Journal of Applied Analysis and Computation
Volume 14, Number 4, August 2024, 2002–2036
Website:http://www.jaac-online.com
DOI:10.11948/20230106

A NEW BCR METHOD FOR COUPLED OPERATOR EQUATIONS WITH SUBMATRIX CONSTRAINT

Wenling Wang[1], Caiqin Song[1,†] and Wenli Wang[2]

Abstract In the present work, a new biconjugate residual (BCR) algorithm is proposed in order to compute the constraint solution of the coupled operator equations, in which the constraint solution include symmetric solution, reflective solution, centrosymmetric solution and anti-centrosymmetric solution as special cases. When the studied coupled operator equations are consistent, it is proved that constraint solutions can be convergent to the exact solutions if giving any initial complex matrices or real matrices. In addition, when the studied coupled operator equations are not consistent, the least norm constraint solutions above can also be computed by selecting any initial matrices. Finally, some numerical examples are provided for illustrating the effectiveness and superiority of new proposed method.

Keywords Operator matrix equations, BCR algorithm, least-norm constraint solutions, submatrix constraint.

MSC(2010) 13F35, 15A06, 41A29, 47A05, 65F30.

1. Introduction

The following notations are used all throughout this essay. The set of $m\times n$ complex matrices is denoted by the symbol $C^{m\times n}$. The complex vector space has n dimensions, and its symbol is C^n. The ith entry of the n-dimensional column unit vector $e_i \in C^n$ is 1. A $m\times n$ matrix with all entries one will be represented as $1^{m\times n}$, and a $m\times n$ matrix with all elements zero will be represented by $0^{m\times n}$. The $n\times n$ unit matrix is represented by the symbols I_n and S_n, respectively. For each A and B, we use $A\otimes B$ to represent Kronecker product of two variables, which is $A\otimes B = (a_{ij}B)$. For $B = (b_1, b_2, \cdots, b_n) \in C^{m\times n}$, we have $vec(B) = \left({b_1}^T, {b_2}^T, \cdots, {b_n}^T\right)^T$, in which $vec(\cdot)$ means vec operator. The symbols B^T, B^H and $\|B\|_F$ stand for the transpose, conjugate transpose and Frobenius norm of matrix B.

In addition, $LC^{p\times q, m\times n}$ represents the set of linear operator from $C^{p\times q}$ onto $C^{m\times n}$. Particularly, when $p = m$ and $q = n$, $LC^{m\times n, p\times q}$ is written as $LC^{m\times n}$. For linear operator $\mathscr{A} \in LC^{p\times q, m\times n}$, we have $\langle \mathscr{A}(M), N\rangle = \langle M, \mathscr{A}^*(N)\rangle$ where $\mathscr{A}^*$

†The corresponding author.
[1]School of Mathematical Sciences, University of Jinan, Jinan 250022, China
[2]School of Mathematics and Statistics, Beijing Jiaotong University, Beijing 100044, China
Email: wangwenling_dp@163.com(W. Wang),
songcaiqin1983@163.com(C. Song), wenliwang_cba@163.com(W. Wang)

is the conjugate operator of A, for all $X \in C^{p\times q}$, $Y \in C^{m\times n}$. As an illustration, if $\mathscr{A} : X \to AXB$, then $\mathscr{A}^* : X \to A^H XB^H$.

The matrix $J \in R^{n\times n}$ is said permutation matrix, if $J = [e_n, e_{n-1}, \cdots, e_1]$, where e_i is unit vector and entry ith is 1. Thus, we have the following constraint solutions.

Definition 1.1. If $X = X^T$, where $X \in C^{n\times n}$ is denoted, then the matrix $X \in CC^{n\times n}$ is said to be symmetric.

Definition 1.2. If $X = JXJ$, then the matrix $X \in CJC^{n\times n}(J)$, where J is an n-order permutation matrix, is said to be centrosymmetric.

Definition 1.3. If $X = -JXJ$, then the matrix $X \in ACJC^{n\times n}(J)$, where J is an n-order permutation matrix, is said to be anti-centrosymmetric matrix.

Definition 1.4. If $X = PXP$, then the matrix $X \in CPC^{n\times n}(P)$, where P is an orthogonal matrix of order $n \times n$ that meets the conditions $P^H = P$ and $P^2 = I_n$, is said to be reflexive matrix.

Definition 1.5. The operator $\mathscr{U} \in LC^{p\times q}$ is said self-conjugate involution if $\mathscr{U}^2 = \mathcal{I}$ and $\mathscr{U}^* = \mathscr{U}$, each and every constraint solution is written as $X = \mathscr{U}(X)$.

Various linear matrix equations have been widely applied in science and engineering [5,19,21,25], neural networks [18,22,23,33,36], robot positioning and tracking [4, 20], intelligent structural system control [11, 26, 30], structural design [15], vibration theory [10], linear optimal control [1, 9, 39], etc. For example, Lyapunov matrix equation $A^T P + PA = -Q$ is related to solving the system stability [2]. Periodic descriptor systems' structural analysis uses the discrete-time periodic coupled Sylvester matrix equations [10]. Moreover, some quaternion equations have been investigated [16, 17, 37].

There are numerous iterative techniques available to solve the numerous matrix equations. To solve big sparse situations, Bouhamidi and Jbilou [3] presented an iterative projection method. By developing an iteration approach, Peng et al. [27] were able to solve the symmetric solution and its best approximation of the following equation

$$AXB = C. \tag{1.1}$$

For the preceding equation (1.1). Peng presented an iterative method for solving the minimal Frobenius norm solutions in [29]. Under any linear subspace constraint, as long as the appropriate linear projection operator is selected, the iterative method proposed by Hailin in [12] can be slightly modified to find the general numerical solution and its best approximation for equation

$$AXB + CYD = E. \tag{1.2}$$

A necessary and sufficient condition for the presence of reflexive (anti-reflexive) solutions of equations (1.2) was provided by Dehghan and Hajarian in [7]. Peng [28] proposed a successful method for solving the least square reflexive solution of matrix equation

$$A_1X_1B_1 + A_2X_2B_2 + \cdots + A_lX_lB_l = C. \tag{1.3}$$

By expanding on the concept of the conjugate gradient approach, Dehghan and Hajarian [8] created an effective numerical algorithm for the equation

$$\sum_{j=1}^{p} A_{ij} X_j B_{ij} = M_i, (i = 1, 2, \cdots, p) . \tag{1.4}$$

In the present work, we propose a brand-new biconjugate residual (BCR) method for obtaining the constraint solution of coupled operator equations

$$\left[\sum_{j=1}^{n} \mathscr{A}_{1j}(X_j), \sum_{j=1}^{n} \mathscr{A}_{2j}(X_j), \cdots, \sum_{j=1}^{n} \mathscr{A}_{mj}(X_j)\right] = [M_1, M_2, \cdots, M_m], \tag{1.5}$$

in which $\mathscr{A}_{ij} \in LC^{p_i \times q_i, m_j \times n_j}$ and $M_i \in C^{p_i \times q_i}$, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n$. Apparently, the equations (1.5) are also included the matrix equations (1.1), (1.2), (1.3) and (1.4). In this case, the constraint solutions covered in this work include those that are provided in Definitions 1.1, 1.2, 1.3, and 1.4. Furthermore, using this new BCR algorithm, we also show that it is possible to arrive at the iterative solution of matrix equations (1.5) in a limited number of steps. We also demonstrate that it is possible to discover the minimal Frobenius norm solutions if the rounding error is ignored. Moreover, we have corrected some errors existed in [24]. Lastly, numerical examples for Eq.(1.5) are provided in order to demonstrate the efficiency and superiority of new presented algorithm.

The rest of this article is arranged as follows. To solve the constraint solutions $[X_1^*, X_2^*, \cdots, X_n^*]$ of matrix equations, we suggest the BCR algorithm in Section 2. (1.5). In Section 3, we demonstrate that by choosing a unique initial matrix group, it is possible to arrive at the lowest norm solutions of the matrix equations (1.5). We provide some numerical examples in Section 4 to demonstrate the viability of the suggested approach. Lastly, Section 5 has the conclusion.

2. BCR algorithm for coupled operator matrix equations (1.5)

We first present a new biconjugate residual method in this part for solving linear matrix equations (1.5) based on BCR algorithm of matrix vector equation by introducing operators and inner products, which is called Algorithm 1 in this paper. And then we give the relevant properties of Algorithm 1.

Algorithm 1

Step 1. Input $\mathscr{A}_{ij} \in LC^{p_i \times q_i, m_j \times n_j}$, $M_i \in C^{p_i \times q_i}$, $\mathscr{U} \in LC^{p_i \times q_i}$, arbitrary initial group $X_j^{(1)} \in \mathscr{S}$, $S_j^{(1)} \in \mathscr{S}$ and $\varepsilon > 0$.

Compute

$$R_i^{(1)} = \sum_{j=1}^{n} \mathscr{A}_{ij}\left(X_j^{(1)}\right) - M_i, U_j^{(1)} = S_j^{(1)}, V_i^{(1)} = R_i^{(1)},$$

$$W_i^{(1)} = \sum_{j=1}^{n} \mathscr{A}_{ij}\left(U_j^{(1)}\right), \tilde{Z}_j^{(1)} = \sum_{i=1}^{m} \mathscr{A}_{ij}^*\left(V_i^{(1)}\right),$$

$$
\begin{aligned}
&Z_j^{(1)} = \frac{1}{2}\left(\tilde{Z}_j^{(1)} + \mathscr{U}\left(\tilde{Z}_j^{(1)}\right)\right),\\
&r_1 = \sqrt{\sum_{i=1}^{m}\left\|R_i^{(1)}\right\|_F^2},\\
&k := 1.
\end{aligned}
$$

Step 2. if $r_k < \varepsilon$ stop; go to Step 3 if not;
Step 3.

$$
\begin{aligned}
&\alpha_k = \frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(k)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(k)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle W_i^{(k)}, W_i^{(k)}\right\rangle},\\
&X_j^{(k+1)} = X_j^{(k)} - \alpha_k U_j^{(k)},\\
&R_i^{(k+1)} = R_i^{(k)} - \alpha_k W_i^{(k)},\\
&\beta_k = \frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(k)}, \sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(k)}\right)\right\rangle}{\sum\limits_{j=1}^{n}\left\langle Z_j^{(k)}, Z_j^{(k)}\right\rangle},\\
&S_j^{(k+1)} = S_j^{(k)} - \beta_k Z_j^{(k)},\\
&\gamma_k = \frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(k+1)}, \sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(k+1)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle R_i^{(k)}, \sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(k)}\right)\right\rangle},\\
&U_j^{(k+1)} = S_j^{(k+1)} + \gamma_k U_j^{(k)},\\
&V_i^{(k+1)} = R_i^{(k+1)} + \gamma_k V_i^{(k)},\\
&W_i^{(k+1)} = \sum_{j=1}^{n}\mathscr{A}_{ij}\left(U_j^{(k+1)}\right),\\
&\tilde{Z}_j^{(k+1)} = \sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(k+1)}\right),\\
&Z_j^{(k+1)} = \frac{1}{2}\left(\tilde{Z}_j^{(k+1)} + \mathscr{U}\left(\tilde{Z}_j^{(k+1)}\right)\right) + \gamma_k Z_j^{(k)},\\
&r_k = \sqrt{\sum_{i=1}^{m}\left\|R_i^{(k)}\right\|_F^2}.
\end{aligned}
$$

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

Remark 2.1. $\mathscr{S}$ represents a set of constraint matrices, which satisfies the general solutions such as symmetric solution, reflexive solution, centrosymmetric solution and anti-centrosymmetric solution in Definition 1.5.

