869 & 0.7176 & 0.4418 \\ 0.0158 & 3.0083 & 0.6927 & 0.3533 \\ 0.0164 & 0.3676 & 3.4497 & 0.1536 \\ 0.1901 & 0.6315 & 0.4544 & 3.4150 \end{pmatrix},$$

$$G_1 = \begin{pmatrix} 6.1085 & 11.7455 & 9.8025 & 7.8521 \\ -7.8425 & -15.6279 & -20.9538 & -11.7712 \\ -11.1380 & -20.4830 & -25.4317 & -13.8007 \\ 12.9317 & 28.7359 & 39.4289 & 28.1247 \end{pmatrix},$$

$$G_2 = \begin{pmatrix} 17.8223 & 14.8889 & 19.9818 & 4.8605 \\ 22.1746 & 22.0385 & 29.7235 & 14.8262 \\ 20.8185 & 20.0357 & 19.5413 & 11.9351 \\ -29.7677 & -17.8096 & -28.6271 & 15.9366 \end{pmatrix}.$$

By Algorithm 5, we get

$$X_1^* = \begin{pmatrix} 8.6666 & -16.4843 & -9.8988 & -32.8552 \\ -9.5983 & 21.3601 & -1.8898 & 31.9227 \\ 11.9932 & 61.3474 & 81.4414 & -0.0952 \\ -19.1405 & -44.7243 & -74.8887 & 12.3186 \end{pmatrix},$$

$$X_2^* = \begin{pmatrix} -43.1103 & -9.9278 & -21.2932 & 34.5128 \\ -13.2501 & -38.8111 & -28.5908 & -50.8758 \\ -28.4303 & -50.6079 & -76.7389 & -14.6883 \\ 40.2481 & 40.0852 & 86.0114 & -28.9959 \end{pmatrix}.$$

Then we can derive periodic state observer gain

$$L_1 = \begin{pmatrix} 0.2221 & 0.2190 & 0.0746 & -0.7496 \\ 1.0525 & -0.8458 & -1.3244 & 1.4393 \\ -1.2140 & 1.4337 & 1.9888 & -2.7621 \\ -1.5071 & 1.8314 & 2.5259 & -3.4675 \end{pmatrix},$$

$$L_2 = \begin{pmatrix} 1.2435 & 3.1756 & 0.2750 & 1.4361 \\ -4.1305 & -9.2566 & -2.0688 & -2.2803 \\ 6.7149 & 15.3388 & 2.8153 & 4.2151 \\ 5.1582 & 11.7399 & 2.1194 & 3.0254 \end{pmatrix}.$$

Remark 2. Explanation of convergence factors of several algorithms in this paper is as follows. The convergence factor μ is used in Algorithms 1-5. In addition, In Algorithm 2, the relaxation factor ω_2 is used. The the relaxation factor ω_1 is used in Algorithm 4, and in Algorithm 5, the relaxation factor ω is used. It should be noted that the convergence factors in the several algorithms can be chosen to be the same or different under conditions that satisfy the theorems. The relaxation factors are selected to be any number between 0 and 1.

Remark 3. The idea of solving linear systems is to transform general linear systems into corresponding Sylvester matrix equations. The algorithm proposed in this paper can be used to solve a variety of Sylvester matrix equations, so it is suitable for solving systems that can be transformed into Sylvester matrix equations.

6 Concluding remark

In this paper, RRJGI algorithm, RRAJGI algorithm, CRJGI algorithm and CRAJGI algorithm are proposed for the discrete-time periodic Sylvester transpose matrix equations. Furthermore, the convergence theorems of the algorithms under arbitrary initial matrices are given by using the Frobenious norm. Two numerical examples show that the algorithms have faster convergence speed than GI algorithm, RGI algorithm and AGI algorithm which shows that the cost of computing time and storage space can be saved. At the end of this paper, the control application on the proposed Algorithm 2 is given. Therefore, our proposed algorithm is benefit to solve the observer design problem. In addition, our proposed algorithm can also solve the iterative solution of the other matrix equations, for example, the periodic coupled Sylvester conjugate matrix equations, the coupled discrete-time periodic Sylvester conjugate (transpose) matrix equations.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] S. Bittanti, P. Colaneri, Analysis of discrete-time linear periodic systems, Control Dyn. Syst. 78 (96) (1996) 313-339.

[2] F. Ding, H.M Zhang, Gradient-based iterative algorithm for a class of the coupled matrix equations related to control systems, IET Control Theory Appl. 8 (15) (2014) 1588-1595.

[3] L. Xu, G.L Song, A recursive parameter estimation algorithm for modeling signals with multi-frequencies, Circ. Syst. Signal Process. 39 (8) (2020) 4198-4224.

[4] B. Zhou, G.R. Duan, Z.L Lin, A parametric periodic lyapunov equation with application in semi-global stabilization of discrete-time periodic systems subject to actuator saturation, Automatica. 47 (2) (2011) 316-325.

[5] G.B. Cai, C.H. Hu, Solving periodic lyapunov matrix equations via finite steps iteration, IET Control Theory Appl. 6 (13) (2012) 2111-2119.

[6] L. Xie, J. Ding, F. Ding, Gradient based iterative solutions for general linear matrix equations, Computers & Mathematics with Applications. 58 (7) (2009) 1441-1448.

[7] L.L Lv, L Zhang, On the periodic Sylvester equations and their applications in periodic Luenberger observers design, Journal of the Franklin Institute. 353 (5) (2016) 1005-1018.