From Algorithm 1, we have $W_i^{(k+1)} = \sum\limits_{j=1}^{n} \mathscr{A}_{ij}\left(U_j^{(k+1)}\right)$ and $U_j^{(k+1)} = S_j^{(k+1)} + \gamma_k U_j^{(k)}$. One can now obtain by putting the second equation in the first one

$$\begin{aligned} & W_i^{(k+1)} \\ =& \sum_{j=1}^{n} \mathscr{A}_{ij}\left(S_j^{(k+1)} + \gamma_k U_j^{(k)}\right) = \sum_{j=1}^{n} \mathscr{A}_{ij}\left(S_j^{(k+1)}\right) + \gamma_k \sum_{j=1}^{n} \mathscr{A}_{ij}\left(U_j^{(k)}\right). \end{aligned} \tag{2.1}$$

The following equality

$$\begin{aligned} & \frac{1}{2}\left(\widetilde{TZ}_j^{(k+1)} + \mathscr{U}\left(\widetilde{TZ}_j^{(k+1)}\right)\right) \\ =& \frac{1}{2}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}^*\left(R_i^{(k+1)}\right) + \mathscr{U}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}^*\left(R_i^{(k+1)}\right)\right)\right) + \gamma_k Z_i^{(k)}, \end{aligned} \tag{2.2}$$

can be demonstrated by induction if we assume $\widetilde{TZ}_j^{(k+1)} = \sum\limits_{i=1}^{m} \mathscr{A}_{ij}^*\left(V_i^{(k+1)}\right)$ and combine it with $V_i^{(k+1)} = R_i^{(k+1)} + \gamma_k V_i^{(k)}$ and $Z_j^{(1)} = \frac{1}{2}\left(\tilde{Z}_j^{(1)} + \mathscr{U}\left(\tilde{Z}_j^{(1)}\right)\right)$. Therefore, we get $Z_j^{(k+1)} = \frac{1}{2}\left(\widetilde{TZ}_j^{(k+1)} + \mathscr{U}\left(\widetilde{TZ}_j^{(k+1)}\right)\right)$.

Remark 2.2. The second formula in page 74 of [24], R_j should be R_i. The correct and detailed proof is stated as follows.

$$\begin{aligned} & \frac{1}{2}\left(\widetilde{TZ}_j^{(k+1)} + \mathrm{P}_i \widetilde{TZ}_j^{(k+1)} \mathrm{P}_i\right) \\ =& \frac{1}{2}\left[\sum_{i=1}^{m} A_{ij}^T V_i^{(k+1)} B_{ij}^T + P_i\left(\sum_{i=1}^{m} A_{ij}^T V_i^{(k+1)} B_{ij}^T\right) P_i\right] \\ =& \frac{1}{2}\left[\sum_{i=1}^{m} A_{ij}^T R_i^{(k+1)} B_{ij}^T + \gamma_k \sum_{i=1}^{m} A_{ij}^T V_i^{(k)} B_{ij}^T + P_i\left(\sum_{i=1}^{m} A_{ij}^T R_i^{(k+1)} B_{ij}^T\right) P_i \right. \\ & \left. + \gamma_k P_i\left(\sum_{i=1}^{m} A_{ij}^T V_i^{(k)} B_{ij}^T\right) P_i\right]\Bigg) \\ =& \frac{1}{2}\left[\sum_{i=1}^{m} A_{ij}^T R_i^{(k+1)} B_{ij}^T + P_i\left(\sum_{i=1}^{m} A_{ij}^T R_i^{(k+1)} B_{ij}^T\right) P_i \right. \\ & \left. + \gamma_k\left(\sum_{i=1}^{m} A_{ij}^T V_i^{(k)} B_{ij}^T + P_i\left(\sum_{i=1}^{m} A_{ij}^T V_i^{(k)} B_{ij}^T\right) P_i\right)\right] \\ =& \frac{1}{2}\left[\sum_{i=1}^{m} A_{ij}^T R_i^{(k+1)} B_{ij}^T + P_i\left(\sum_{i=1}^{m} A_{ij}^T R_i^{(k+1)} B_{ij}^T\right) P_i\right] \\ & + \frac{1}{2}\gamma_k\left(\widetilde{TZ}_j^{(k)} + P_i \widetilde{TZ}_j^{(k)} P_i\right). \end{aligned}$$

Lemma 2.1. *The Algorithm 1 generated sequences* $\left\{X_j^{(k)}\right\}$, $\left\{S_j^{(k)}\right\}$, $\left\{U_j^{(k)}\right\}$ *and* $\left\{Z_j^{(k)}\right\}$, $j = 1, 2, \cdots, n$, *are contained in the constraint set* $\mathscr{S}$.

Proof. By means of induction, we demonstrate the conclusion. By using $\mathscr{U}^2 = \mathcal{I}$ and Algorithm 1 for $k = 1$, $X_j^{(1)} \in \mathscr{S}$, $S_j^{(1)} \in \mathscr{S}$, we get

$$\mathscr{U}\left(U_j^{(1)}\right) = \mathscr{U}\left(S_j^{(1)}\right) = S_j^{(1)} = U_j^{(1)},$$

and

$$\begin{aligned}
\mathscr{U}\left(Z_j^{(1)}\right) =& \mathscr{U}\left(\frac{1}{2}\left(\tilde{Z}_j^{(1)} + \mathscr{U}\left(\tilde{Z}_j^{(1)}\right)\right)\right) \\
=& \frac{1}{2}\left(\mathscr{U}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(V_i^{(1)}\right)\right) + \sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(V_i^{(1)}\right)\right) \\
=& \frac{1}{2}\left(\tilde{Z}_j^{(1)} + \mathscr{U}\left(\tilde{Z}_j^{(1)}\right)\right) \\
=& Z_j^{(1)},
\end{aligned}$$

in which $U_j^{(1)}, Z_j^{(1)} \in \mathscr{S}$, $j = 1, 2, \cdots, n$.

For $k = 2$, we obtain

$$\begin{aligned}
\mathscr{U}\left(X_j^{(2)}\right) &= \mathscr{U}\left(X_j^{(1)} - \alpha_1 U_j^{(1)}\right) = \mathscr{U}\left(X_j^{(1)}\right) - \alpha_1 \mathscr{U}\left(U_j^{(1)}\right) = X_j^{(1)} - \alpha_1 U_j^{(1)} \\
&= X_j^{(2)}, \\
\mathscr{U}\left(S_j^{(2)}\right) &= \mathscr{U}\left(S_j^{(1)} - \beta_1 Z_j^{(1)}\right) = \mathscr{U}\left(S_j^{(1)}\right) - \beta_1 \mathscr{U}\left(Z_j^{(1)}\right) = S_j^{(1)} - \beta_1 Z_j^{(1)} = S_j^{(2)}, \\
\mathscr{U}\left(U_j^{(2)}\right) &= \mathscr{U}\left(S_j^{(2)} + \gamma_1 U_j^{(1)}\right) = \mathscr{U}\left(S_j^{(2)}\right) + \gamma_1 \mathscr{U}\left(U_j^{(1)}\right) = S_j^{(2)} + \gamma_1 U_j^{(1)} = U_j^{(2)},
\end{aligned}$$

and

$$\begin{aligned}
\mathscr{U}\left(Z_j^{(2)}\right) =& \mathscr{U}\left(\frac{1}{2}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(R_i^{(2)}\right) + \mathscr{U}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(R_i^{(2)}\right)\right)\right) + \gamma_1 Z_j^{(1)}\right) \\
=& \frac{1}{2}\left(\mathscr{U}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(R_i^{(2)}\right)\right) + \sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(R_i^{(2)}\right)\right) + \gamma_1 \mathscr{U}\left(Z_j^{(1)}\right) \\
=& \frac{1}{2}\left(\mathscr{U}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(R_i^{(2)}\right)\right) + \sum_{i=1}^{m} \mathscr{A}_{ij}{}^{*}\left(R_i^{(2)}\right)\right) + \gamma_1 Z_j^{(1)} \\
=& Z_j^{(2)},
\end{aligned}$$

that is to say $X_j^{(2)}, S_j^{(2)}, U_j^{(2)}, Z_j^{(2)} \in \mathscr{S}$, $j = 1, 2, \cdots, n$.

Now, we assume the conclusion is real for $k = u\,(u \geq 2)$, namely $X_j^{(u)}, S_j^{(u)}, U_j^{(u)}$, $Z_j^{(u)} \in \mathscr{S}$. It follows from Algorithm 1 that

$$\begin{aligned}
\mathscr{U}\left(X_j^{(u+1)}\right) &= \mathscr{U}\left(X_j^{(u)} - \alpha_u U_j^{(u)}\right) = \mathscr{U}\left(X_j^{(u)}\right) - \alpha_u \mathscr{U}\left(U_j^{(u)}\right) = X_j^{(u)} - \alpha_u U_j^{(u)} \\
&= X_j^{(u+1)},
\end{aligned}$$

$$\begin{aligned}\mathscr{U}\left(S_j^{(u+1)}\right) &= \mathscr{U}\left(S_j^{(u)} - \beta_u Z_j^{(u)}\right) = \mathscr{U}\left(S_j^{(u)}\right) - \beta_u \mathscr{U}\left(Z_j^{(u)}\right) = S_j^{(u)} - \beta_u Z_j^{(u)} \\ &= S_j^{(u+1)}, \\ \mathscr{U}\left(U_j^{(u+1)}\right) &= \mathscr{U}\left(S_j^{(u+1)} + \gamma_u U_j^{(u)}\right) = \mathscr{U}\left(S_j^{(u+1)}\right) + \gamma_u \mathscr{U}\left(U_j^{(u)}\right) \\ &= S_j^{(u+1)} + \gamma_u U_j^{(u)} = U_j^{(u+1)},\end{aligned}$$

and

$$\begin{aligned}\mathscr{U}\left(Z_j^{(u+1)}\right) =&\mathscr{U}\left(\frac{1}{2}\left(\sum_{i=1}^m \mathscr{A}_{ij}{}^*\left(R_i^{(u+1)}\right) + \mathscr{U}\left(\sum_{i=1}^m \mathscr{A}_{ij}{}^*\left(R_i^{(u+1)}\right)\right)\right) + \gamma_u Z_j^{(u)}\right) \\ =&\frac{1}{2}\left(\mathscr{U}\left(\sum_{i=1}^m \mathscr{A}_{ij}{}^*\left(R_i^{(u+1)}\right)\right) + \sum_{i=1}^m \mathscr{A}_{ij}{}^*\left(R_i^{(u+1)}\right)\right) + \gamma_u \mathscr{U}\left(Z_j^{(u)}\right) \\ =&\frac{1}{2}\left(\mathscr{U}\left(\sum_{i=1}^m \mathscr{A}_{ij}{}^*\left(R_i^{(u+1)}\right)\right) + \sum_{i=1}^m \mathscr{A}_{ij}{}^*\left(R_i^{(u+1)}\right)\right) + \gamma_u Z_j^{(u)} \\ =&Z_j^{(u+1)}.\end{aligned}$$

So, we can get $X_j^{(u+1)}, S_j^{(u+1)}, U_j^{(u+1)}, Z_j^{(u+1)} \in \mathscr{S}$, $j = 1, 2, \cdots, n$.