[8] B Zhou, G.R Duan, Periodic Lyapunov Equation Based Approaches to the Stabilization of Continuous-Time Periodic Linear Systems, IEEE Transactions on Automatic Control. 57 (8) (2012) 2139-2146.

[9] P. Benner, M.S. Hossain, T. Stykel, Low-rank iterative methods for periodic projected Lyapunov equations and their application in model reduction of periodic descriptor systems, Numerical Algorithms. 67 (3) (2012) 669-690.

[10] K.W. Chu, H.Y. Fan, W.W. Lin, Projected generalized discrete-time periodic Lyapunov equations and balanced realization of periodic descriptor systems, SIAM Journal on Matrix Analysis and Applications. 29 (3) (2007) 982-1006.

[11] A. Varga, On computing minimal realizations of periodic descriptor systems, IFAC Proceedings Volumes. 40 (14) (2007) 175-180.

[12] M. Hajarian, Matrix form of biconjugate residual algorithm to solve the discrete-time periodic sylvester matrix equations, Asian Journal of Control. 20 (1) (2018) 49-56.

[13] M. Hajarian, Extending LSQR methods to solve the generalized Sylvester-transpose and periodic Sylvester matrix equations, Mathematical Methods in the Applied Sciences. 37 (13) (2014) 2017-2028.

[14] W.L Wang, C.Q Song, Iterative algorithms for discrete-time periodic Sylvester matrix equations and its application in antilinear periodic system, Applied Numerical Mathematics. 168 (2021) 251-273.

[15] B.H. Huang, C.F. Ma, Finite iterative algorithm for the symmetric periodic least squares solutions of a class of periodic sylvester matrix equations, Numer. Algor. 81 (2019) 377-406.

[16] L.L. Lv, Z. Zhang, L. Zhang, X.X. Liu, Gradient based approach for generalized discrete-time periodic coupled Sylvester matrix equations, Journal of the Franklin Institute. 355 (15) (2018) 7691-7705.

[17] C.F Ma, T.X Yan, A finite iterative algorithm for the general discrete-time periodic Sylvester matrix equations, Journal of the Franklin Institute. 359 (9) (2022) 4410-4432.

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

[19] M. Hajarian, Convergence analysis of the MCGNR algorithm for the least squares solution group of discrete-time periodic coupled matrix equations, Transactions of the Institute of Measurement and Control. 39 (1) (2017) 29-42.

[20] Z.H. He, Q.W. Wang, A system of periodic discrete-time coupled sylvester quaternion matrix equations, Algebra Coll. 24 (1) (2017) 169-180.

[21] B. Zhou, Solutions to linear bimatrix equations with applications to pole assignment of complex-valued linear systems, J. Franklin Inst. 355 (15) (2018) 7246-7280.

[22] Q.W. Wang, Bisymmetric and centrosymmetric solutions to systems of real quaternion matrix equations, Comput. Math. Applic. 49 (5-6) (2005) 641-650.

[23] X.P. Sheng, A relaxed gradient based algorithm for solving generalized coupled sylvester matrix equations, J. Franklin Inst. 355 (10) (2018) 4282-4297.

[24] H.M. Zhang, H.C. Yin, New proof of the gradient-based iterative algorithm for a complex conjugate and transpose matrix equation, J. Frankl. Inst. 354 (16) (2017) 7585-7603.

[25] M. Hajarian, Matrix form of the CGS method for solving general coupled matrix equations, Applied Mathematics Letters. 34 (2014) 37-42.

[26] Y.J. Xie, C.F. Ma, The accelerated gradient based iterative algorithm for solving a class of generalized Sylvester-transpose matrix equation, Appl. Math. Comput. 273 (2016) 1257-1269.

[27] F. Ding, P.X. Liu, J. Ding, Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle, Applied Mathematics and Computation. 197 (1) (2008) 41-50.

[28] Z.Y. Li, Y. Wang, B. Zhou, G.R. Duan, Least squares solution with the minimum-norm to general matrix equations via iteration, Applied Mathematics and Computation. 215 (10) (2010) 3547-3562.

[29] A. Kittisopaporn, P. Chansangiam, Approximated least-squares solutions of a generalized Sylvester-transpose matrix equation via gradient-descent iterative algorithm, Advances in D-ifference Equations. 2021 (1) (2021) 1-18.

[30] Z.B. Chen, X.S Chen, Conjugate gradient-based iterative algorithm for solving generalized periodic coupled Sylvester matrix equations, Journal of the Franklin Institute. 359 (17) (2022) 9925-9951.

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

[32] K. Tansri, P. Chansangiam, Conjugate Gradient Algorithm for Least-Squares Solutions of a Generalized Sylvester-Transpose Matrix Equation, Symmetry. 14 (9) (2022) 1868.

[33] S.H. Li, C.F. Ma, Factor gradient iterative algorithm for solving a class of discrete periodic Sylvester matrix equations, Journal of the Franklin Institute. 359 (17) (2022) 9952-9970.

[34] B.H. Huang, C.F. Ma, Finite iterative algorithm for the symmetric periodic least squares solutions of a class of periodic Sylvester matrix equations, Numerical Algorithms. 81 (2019) 377-406.

[35] N. Boonruangkan, P. Chansangiam, Convergence analysis of a gradient iterative algorithm with optimal convergence factor for a generalized Sylvester-transpose matrix equation, AIMS Math. 6 (2021) 8477-8496.