Therefore, by the principle of induction, the conclusion holds. □

Remark 2.3. From Algorithm 1 and the relevant definitions of constraint solutions, it can be seen that the matrix sequence $X_j^{(k)}, S_j^{(k)}, U_j^{(k)}, W_i^{(k)}, Z_j^{(k)}$ generated by Algorithm 1 belongs to the constraint solution set $\mathscr{S}$.

Lemma 2.2. *If the initial matrix group is selected as $X_j^{(1)} \in \mathscr{S}$, $S_j^{(1)} \in \mathscr{S}$, $j = 1, 2, \cdots, n$, let the matrix sequences produced by Algorithm 1 be $\left\{R_i^{(k)}\right\}$, $\left\{W_i^{(k)}\right\}$, $i = 1, 2, \cdots, m$, and $\left\{S_j^{(k)}\right\}$, $\left\{Z_j^{(k)}\right\}$, then we have*

$$\sum_{i=1}^m \left\langle W_i^{(u)}, R_i^{(v)}\right\rangle = 0, \ u < v, \tag{2.3}$$

$$\sum_{j=1}^n \left\langle Z_j^{(u)}, S_j^{(v)}\right\rangle = 0, \ u < v, \tag{2.4}$$

$$\sum_{i=1}^m \left\langle W_i^{(u)}, W_i^{(v)}\right\rangle = 0, \ u \neq v, \tag{2.5}$$

$$\sum_{j=1}^n \left\langle Z_j^{(u)}, Z_j^{(v)}\right\rangle = 0, \ u \neq v, \tag{2.6}$$

in which $u, v = 1, 2, \cdots$.

Proof. We adopt induction on k since we only need to demonstrate that (2.3)-(2.6) holds for all $0 \le u < v \le k$. The reason is that $\langle M, N\rangle = \langle N, M\rangle$ holds for M and N. To begin with, accordance with Algorithm 1, for $k = 2$ we derive

$$\sum_{i=1}^m \left\langle W_i^{(1)}, R_i^{(2)}\right\rangle$$

$$
\begin{aligned}
&=\sum_{i=1}^{m}\left\langle W_i^{(1)}, R_i^{(1)}-\alpha_1 W_i^{(1)}\right\rangle \\
&=\sum_{i=1}^{m}\left\langle W_i^{(1)}, R_i^{(1)}\right\rangle-\alpha_1\sum_{i=1}^{m}\left\langle W_i^{(1)}, W_i^{(1)}\right\rangle \\
&=\sum_{i=1}^{m}\left\langle W_i^{(1)}, R_i^{(1)}\right\rangle-\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(1)}, \sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle W_i^{(1)}, W_i^{(1)}\right\rangle}\sum_{i=1}^{m}\left\langle W_i^{(1)}, W_i^{(1)}\right\rangle \\
&=\sum_{i=1}^{m}\left\langle W_i^{(1)}, R_i^{(1)}\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(1)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(U_j^{(1)}\right)\right\rangle \\
&=0,
\end{aligned}
$$

and

$$
\begin{aligned}
\sum_{j=1}^{n}\left\langle Z_j^{(1)}, S_j^{(2)}\right\rangle &=\sum_{j=1}^{n}\left\langle Z_j^{(1)}, S_j^{(1)}-\beta_1 Z_j^{(1)}\right\rangle \\
&=\sum_{j=1}^{n}\left\langle Z_j^{(1)}, S_j^{(1)}\right\rangle-\beta_1\sum_{j=1}^{n}\left\langle Z_j^{(1)}, Z_j^{(1)}\right\rangle \\
&=\sum_{j=1}^{n}\left\langle Z_j^{(1)}, S_j^{(1)}\right\rangle-\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(1)}, \sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle}{\sum\limits_{j=1}^{n}\left\langle Z_j^{(1)}, Z_j^{(1)}\right\rangle}\sum_{j=1}^{n}\left\langle Z_j^{(1)}, Z_j^{(1)}\right\rangle \\
&=\sum_{j=1}^{n}\left\langle Z_j^{(1)}, S_j^{(1)}\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(1)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle \\
&=\sum_{j=1}^{n}\left\langle Z_j^{(1)}, S_j^{(1)}\right\rangle-\sum_{j=1}^{n}\left\langle S_j^{(1)}, \sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(1)}\right)\right\rangle \\
&=0.
\end{aligned}
$$

Also, by Algorithm 1, for $k=2$ we can get

$$
\begin{aligned}
\sum_{i=1}^{m}\left\langle W_i^{(1)}, W_i^{(2)}\right\rangle &=\frac{1}{\alpha_1}\left(\sum_{i=1}^{m}\left\langle R_i^{(1)}-R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)+\gamma_1 W_i^{(1)}\right\rangle\right) \\
&=\frac{1}{\alpha_1}\left(\sum_{i=1}^{m}\left\langle R_i^{(1)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right. \\
&\qquad\left.+\gamma_1\sum_{i=1}^{m}\left\langle R_i^{(1)}, W_i^{(1)}\right\rangle-\gamma_1\sum_{i=1}^{m}\left\langle R_i^{(2)}, W_i^{(1)}\right\rangle\right)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{\alpha_1}\left(\sum_{j=1}^{n}\left\langle S_j^{(2)}, \sum_{i=1}^{m}\mathscr{A}_{ij}^*\left(R_i^{(1)}\right)\right\rangle - \sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right.\\
&\quad\left.+\frac{\sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle}{\sum_{i=1}^{m}\left\langle R_i^{(1)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle}\sum_{i=1}^{m}\left\langle R_i^{(1)}, W_i^{(1)}\right\rangle\right)\\
&= \frac{1}{\alpha_1}\left(\sum_{j=1}^{n}\left\langle S_j^{(2)}, Z_j^{(1)}\right\rangle - \sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right.\\
&\quad\left.+\frac{\sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle}{\sum_{i=1}^{m}\left\langle R_i^{(1)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle}\sum_{i=1}^{m}\left\langle R_i^{(1)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle\right)\\
&= \frac{1}{\alpha_1}\left(\sum_{j=1}^{n}\left\langle S_j^{(2)}, Z_j^{(1)}\right\rangle - \sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right.\\
&\quad\left.+\sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right)\\
&= 0,
\end{aligned}
$$

and

$$
\begin{aligned}
&\sum_{j=1}^{n}\left\langle Z_j^{(1)}, Z_j^{(2)}\right\rangle\\
=&\frac{1}{\beta_1}\left(\sum_{j=1}^{n}\left\langle S_j^{(1)} - S_j^{(2)}, \frac{1}{2}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^*\left(R_i^{(2)}\right) + \mathscr{U}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^*\left(R_i^{(2)}\right)\right)\right) + \gamma_1 Z_j^{(1)}\right\rangle\right)\\
=&\frac{1}{\beta_1}\left(\sum_{j=1}^{n}\left\langle S_j^{(1)} - S_j^{(2)}, \sum_{i=1}^{m}\mathscr{A}_{ij}^*\left(R_i^{(2)}\right)\right\rangle + \sum_{j=1}^{n}\left\langle S_j^{(1)} - S_j^{(2)}, \gamma_1 Z_j^{(1)}\right\rangle\right)\\
=&\frac{1}{\beta_1}\left(\sum_{j=1}^{n}\left\langle S_j^{(1)}, \sum_{i=1}^{m}\mathscr{A}_{ij}^*\left(R_i^{(2)}\right)\right\rangle - \sum_{j=1}^{n}\left\langle S_j^{(2)}, \sum_{i=1}^{m}\mathscr{A}_{ij}^*\left(R_i^{(2)}\right)\right\rangle\right.\\
&\left.+\gamma_1\sum_{j=1}^{n}\left\langle S_j^{(1)}, Z_j^{(1)}\right\rangle\right)\\
=&\frac{1}{\beta_1}\left(\sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle - \sum_{i=1}^{m}\left\langle R_i^{(2)}, \sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right)
\end{aligned}
$$

$$+\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(2)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle R_i^{(1)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle}\sum_{j=1}^{n}\left\langle\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(1)}\right),S_j^{(1)}\right\rangle\Bigg)$$

$$=\frac{1}{\beta_1}\left(\sum_{i=1}^{m}\left\langle R_i^{(2)},W_i^{(1)}\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(2)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right)$$

$$+\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(2)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle R_i^{(1)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle}\sum_{i=1}^{m}\left\langle R_i^{(1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(1)}\right)\right\rangle\Bigg)$$

$$=\frac{1}{\beta_1}\left(0-\sum_{i=1}^{m}\left\langle R_i^{(2)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle+\sum_{i=1}^{m}\left\langle R_i^{(2)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(2)}\right)\right\rangle\right)$$

$$=0.$$

Hence, for $k=2$, the equalities (2.3)-(2.6) holds. Now assume that (2.3)-(2.6) holds for $0\le u<v$, $0<v\le k$.

For $k=u+1$, we can get

$$\begin{aligned}
\sum_{i=1}^{m}\left\langle W_i^{(u)},R_i^{(u+1)}\right\rangle &= \sum_{i=1}^{m}\left\langle W_i^{(u)},R_i^{(u)}-\alpha_u W_i^{(u)}\right\rangle\\
&= \sum_{i=1}^{m}\left\langle W_i^{(u)},R_i^{(u)}\right\rangle-\alpha_u\sum_{i=1}^{m}\left\langle W_i^{(u)},W_i^{(u)}\right\rangle\\
&= \sum_{i=1}^{m}\left\langle R_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)+\gamma_{u-1}W_i^{(u-1)}\right\rangle\\
&\quad-\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(u)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle W_i^{(u)},W_i^{(u)}\right\rangle}\sum_{i=1}^{m}\left\langle W_i^{(u)},W_i^{(u)}\right\rangle\\
&= \sum_{i=1}^{m}\left\langle R_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle\\
&= 0,
\end{aligned}$$

and

$$\sum_{j=1}^{n}\left\langle Z_j^{(u)},S_j^{(u+1)}\right\rangle$$

$$=\sum_{j=1}^{n}\left\langle Z_j^{(u)},S_j^{(u)}-\beta_u Z_j^{(u)}\right\rangle$$

$$
\begin{aligned}
&=\sum_{j=1}^{n}\left\langle Z_j^{(u)},S_j^{(u)}\right\rangle-\beta_u\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(u)}\right\rangle\\
&=\sum_{j=1}^{n}\left\langle \frac{1}{2}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)+\mathscr{U}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)\right)\right)+\gamma_{u-1}Z_j^{(u-1)},S_j^{(u)}\right\rangle\\
&\quad-\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(u)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle}{\sum\limits_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(u)}\right\rangle}\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(u)}\right\rangle\\
&=\sum_{j=1}^{n}\left\langle \sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)+\gamma_{u-1}Z_j^{(u-1)},S_j^{(u)}\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle\\
&=\sum_{j=1}^{n}\left\langle \sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right),S_j^{(u)}\right\rangle-\sum_{j=1}^{n}\left\langle \sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right),S_j^{(u)}\right\rangle\\
&=0.
\end{aligned}
$$

In addition, for $k=u+1$, we can obtain

$$
\begin{aligned}
&\sum_{i=1}^{m}\left\langle W_i^{(u)},W_i^{(u+1)}\right\rangle\\
=&\frac{1}{\alpha_u}\left(\sum_{i=1}^{m}\left\langle R_i^{(u)}-R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)+\gamma_u W_i^{(u)}\right\rangle\right)\\
=&\frac{1}{\alpha_u}\left(\sum_{i=1}^{m}\left\langle R_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle\right.\\
&\left.+\gamma_u\sum_{i=1}^{m}\left\langle R_i^{(u)},W_i^{(u)}\right\rangle\right)\\
=&\frac{1}{\alpha_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(u+1)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{i=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle\right.\\
&\left.+\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(u+1)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle R_i^{(u)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle}\sum_{i=1}^{m}\left\langle R_i^{(u)},W_i^{(u)}\right\rangle\right)\\
=&\frac{1}{\alpha_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(u+1)},\frac{1}{2}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)+\mathscr{U}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)\right)\right)+\gamma_{u-1}Z_j^{(u-1)}\right\rangle\right.\\
&-\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle
\end{aligned}
$$

$$
\left.+\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(u+1)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle R_i^{(u)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle}\sum_{i=1}^{m}\left\langle R_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)+\gamma_{u-1}W_i^{(u-1)}\right\rangle\right)
$$

$$
=\frac{1}{\alpha_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(u+1)},Z_j^{(u)}\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle\right.
$$

$$
\left.+\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(u+1)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle}{\sum\limits_{i=1}^{m}\left\langle R_i^{(u)},\sum\limits_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle}\sum_{i=1}^{m}\left\langle R_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle\right)
$$

$$
=0,
$$

and

$$
\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(u+1)}\right\rangle
$$

$$
=\frac{1}{\beta_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(u)}-S_j^{(u+1)},\frac{1}{2}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u+1)}\right)+\mathscr{U}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u+1)}\right)\right)\right)\right.\right.
$$

$$
\left.\left.+\gamma_u Z_j^{(u)}\right\rangle\right)
$$

$$
=\frac{1}{\beta_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(u)}-S_j^{(u+1)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u+1)}\right)\right\rangle+\gamma_u\sum_{j=1}^{n}\left\langle S_j^{(u)}-S_j^{(u+1)},Z_j^{(u)}\right\rangle\right)
$$

$$
=\frac{1}{\beta_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(u)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u+1)}\right)\right\rangle-\sum_{j=1}^{n}\left\langle S_j^{(u+1)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u+1)}\right)\right\rangle\right.
$$

$$
\left.+\gamma_u\sum_{j=1}^{n}\left\langle S_j^{(u)},Z_j^{(u)}\right\rangle\right)
$$

$$
=\frac{1}{\beta_u}\left(\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle\right.
$$

$$
\left.+\gamma_u\sum_{j=1}^{n}\left\langle S_j^{(u)},\frac{1}{2}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)+\mathscr{U}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left({R_i}^{(u)}\right)\right)\right)+\gamma_{u-1}Z_j^{(u-1)}\right\rangle\right)
$$

$$
=\frac{1}{\beta_u}\left(\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle\right.
$$

$$+\gamma_u \sum_{j=1}^{n} \left\langle S_j^{(u)}, \sum_{i=1}^{m} \mathscr{A}_{ij}^* \left(R_i{}^{(u)} \right) \right\rangle + \gamma_u \gamma_{u-1} \sum_{j=1}^{n} \left\langle S_j^{(u)}, Z_j^{(u-1)} \right\rangle \Bigg)$$

$$= \frac{1}{\beta_u} \left(\sum_{i=1}^{m} \left\langle R_i^{(u+1)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(U_j^{(u)} - \gamma_{u-1} U_j^{(u-1)} \right) \right\rangle \right.$$

$$- \sum_{i=1}^{m} \left\langle R_i^{(u+1)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(u+1)} \right) \right\rangle$$

$$\left. + \frac{\sum_{i=1}^{m} \left\langle R_i^{(u+1)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(u+1)} \right) \right\rangle}{\sum_{i=1}^{m} \left\langle R_i^{(u)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(u)} \right) \right\rangle} \sum_{j=1}^{n} \left\langle S_j^{(u)}, \sum_{i=1}^{m} \mathscr{A}_{ij}^* \left(R_j^{(u)} \right) \right\rangle \right)$$

$$= \frac{1}{\beta_u} \left(\sum_{i=1}^{m} \left\langle R_i^{(u+1)}, W_i{}^{(u)} - \gamma_{u-1} W_i^{(u-1)} \right\rangle - \sum_{i=1}^{m} \left\langle R_i^{(u+1)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(u+1)} \right) \right\rangle \right.$$

$$\left. + \frac{\sum_{i=1}^{m} \left\langle R_i^{(u+1)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(u+1)} \right) \right\rangle}{\sum_{i=1}^{m} \left\langle R_i^{(u)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(u)} \right) \right\rangle} \sum_{i=1}^{m} \left\langle R_i^{(u)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(u)} \right) \right\rangle \right)$$

$$= 0.$$

Thus, in the previous proof, we have proved the case $u = v$ and $v = u+1$. So we only need to prove that the equality statements (2.3)-(2.6) apply for all $0 \leq u < v+1$, $0 < v+1 \leq k$. Similarly, by Algorithm 1, we also get

$$\sum_{i=1}^{m} \left\langle W_i^{(u)}, R_i^{(v+1)} \right\rangle$$

$$= \sum_{i=1}^{m} \left\langle W_i^{(u)}, R_i{}^{(v)} - \alpha_v W_i^{(v)} \right\rangle$$

$$= \sum_{i=1}^{m} \left\langle W_i^{(u)}, R_i^{(v)} \right\rangle - \alpha_v \sum_{i=1}^{m} \left\langle W_i^{(u)}, W_i^{(v)} \right\rangle$$

$$= 0,$$

$$\sum_{j=1}^{n} \left\langle Z_j^{(u)}, S_j^{(v+1)} \right\rangle$$

$$= \sum_{j=1}^{n} \left\langle Z_j^{(u)}, S_j^{(v)} - \beta_v Z_j^{(v)} \right\rangle$$

$$
\begin{aligned}
&=\sum_{j=1}^{n}\left\langle Z_j^{(u)},S_j^{(v)}\right\rangle-\beta_v\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(v)}\right\rangle\\
&=0,\\
&\quad\sum_{i=1}^{m}\left\langle W_i^{(u)},W_i^{(v+1)}\right\rangle\\
&=\sum_{i=1}^{m}\left\langle W_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(v+1)}\right)+\gamma_v W_i^{(v)}\right\rangle\\
&=\sum_{i=1}^{m}\left\langle W_i^{(u)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(v+1)}\right)\right\rangle+\gamma_v\sum_{i=1}^{m}\left\langle W_i^{(u)},W_i^{(v)}\right\rangle\\
&=\frac{1}{\alpha_u}\sum_{i=1}^{m}\left\langle R_i^{(u)}-R_i^{(u+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(v+1)}\right)\right\rangle\\
&=\frac{1}{\alpha_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(v+1)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u)}\right)\right\rangle-\sum_{j=1}^{n}\left\langle S_j^{(v+1)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(u+1)}\right)\right\rangle\right)\\
&=\frac{1}{\alpha_u}\sum_{j=1}^{n}\left\langle S_j^{(v+1)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(V_i^{(u)}-\gamma_{u-1}V_i^{(u-1)}\right)\right\rangle\\
&\quad-\frac{1}{\alpha_u}\sum_{j=1}^{n}\left\langle S_j^{(v+1)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(V_i^{(u+1)}-\gamma_u V_i^{(u)}\right)\right\rangle\\
&=\frac{1}{\alpha_u}\left(\sum_{j=1}^{n}\left\langle S_j^{(v+1)},Z_j^{(u)}-\gamma_{u-1}Z_j^{(u-1)}\right\rangle-\sum_{j=1}^{n}\left\langle S_j^{(v+1)},Z_j^{(u+1)}-\gamma_u Z_j^{(u)}\right\rangle\right)\\
&=-\frac{1}{\alpha_u}\sum_{j=1}^{n}\left\langle S_j^{(v+1)},Z_j^{(u+1)}\right\rangle\\
&=0,
\end{aligned}
$$

and

$$
\begin{aligned}
&\quad\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(v+1)}\right\rangle\\
&=\sum_{j=1}^{n}\left\langle Z_j^{(u)},\frac{1}{2}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(v+1)}\right)+\mathscr{U}\left(\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(v+1)}\right)\right)\right)+\gamma_v Z_j^{(v)}\right\rangle\\
&=\sum_{j=1}^{n}\left\langle Z_j^{(u)},\sum_{i=1}^{m}\mathscr{A}_{ij}^{*}\left(R_i^{(v+1)}\right)\right\rangle+\gamma_v\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(v)}\right\rangle
\end{aligned}
$$

$$
\begin{aligned}
&=\sum_{i=1}^{m}\left\langle R_i^{(v+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(Z_j^{(u)}\right)\right\rangle\\
&=\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}-S_j^{(u+1)}\right)\right\rangle\\
&=\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u)}\right)\right\rangle-\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(S_j^{(u+1)}\right)\right\rangle\\
&=\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(U_j^{(u)}-\gamma_{u-1}U_j^{(u-1)}\right)\right\rangle\\
&\quad-\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},\sum_{j=1}^{n}\mathscr{A}_{ij}\left(U_j^{(u+1)}-\gamma_uU_j^{(u)}\right)\right\rangle\\
&=\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},W_i^{(u)}-\gamma_{u-1}W_i^{(u-1)}\right\rangle-\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},W_i^{(u+1)}-\gamma_uW_i^{(u)}\right\rangle\\
&=-\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(v+1)},W_i^{(u+1)}\right\rangle\\
&=0.
\end{aligned}
$$

Hence, we have demonstrated that the equalities (2.3)-(2.6) keep all $0 \leq u < v \leq k$, $k = 2, 3, \cdots$. In addition, as to $u > v$, by applying of the properties of inner product we have

$$
\sum_{i=1}^{m}\left\langle W_i^{(u)},W_i^{(v)}\right\rangle=\sum_{i=1}^{m}\left\langle W_i^{(v)},W_i^{(u)}\right\rangle,
$$

$$
\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j{}^{(v)}\right\rangle=\sum_{j=1}^{n}\left\langle Z_j^{(v)},Z_j^{(u)}\right\rangle.
$$

Therefore, the proof of this lemma has been finished. □

Remark 2.4. In Lemma 2.4 of the reference [24], there are some errors on the superscripts, subscripts and summation symbol. Now we correct them as follows.
(a).

$$
\begin{aligned}
&\sum_{j=1}^{\mathrm{n}}\left\langle Z_{\mathrm{j}}^{(1)},Z_j^{(2)}\right\rangle\\
=&\frac{1}{\beta_1}\left(\sum_{i=1}^{m}\left\langle R_i^{(1)},\sum_{j=1}^{n}A_{ij}S_j^{(1)}B_{ij}\right\rangle-\sum_{i=1}^{m}\left\langle R_i^{(2)},\sum_{j=1}^{n}A_{ij}S_j^{(2)}B_{ij}\right\rangle\right.
\end{aligned}
$$

$$+\frac{\sum\limits_{i=1}^{m}\left\langle R_i^{(2)},\sum\limits_{j=1}^{n}A_{ij}S_j^{(2)}B_{ij}\right\rangle}{\sum\limits_{i=1}^{m}\left\langle R_i^{(1)},\sum\limits_{j=1}^{n}A_{ij}S_j^{(1)}B_{ij}\right\rangle}\sum_{j=1}^{n}\left\langle\sum_{i=1}^{m}A_{ij}^{T}R_i{}^{(1)}B_{ij}^{T},S_j^{(1)}\right\rangle\Bigg).$$

(b).

$$\sum_{i=1}^{m}\left\langle Z_i^{(l)},Z_i^{(l+1)}\right\rangle \text{ should be } \sum_{j=1}^{n}\left\langle Z_j^{(l)},Z_j^{(l+1)}\right\rangle.$$

(c).

$$\begin{aligned}&\sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(l+1)}\right\rangle\\ =&\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(l+1)},\sum_{j=1}^{n}A_{ij}S_j^{(u)}B_{ij}\right\rangle-\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(l+1)},\sum_{j=1}^{n}A_{ij}S_j^{(u+1)}B_{ij}\right\rangle\\ =&\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(l+1)},W_i^{(u)}-\gamma_{u-1}W_i^{(u-1)}\right\rangle-\frac{1}{\beta_u}\sum_{i=1}^{m}\left\langle R_i^{(l+1)},W_i^{(u+1)}-\gamma_uW_i^{(u)}\right\rangle\\ =&0.\end{aligned}$$

(d).

$$\text{The last formula should be } \sum_{j=1}^{n}\left\langle Z_j^{(u)},Z_j^{(v)}\right\rangle=\sum_{j=1}^{n}\left\langle Z_j^{(v)},Z_j^{(u)}\right\rangle.$$

Remark 2.5. In the proof of Theorem 2.5 in [24], there are omissions in summation symbols and brackets. Let's correct them in the following.

$$\begin{aligned}&\sum_{i=1}^{m}\left\|R_i^{(k+1)}\right\|^2\\ =&\sum_{i=1}^{m}\left\langle R_i^{(k)}-\alpha_kW_i^{(k)},R_i^{(k)}-\alpha_kW_i^{(k)}\right\rangle\\ =&\sum_{i=1}^{m}\left\|R_i^{(k)}\right\|^2+\alpha_k^2\sum_{i=1}^{m}\left\|W_i^{(k)}\right\|^2-2\alpha_k\sum_{i=1}^{m}\left\langle R_i^{(k)},W_i^{(k)}\right\rangle\\ =&\sum_{i=1}^{m}\left\|R_i^{(k)}\right\|^2+\left(\frac{\sum_{i=1}^{m}\left\langle R_i^{(k)},\sum_{j=1}^{n}A_{ij}S_j^{(k)}B_{ij}\right\rangle}{\sum_{i=1}^{m}\left\langle W_i^{(k)},W_i^{(k)}\right\rangle}\right)^2\sum_{i=1}^{m}\left\|W_i^{(k)}\right\|^2\\ &-2\frac{\sum_{i=1}^{m}\left\langle R_i^{(k)},\sum_{j=1}^{n}A_{ij}S_j^{(k)}B_{ij}\right\rangle}{\sum_{i=1}^{m}\left\langle W_i^{(k)},W_i^{(k)}\right\rangle}\sum_{i=1}^{m}\left\langle R_i^{(k)},W_i^{(k)}\right\rangle\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|^2 + \left(\frac{\sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle}{\sum_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \right)^2 \sum_{i=1}^{m} \left\| W_i^{(k)} \right\|^2 \\
&\quad - 2 \frac{\sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle}{\sum_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \\
&= \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|^2 - \frac{\left(\sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \right)^2}{\sum_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \\
&\leq \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|^2 .
\end{aligned}
$$

Lemma 2.3. *If matrix groups $\left\{R_i^{(k)}\right\}$, $\left\{W_i^{(k)}\right\}$, $\left\{U_j^{(k)}\right\}$ and $\left\{S_j^{(k)}\right\}$ are the sequences produced by Algorithm 1, then $\sum\limits_{i=1}^{m} \left\| R_i^{(k)} \right\|_F^2$ is monotonically decreasing.*

Proof. Owing to matrix groups $\left\{R_i^{(k)}\right\}$, $\left\{W_i^{(k)}\right\}$, $\left\{U_j^{(k)}\right\}$, $\left\{S_j^{(k)}\right\}$ are produced with Algorithm 1, then, in accordance with Lemma 2.2, we get

$$
\begin{aligned}
&\sum_{i=1}^{m} \left\langle R_i^{(k)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(k)} \right) \right\rangle \\
&= \sum_{i=1}^{m} \left\langle R_i^{(k)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(U_j^{(k)} - \gamma_{k-1} U_j^{(k-1)} \right) \right\rangle \\
&= \sum_{i=1}^{m} \left\langle R_i^{(k)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(U_j^{(k)} \right) \right\rangle - \gamma_{k-1} \sum_{i=1}^{m} \left\langle R_i^{(k)}, \sum_{j=1}^{n} \mathscr{A}_{ij} \left(U_j^{(k-1)} \right) \right\rangle \\
&= \sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle - \gamma_{k-1} \sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k-1)} \right\rangle \\
&= \sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle .
\end{aligned} \tag{2.7}
$$

Hence, by Eq.(2.7) and Lemma 2.2, we get

$$
\frac{\left(\sum\limits_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \right)^2}{\sum\limits_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} > 0,
$$

and

$$
\sum_{i=1}^{m} \left\| R_i^{(k+1)} \right\|_F^2 = \sum_{i=1}^{m} \left\langle R_i^{(k)} - \alpha_k W_i^{(k)}, R_i^{(k)} - \alpha_k W_i^{(k)} \right\rangle
$$

$$
\begin{aligned}
&= \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|_F^2 + {\alpha_k}^2 \sum_{i=1}^{m} \left\| W_i^{(k)} \right\|_F^2 - 2\alpha_k \sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \\
&= \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|_F^2 + \left(\frac{\sum\limits_{i=1}^{m} \left\langle R_i^{(k)}, \sum\limits_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(k)} \right) \right\rangle}{\sum\limits_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \right)^2 \sum_{i=1}^{m} \left\| W_i^{(k)} \right\|_F^2 \\
&\quad - 2 \frac{\sum\limits_{i=1}^{m} \left\langle R_i^{(k)}, \sum\limits_{j=1}^{n} \mathscr{A}_{ij} \left(S_j^{(k)} \right) \right\rangle}{\sum\limits_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \\
&= \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|_F^2 + \left(\frac{\sum\limits_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle}{\sum\limits_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \right)^2 \sum_{i=1}^{m} \left\| W_i^{(k)} \right\|_F^2 \\
&\quad - 2 \frac{\sum\limits_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle}{\sum\limits_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \sum_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \\
&= \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|_F^2 - \frac{\left(\sum\limits_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \right)^2}{\sum\limits_{i=1}^{m} \left\langle W_i^{(k)}, W_i^{(k)} \right\rangle} \\
&\leq \sum_{i=1}^{m} \left\| R_i^{(k)} \right\|_F^2.
\end{aligned}
$$

So the conclusion on this lemma has been proved. □

Remark 2.6. Lemma 2.3 signify that $\sum\limits_{i=1}^{m} \left\| R_i^{(k+1)} \right\|_F^2$ is strictly monotonically decreasing if $\sum\limits_{i=1}^{m} \left\| R_i^{(k+1)} \right\|_F^2 \neq 0$ and $\sum\limits_{i=1}^{m} \left\langle R_i^{(k)}, W_i^{(k)} \right\rangle \neq 0$.

Theorem 2.1. *If there is a solution to Eq(1.5), then for any initial matrix $X_j \in \mathscr{S}\ (j = 1, 2, \cdots, n)$, the solution of Eq(1.5) can be acquired in a maximum of $u + 1$ iteration steps without rounding error in Algorithm 1, where $u = \sum\limits_{i=1}^{m} r_i u_i$.*

Proof. Let $u = \sum\limits_{i=1}^{m} r_i u_i$. Suppose $R_i^{(k)} \neq 0$ and $W_i^{(k)} \neq 0$ hold for $i = 1, 2, \cdots, u$. Now we let $W_i = diag\left(W_1^{(i)}, W_2^{(i)}, \cdots, W_m^{(i)} \right)$. By Lemma 2.2, we derive

$$
\langle W_i, W_j \rangle = \begin{cases} \|W_i\|^2, & i = j, \\ 0, & i \neq j, \end{cases} \tag{2.8}
$$

where W_1, W_2,$\cdots$, W_u are the orthogonal bases of the subspace of

$$E = \left\{ W \left| W = diag\left(W_1, W_2, \cdots, W_m\right), \ W_t \in C^{p_t \times q_t}, \text{ for } t = 1, 2, \cdots, m \right.\right\}.$$

Therefore, according to (2.3), we can get $R_{u+1} = 0$.

This implies that $\left(X_1^{(u+1)}, X_2^{(u+1)}, \cdots, X_n^{(u+1)}\right)$ are the solutions of equations. It is verified that the solutions of equations (1.5) can be obtained in a maximum of $u+1$ iteration steps. $\square$

Remark 2.7. Theorem 2.1 in [24] is missing the second power on the right side of the equation. The correct formula should be (2.8).

3. The least norm solution

In this section, we investigate the least norm solutions of matrix equations (1.5), in which $\mathscr{R}(A)$ stand for the column spaces of matrix A. First of all, we give some lemmas.

Lemma 3.1 (Lemma 2.5, [34])**.** *If the system of linear equations $Ax = b$ is consistent and has a solution $x^* \in \mathscr{R}\left(A^H\right)$, then x^* is the system's only least norm solutions.*

Lemma 3.2 (Lemma 6, [31])**.** *If and only if the matrix equations*

$$\begin{cases} \sum\limits_{j=1}^{n} \mathscr{A}_{ij}\left(X_j\right) = M_i, \ i = 1, 2, \cdots, m, \\ \sum\limits_{j=1}^{n} \mathscr{A}_{ij}^*\left(\mathscr{U}\left(X_j\right)\right) = M_i^H, \ i = 1, 2, \cdots, m, \end{cases} \tag{3.1}$$

are consistent, equation (1.5) is solvable.

Remark 3.1. Let $\mathscr{A}^*$ be the conjugate operator of $\mathscr{A}$, then $vec\left(\mathscr{A}\left(X\right)\right) = Mvec\left(X\right), vec\left(\mathscr{A}^*\left(X\right)\right) = M^H vec\left(X\right)$ for all $X \in C^{p \times q}$.

Theorem 3.1. *If Eq.(1.5) have solutions $X_j^{(1)} \in \mathscr{S}$, $j = 1, 2, \cdots, n$, and the initial matrix group are chosen as*

$$X_j^{(1)} = \sum_{i=1}^{m} \mathscr{A}_{ij}^*\left(Q_i\right) + \mathscr{U}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}^*\left(Q_i\right)\right), \tag{3.2}$$

$$S_j^{(1)} = \sum_{i=1}^{m} \mathscr{A}_{ij}^*\left(G_i\right) + \mathscr{U}\left(\sum_{i=1}^{m} \mathscr{A}_{ij}^*\left(G_i\right)\right), \tag{3.3}$$

where $Q_i, G_i \in C^{p_i \times q_i}$, $i = 1, 2, \cdots, m$, are arbitrary matrices, or more especially $X_j^{(1)} = 0$, $j = 1, 2, \cdots, n$, then the solutions $[{X_1}^*, {X_2}^*, \cdots, {X_n}^*]$ generalized by Algorithm 1 are the unique least Frobrnius norm solutions of system (1.5).

Proof. By Lemma 3.2, equations (1.5) have solutions if and only if equations (3.1) have solutions. Now we let E_t and T satisfy

$$vec\left(\sum_{j=1}^{n}\mathscr{A}_{tj}\left(X_j\right)\right)=E_t\begin{pmatrix}vec\left(X_1\right)\\ \vdots\\ vec\left(X_n\right)\end{pmatrix},\text{ and }\begin{pmatrix}vec\left(\mathscr{U}\left(X_1\right)\right)\\ \vdots\\ vec\left(\mathscr{U}\left(X_n\right)\right)\end{pmatrix}=T\begin{pmatrix}vec\left(X_1\right)\\ \vdots\\ vec\left(X_n\right)\end{pmatrix}.$$

Therefore coupled operator matrix equations (1.5) are equivalent to

$$\mathcal{T}\mathcal{Z}=f,$$

where

$$\mathcal{T}=\begin{pmatrix}E_1\\ E_2\\ \vdots\\ E_m\\ E_1T\\ E_2T\\ \vdots\\ E_mT\end{pmatrix},\ \mathcal{Z}=\left(vec(X_1)^H,\ vec(X_2)^H,\cdots,\ vec(X_n)^H\right)^H,$$

and $f=\left(vec(M_1)^H,vec(M_2)^H,\cdots,vec(M_m)^H,vec(M_1)^H,vec(M_2)^H,\cdots,vec(M_m)^H\right)^H$.

Assume that Q_i and G_i are the matrices with appropriate dimensions. Therefore, due to (3.2), (3.3) and Remark 3.1, we obtain

$$\begin{pmatrix}vec\left(X_1^{(1)}\right)\\ vec\left(X_2^{(1)}\right)\\ \vdots\\ \vdots\\ vec\left(X_n^{(1)}\right)\end{pmatrix}=\Gamma\begin{pmatrix}vec\left(Q_1\right)\\ vec\left(Q_2\right)\\ \vdots\\ vec\left(Q_m\right)\\ vec\left(Q_1\right)\\ vec\left(Q_2\right)\\ \vdots\\ vec\left(Q_m\right)\end{pmatrix},\tag{3.4}$$

in which

$$\Gamma=\left(E_1^H,E_2^H,\cdots,E_m^H,(E_1T)^H,(E_2T)^H,\cdots,(E_mT)^H\right).$$

Then we can obtain

$$\begin{pmatrix} vec\left(X_1^{(1)}\right) \\ vec\left(X_2^{(1)}\right) \\ \vdots \\ \vdots \\ vec\left(X_n^{(1)}\right) \end{pmatrix} = \Gamma \begin{pmatrix} vec\left(Q_1\right) \\ vec\left(Q_2\right) \\ \vdots \\ vec\left(Q_m\right) \\ vec\left(Q_1\right) \\ vec\left(Q_2\right) \\ \vdots \\ vec\left(Q_m\right) \end{pmatrix} \in \mathscr{R}\left(\Gamma\right).$$

Therefore, according to Algorithm 1 and $\|R_i\|_F^2 \neq 0, i = 1, 2, \cdots, m$, in a limited number of iterative steps, the solution $\left\{X_j^{(k)}\right\}$, $j = 1, 2, \cdots, n$ to equations (1.5) can be achieved. So we get

$$\begin{pmatrix} vec\left(X_1^{(k)}\right) \\ vec\left(X_2^{(k)}\right) \\ \vdots \\ \vdots \\ vec\left(X_n^{(k)}\right) \end{pmatrix} \in \mathscr{R}\left(\Gamma\right).$$

Thus, we derive

$$\begin{pmatrix} vec\left(X_1^*\right) \\ vec\left(X_2^*\right) \\ \vdots \\ \vdots \\ vec\left(X_n^*\right) \end{pmatrix} = \sum_{t=1}^{m} \left[E_t^H \begin{pmatrix} vec\left(Q_1\right) \\ \vdots \\ vec\left(Q_m\right) \end{pmatrix} + T^H E_t^H \begin{pmatrix} vec\left(Q_1\right) \\ \vdots \\ vec\left(Q_m\right) \end{pmatrix} \right] \in \mathscr{R}\left(\mathcal{T}^H\right).$$

So, from Lemma 3.1 and formulas (3.4), if the initial groups are chosen as $\left[X_1^{(1)}, X_2^{(1)}, \cdots, X_n^{(1)}\right]$ generated by formula (3.2) (especially $X_j^{(1)} = 0$, $j = 1, 2, \cdots, n$). From Remark 3.1 and Algorithm 1, the least norm constraint solution group $[X_1^*, X_2^*, \cdots, X_n^*]$ of (3.4) can be obtained. Therefore, it is also the only least Frobenius norm constrained solution of equations (1.5). □

Remark 3.2. The matrix Π of Theorem 3.1 in [24] is error. The right Π is

$$\begin{aligned}&\Pi\\ =&\mathcal{T}^T\\ =&\begin{pmatrix} B_{11}\otimes A_{11}^T & \cdots & B_{m1}\otimes A_{m1}^T & (P_1B_{11})\otimes\left(P_1A_{11}^T\right) & \cdots & (P_1B_{m1})\otimes\left(P_1A_{m1}^T\right)\\ B_{12}\otimes A_{12}^T & \cdots & B_{m2}\otimes A_{m2}^T & (P_2B_{12})\otimes\left(P_2A_{12}^T\right) & \cdots & (P_2B_{m2})\otimes\left(P_2A_{m2}^T\right)\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ B_{1n}\otimes A_{1n}^T & \cdots & B_{mn}\otimes A_{mn}^T & (P_nB_{1n})\otimes\left(P_nA_{1n}^T\right) & \cdots & (P_nB_{mn})\otimes\left(P_nA_{mn}^T\right)\end{pmatrix}.\end{aligned}$$

4. Numerical experiments

In this part, four numerical examples are provided for comparing Algorithm 1 with modified conjugate gradient algorithm (MCG) [27,34] and another modified biconjugate residual method (BCR2O) [14] under different constraint solutions [6,13,35] . Let $\varepsilon = 10^{-15}$ here. The iterative procedure in this paper is completed by MATLAB R2020b.

In the following numerical examples, the initial iteration matrices are selected according to the conditions of the constrained solution. Moreover, the residual and the relative error are defined as

$$r_k=\sqrt{\|R(k)\|_F^2},\ e_k=\frac{\|x(k)-x^*\|_F}{\|x^*\|_F}, \tag{4.1}$$

where k is the number of iterative step and $x(k)$ is the kth solution obtained.

Example 4.1. In this example, we compute the generalized symmetric solution of the following coupled transpose equations

$$\begin{cases} A_1XB_1+C_1Y^TD_1+E_1ZF_1=G_1,\\ A_2XB_2+C_2YD_2=G_2,\end{cases}$$

where

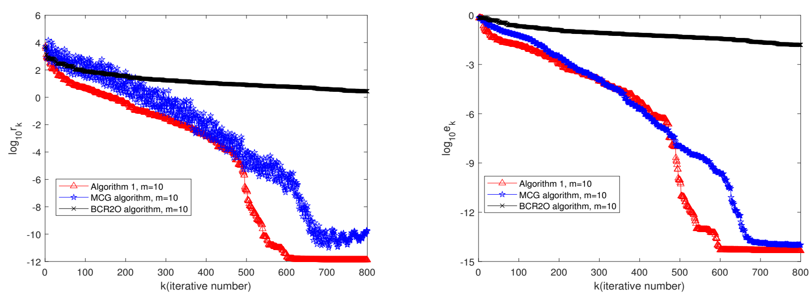
$$\begin{aligned}
A_1&=-tril\left(rand\left(m,m\right),1\right)+diag\left(30+diag\left(rand\left(m\right)\right)\right),\\
B_1&=-tril\left(rand\left(m,m\right),1\right)+diag\left(19+diag\left(rand\left(m\right)\right)\right),\\
C_1&=tril\left(rand\left(m,m\right),1\right)-diag\left(100+diag\left(rand\left(m\right)\right)\right),\\
D_1&=diag\left(2+diag\left(rand\left(m\right)\right)\right),\\
E_1&=-tril\left(rand\left(m,m\right),1\right)-diag\left(51+diag\left(rand\left(m\right)\right)\right),\\
F_1&=-tril\left(rand\left(m,m\right),1\right)+diag\left(13+diag\left(rand\left(m\right)\right)\right),\\
A_2&=-tril\left(rand\left(m,m\right),1\right)+diag\left(59+diag\left(rand\left(m\right)\right)\right),\\
B_2&=-tril\left(rand\left(m,m\right),1\right)+diag\left(64+diag\left(rand\left(m\right)\right)\right),\\
C_2&=-tril\left(rand\left(m,m\right),1\right)-diag\left(9+diag\left(rand\left(m\right)\right)\right),\\
D_2&=-diag\left(30+diag\left(rand\left(m\right)\right)\right).
\end{aligned}$$

Now we choose the initial matrices $X^{(1)}, Y^{(1)}, Z^{(1)} \in CC^{m\times m}, S_j^{(1)} \in CC^{m\times m}$, $j = 1, 2, 3$ as

$$\left(X^{(1)}, Y^{(1)}, Z^{(1)}\right) = \left(S_1^{(1)}, S_2^{(1)}, S_3^{(1)}\right) = \left(1^{m\times m}, 1^{m\times m}, 1^{m\times m}\right).$$

And let $(X^*, Y^*, Z^*) = \left(1^{m\times m}, 2\times 1^{m\times m}, 0^{m\times m}\right)$.

Fig.1 clearly shows the convergence performance comparison results of the residual and relative error of Algorithm 1, MCG algorithm and BCR2O algorithm. From Fig.1 (a) and Fig.1 (b), it is showed that the residual gradually decreases and tends to be stable with the increase of iteration steps, which means that Algorithm 1 is convergent and effective. Furthermore, it can be seen that Algorithm 1 converges with fewer iterative steps than MCG algorithm and BCR2O algorithm.



(a) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for residual (b) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for relative error

Fig. 1. Comparison of convergence curves for Example 4.1

In Table 1 and Table 2, we give the relationship between the iterative step and computational time of Algorithm 1, MCG algorithm and BCR2O algorithm on similar residual and relative error respectively. It can be more clearly seen that Algorithm 1 requires the least number of iterative steps in terms of similar residual and relative error, while BCR2O algorithm requires the most number of iterative steps.

Table 1. Iterative step, residual and computational time of Fig.1

Method						
Algorithm 1	$\log_{10} r_k$	-11.0132	-10.0916	-9.0389	-8.0021	-7.0491
	Steps	602	544	529	524	510
	Time (s)	0.1172	0.1048	0.1027	0.1005	0.1003
MCG algorithm	$\log_{10} r_k$	-11.0056	-10.3410	-9.0265	-8.7718	-7.0393
	Steps	796	660	639	638	620
	Time (s)	0.0680	0.0604	0.0596	0.0596	0.0595
BCR2O algorithm	$\log_{10} r_k$	-11.0029	-10.0005	-9.0168	-8.0029	-7.0000
	Steps	4899	4440	4065	3801	3363
	Time (s)	0.6295	0.5682	0.5141	0.4748	0.4269

Table 2. Iterative step, relative error and computational time of Fig.1

Method						
Algorithm 1	$\log_{10} e_k$	−14.0256	−13.1943	−12.0075	−11.0205	−10.0141
	Steps	617	597	527	522	501
	Time (s)	0.1179	0.1178	0.1062	0.1028	0.1005
MCG algorithm	$\log_{10} e_k$	−14.0268	−13.3053	−12.3002	−11.0664	−10.0049
	Steps	678	656	637	628	614
	Time (s)	0.0670	0.0585	0.0575	0.0565	0.0563
BCR2O algorithm	$\log_{10} e_k$	−14.0024	−13.0000	−12.0078	−11.0028	−10.0471
	Steps	5371	4749	4303	3927	3673
	Time (s)	0.6643	0.5433	0.5318	0.4818	0.4377

Example 4.2. In this example, we consider the reflexive solution of equation

$$AXB + CX^H D = E,$$

where, matrices are

$$A = \begin{pmatrix} 1.8147 + 0.7577i & 0.0000 + 0.0000i & 0.0000 + 0.0000i & 0.0000 + 0.0000i & 0.0000 + 0.0000i \\ 0.9058 + 0.0000i & 1.2785 + 0.0318i & 0.0000 + 0.0000i & 0.0000 + 0.0000i & 0.0000 + 0.0000i \\ 0.1270 + 0.0000i & 0.5469 + 0.0000i & 1.9572 + 0.3171i & 0.0000 + 0.0000i & 0.0000 + 0.0000i \\ 0.9134 + 0.0000i & 0.9575 + 0.0000i & 0.4854 + 0.0000i & 1.7922 + 0.7952i & 0.0000 + 0.0000i \\ 0.6324 + 0.0000i & 0.9649 + 0.0000i & 0.8003 + 0.0000i & 0.9595 + 0.0000i & 1.6787 + 0.7547i \end{pmatrix},$$

$$B = \begin{pmatrix} 0.2760 + 0.0000i & 0.4984 + 0.3517i & 0.7513 + 0.2858i & 0.9593 + 0.0759i & 0.8407 + 0.1299i \\ 0.6797 + 0.0000i & 0.9597 + 0.0000i & 0.2551 + 0.7572i & 0.5472 + 0.0540i & 0.2543 + 0.5688i \\ 0.6551 + 0.0000i & 0.3404 + 0.0000i & 0.5060 + 0.0000i & 0.1386 + 0.5308i & 0.8143 + 0.4694i \\ 0.1626 + 0.0000i & 0.5853 + 0.0000i & 0.6991 + 0.0000i & 0.1493 + 0.0000i & 0.2435 + 0.0119i \\ 0.1190 + 0.0000i & 0.2238 + 0.0000i & 0.8909 + 0.0000i & 0.2575 + 0.0000i & 0.9293 + 0.0000i \end{pmatrix},$$

$$C = \begin{pmatrix} 0.9631 - 2.9037i & 0.6241 + 0.0000i & 0.0377 + 0.0000i & 0.2619 + 0.0000i & 0.1068 + 0.0000i \\ 0.5468 + 0.0000i & 0.6791 - 2.7441i & 0.8852 + 0.0000i & 0.3354 + 0.0000i & 0.6538 + 0.0000i \\ 0.5211 + 0.0000i & 0.3955 + 0.0000i & 0.9133 - 2.8594i & 0.6797 + 0.0000i & 0.4942 + 0.0000i \\ 0.2316 + 0.0000i & 0.3674 + 0.0000i & 0.7962 + 0.0000i & 0.1366 - 2.0287i & 0.7791 + 0.0000i \\ 0.4889 + 0.0000i & 0.9880 + 0.0000i & 0.0987 + 0.0000i & 0.7212 + 0.0000i & 0.7150 - 2.4711i \end{pmatrix},$$

$$D = \begin{pmatrix} 0.0596 - 0.3993i & 0.0967 + 0.0000i & 0.6596 + 0.0000i & 0.4538 + 0.0000i & 0.1734 + 0.0000i \\ 0.6820 - 0.5269i & 0.8181 - 0.4317i & 0.5186 + 0.0000i & 0.4324 + 0.0000i & 0.3909 + 0.0000i \\ 0.0424 - 0.4168i & 0.8175 - 0.0155i & 0.9730 - 0.1981i & 0.8253 + 0.0000i & 0.8314 + 0.0000i \\ 0.0714 - 0.6569i & 0.7224 - 0.9841i & 0.6491 - 0.4897i & 0.0835 - 0.7379i & 0.8034 + 0.0000i \\ 0.5216 - 0.6280i & 0.1499 - 0.1672i & 0.8003 - 0.3395i & 0.1332 - 0.2691i & 0.0605 - 0.9831i \end{pmatrix}.$$

Here, we choose the initial matrices $X^{(1)} \in CPC^{5\times 5}(P)$, $S^{(1)} \in CPC^{5\times 5}(P)$

as

$$X^{(1)} = S^{(1)} = \begin{pmatrix} -0.2900 + 0.6713i & 0 & 0 & 0.8308 - 1.5996i & -0.2854 + 0.4219i \\ -0.0966 + 0.7364i & 0 & 0 & 0.0556 - 2.0454i & -0.0749 + 5.2002i \\ -0.5804 + 1.8934i & 0 & 0 & 0.0719 - 1.0964i & -0.6272 - 1.7749i \\ -0.4833 - 2.8318i & 0 & 0 & -0.5475 - 0.9384i & -1.4595 - 5.4380i \\ -0.5804 + 1.8934i & 0 & 0 & -0.0402 + 1.4562i & 1.0761 - 0.9343i \end{pmatrix},$$

and the orthogonal matrix in Definition 1.4 is selected as

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

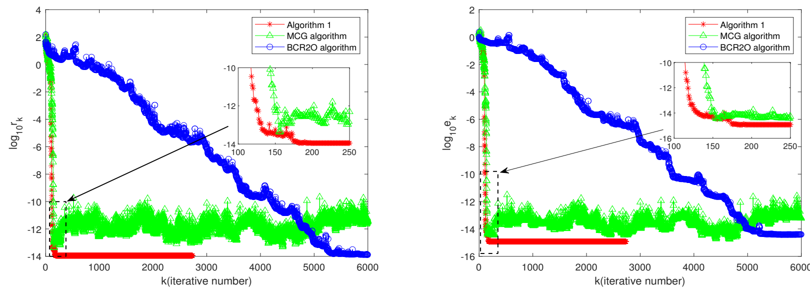
Let

$$X^{*} = \begin{pmatrix} 3.3918 + 3.7508i & 0 & 0 & 1.0618 + 3.3522i & 0.3074 + 0.0000i \\ 0.0000 + 0.0000i & 0 & 0 & 0.0000 + 0.0000i & 0.0000 + 0.0000i \\ 0.0000 + 0.0000i & 0 & 0 & 0.0000 + 0.0000i & 0.0000 + 0.0000i \\ 1.0626 + 2.4480i & 0 & 0 & 0.1816 + 2.2334i & 1.0542 + 0.0000i \\ 0.1376 + 0.0000i & 0 & 0 & 0.5330 + 0.0000i & 0.9148 + 2.8486i \end{pmatrix},$$

through iteration of Algorithm 1, we get

$$X^{(214)} = \begin{pmatrix} 3.3918 + 3.7508i & 0 & 0 & 1.0618 + 3.3522i & 0.3074 + 0.0000i \\ 0.0000 + 0.0000i & 0 & 0 & 0.0000 + 0.0000i & 0.0000 + 0.0000i \\ 0.0000 + 0.0000i & 0 & 0 & 0.0000 + 0.0000i & 0.0000 + 0.0000i \\ 1.0626 + 2.4480i & 0 & 0 & 0.1816 + 2.2334i & 1.0542 + 0.0000i \\ 0.1376 + 0.0000i & 0 & 0 & 0.5330 + 0.0000i & 0.9148 + 2.8486i \end{pmatrix} \in CPC^{5\times 5}(P).$$

Fig.2 illustrates the convergence performance comparison results of the residual and relative error of Algorithm 1, MCG algorithm and BCR2O algorithm. Fig.2 (a) and Fig.2 (b) show that Algorithm 1 is convergent and effective with the increase of iterative steps. Moreover, we can see that Algorithm 1 is faster than MCG algorithm and BCR2O algorithm.



(a) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for residual (b) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for relative error

Fig. 2. Comparison of convergence curves for Example 4.2

In Table 3 and Table 4, we give the relationship between the iterative step and computational time of Algorithm 1, MCG algorithm and BCR2O algorithm on the similar residual and relative error respectively. It can be more clearly seen that Algorithm 1 requires the least number of iterative steps and has the best convergence effect in terms of similar residual and relative error.

Table 3. Iterative step, residual and computational time of Fig.2

Method						
Algorithm 1	$\log_{10} r_k$	−13.2319	−12.2240	−11.0776	−10.4607	−9.1615
	Steps	132	125	120	118	115
	Time (s)	0.0288	0.0284	0.0282	0.0281	0.0281
MCG algorithm	$\log_{10} r_k$	−13.3343	−12.0808	−11.0865	−10.0900	−9.1201
	Steps	157	149	147	143	141
	Time (s)	0.0148	0.0148	0.0147	0.0145	0.0143
BCR2O algorithm	$\log_{10} r_k$	−13.0009	−12.0672	−11.0390	−10.0388	−9.0031
	Steps	5016	4868	4644	4216	3628
	Time (s)	0.3839	0.3706	0.3534	0.3271	0.2815

Table 4. Iterative step, relative error and computational time of Fig.2

Method						
Algorithm 1	$\log_{10} e_k$	−14.0878	−13.3449	−12.0370	−11.3456	−10.7950
	Steps	132	121	118	116	115
	Time (s)	0.0296	0.0291	0.0285	0.0285	0.0281
MCG algorithm	$\log_{10} e_k$	−14.3124	−13.5853	−12.2099	−11.0374	−10.4737
	Steps	153	148	144	141	139
	Time (s)	0.0155	0.0148	0.0146	0.0145	0.0145
BCR2O algorithm	$\log_{10} e_k$	−14.0036	−13.0107	−12.0219	−11.1523	−10.1196
	Steps	5014	4820	4634	4184	3606
	Time (s)	0.3680	0.3516	0.3399	0.3129	0.2884

Example 4.3. We consider the anti-centrosymmetric solutions of the following generalized coupled matrix equations

$$\begin{cases} A_1\overline{X_1}B_1 + C_1X_2D_1 = E_1, \\ \\ A_2X_1B_2 + C_2\overline{X_2}D_2 = E_2, \end{cases}$$

in which the coefficients matrices are

$$A_1 = \begin{pmatrix} 8i & 1-9i & 9 \\ 9i & 6+5i & 5-2i \\ 1 & 0 & 2i \end{pmatrix}, \ B_1 = \begin{pmatrix} 1 & i & 7+i \\ 0 & 3i & 4-5i \\ 2+3i & 9 & 5+5i \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 4-i & 3+7i & 8+3i \\ 5 & 2+2i & 6+9i \\ 7i & 7-i & -7i \end{pmatrix}, \ D_1 = \begin{pmatrix} 7+5i & 4-3i & 5 \\ 8i & 4+8i & 9-3i \\ 0 & 5i & 5 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 7 & 0 & 3i \\ 1+9i & 8i & 4+6i \\ 6-9i & 0 & 8-8i \end{pmatrix}, \ B_2 = \begin{pmatrix} 3i & 6 & i \\ 4 & 5 & 9 \\ 3 & 7-2i & 5 \end{pmatrix}, \ C_2 = \begin{pmatrix} 7+7i & 2 & 7i \\ i & 5 & 4+i \\ 0 & 7 & 5+i \end{pmatrix},$$

$$D_2 = \begin{pmatrix} 3-i & 7-11i & 0 \\ 6i & 0 & 0 \\ 1+i & 2 & 8 \end{pmatrix}, \ E_1 = \begin{pmatrix} -112+292i & 412+342i & 374+482i \\ -366-194i & -478+786i & 290-366i \\ 118+226i & 98+184i & 270-70i \end{pmatrix},$$

$$E_2 = \begin{pmatrix} 6+214i & 254+236i & 120+600i \\ 180-384i & -119-369i & 146-378i \\ 18-64i & -15-145i & -20-262i \end{pmatrix}.$$

Here, we choose the initial matrices $X_j^{(1)} \in ACJC^{3\times 3}(J), S_j^{(1)} \in ACJC^{3\times 3}(J)$, $(j = 1, 2)$, as

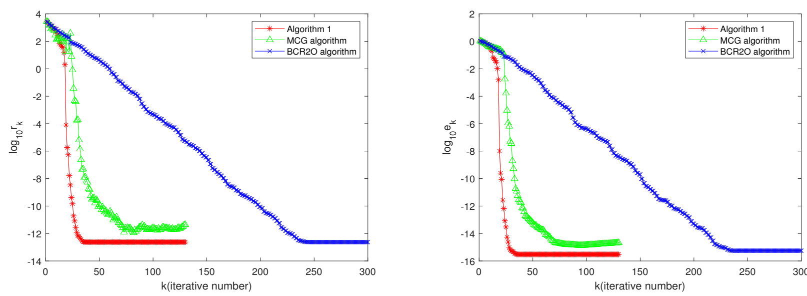
$$X_1^{(1)} = X_2^{(1)} = \begin{pmatrix} 0 & \mathrm{i} & 0 \\ -i & 0 & i \\ 0 & -i & 0 \end{pmatrix}, \ S_1^{(1)} = S_2^{(1)} = \begin{pmatrix} 0 & \mathrm{i} & 0 \\ -i & 0 & i \\ 0 & -i & 0 \end{pmatrix}, \ J = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Let $X_1^* = \begin{pmatrix} 2i & 8i & 0 \\ 8i & 0 & -8i \\ 0 & -8i & -2i \end{pmatrix}, X_2^* = \begin{pmatrix} 0 & -2i & -2i \\ -i & 0 & i \\ 2i & 2i & 0 \end{pmatrix}$. By Algorithm 1, we get

$$X_1^{(44)} = \begin{pmatrix} 1.9999i & 7.9999i & 0.0000 \\ 7.9999i & 0.0000 & -7.9999i \\ 0.0000 & -7.9999i & -2.0000i \end{pmatrix} \in ACJC^{3\times 3}(J),$$

$$X_2^{(44)} = \begin{pmatrix} 0.0000 & -2.0000i & -2.0000i \\ -1.0000i & 0.0000 & 1.0000i \\ 2.0000i & 2.0000i & 0.0000 \end{pmatrix} \in ACJC^{3\times 3}(J).$$

In Fig.3, the convergence performance comparison results of the residual and relative error of Algorithm 1, MCG algorithm and BCR2O algorithm are demonstrated. By Fig.3 (a) and Fig.3 (b), it can be seen that Algorithm 1 converges faster than MCG algorithm and BCR2O algorithm, and Algorithm 1 and MCG algorithm have better convergence accuracy than BCR2O algorithm. From Fig.3, we can draw a conclusion that with the increase of iterative step, the residual gradually tends to be stable, which means that Algorithm 1 is convergent and effective.



(a) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for residual

(b) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for relative error

Fig. 3. Comparison of convergence curves for Example 4.3

In Table 5 and Table 6, we give the relationship between the iterative step and computational time of Algorithm 1, MCG algorithm and BCR2O algorithm on the similar residual and relative error. It can be clearly seen that Algorithm 1 requires the least number of iterative steps for similar residual and relative error.

Example 4.4. We solve the centrosymmetric solutions of generalized coupled equations

$$\begin{cases} A_{11}X_1B_{11} + A_{12}X_2B_{12} + A_{13}X_3B_{13} = M_1, \\ A_{21}X_1B_{21} + A_{22}X_2B_{22} + A_{23}X_3B_{23} = M_2, \end{cases}$$

with parametric matrices

$$A_{11} = -triu(rand(m,m),1) + diag(37 + diag(rand(m))),$$

$$B_{11} = -triu(rand(m,m),1) + diag(57 + diag(rand(m))),$$

$$A_{12} = -triu(rand(m,m),1) - diag(73 + diag(rand(m))),$$

Table 5. Iterative step, residual and computational time of Fig.3

Method						
Algorithm 1	$\log_{10} r_k$	−11.0798	−10.7152	−9.3829	−8.4575	−7.7838
	Steps	27	26	24	23	22
	Time (s)	0.0326	0.0318	0.0302	0.0300	0.0299
MCG algorithm	$\log_{10} r_k$	−11.0179	−10.0331	−9.2481	−8.1853	−7.3318
	Steps	63	49	42	38	34
	Time (s)	0.0179	0.0171	0.0168	0.0167	0.0165
BCR2O algorithm	$\log_{10} r_k$	−11.0699	−10.0575	−9.1060	−8.1707	−7.2459
	Steps	215	199	181	167	155
	Time (s)	0.0491	0.0430	0.0416	0.0404	0.0397

Table 6. Iterative step, relative error and computational time of Fig.3

Method						
Algorithm 1	$\log_{10} e_k$	−14.2750	−13.0706	−12.1851	−11.5657	−10.0645
	Steps	26	24	23	22	21
	Time (s)	0.0379	0.0376	0.0375	0.0374	0.0371
MCG algorithm	$\log_{10} e_k$	−14.0800	−13.0806	−12.1299	−11.2427	−10.0157
	Steps	62	48	40	36	33
	Time (s)	0.0214	0.0194	0.0192	0.0184	0.0178
BCR2O algorithm	$\log_{10} e_k$	−14.0894	−13.1151	−12.1649	−11.0526	−10.0202
	Steps	213	197	181	163	151
	Time (s)	0.0442	0.0441	0.0400	0.0393	0.0386

$$B_{12} = -triu(rand(m,m),1) + diag(7 + diag(rand(m))),$$

$$A_{13} = -triu(rand(m,m),1) - diag(100 + diag(rand(m))),$$

$$B_{13} = diag(70 + diag(rand(m))),$$

$$A_{21} = triu(rand(m,m),1) + diag(60 + diag(rand(m))),$$

$$B_{21} = -triu(rand(m,m),1) + diag(77 + diag(rand(m))),$$

$$A_{22} = -triu(rand(m,m),1) - diag(27 + diag(rand(m))),$$

$$B_{22} = -triu(rand(m,m),1) + diag(39 + diag(rand(m)),$$

$$A_{23} = -triu(rand(m,m),1) - diag(99 + diag(rand(m))),$$

$$B_{23} = diag(33 + diag(rand(m))),$$

$$M_1 = rand\,(m)\,,\ \ M_2 = rand\,(m)\,.$$

We choose the initial matrix $X_j^{(1)}, S_j^{(1)} \in CJC^{m\times m}(J)$ as $X_j^{(1)}, S_j^{(1)} = 1^{m\times m}$, $j = 1,2,3$, $J = flipud\,(eye\,(m))$, and let $X_1^* = 1^{m\times m}, X_2^* = rand\,(m) + J \cdot rand\,(m) \cdot J, X_3^* = 2eye\,(m)$.

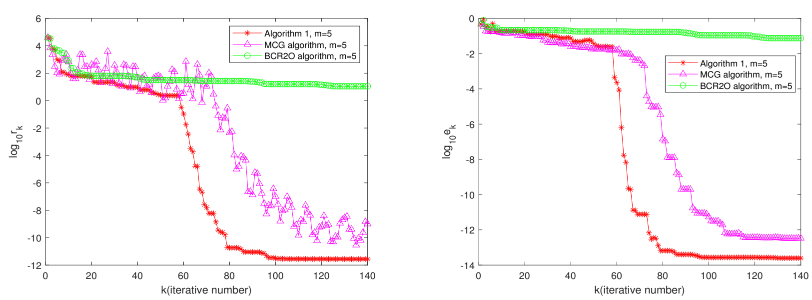
Through Algorithm 1, we get

$$X_1^{(130)} = \begin{pmatrix} 0.9999 & 0.9999 & 0.9999 & 0.9999 & 0.9999 \\ 0.9999 & 0.9999 & 0.9999 & 0.9999 & 0.9999 \\ 1.0000 & 1.0000 & 0.9999 & 1.0000 & 1.0000 \\ 0.9999 & 0.9999 & 0.9999 & 0.9999 & 0.9999 \\ 0.9999 & 0.9999 & 0.9999 & 0.9999 & 0.9999 \end{pmatrix} \in CJC^{m\times m}(J),$$

$$X_2^{(130)} = \begin{pmatrix} 1.4846 & 0.4966 & 1.1735 & 0.8646 & 1.3744 \\ 1.7796 & 0.9727 & 0.9352 & 0.7574 & 1.7259 \\ 0.3417 & 0.8438 & 1.7670 & 0.8438 & 0.3417 \\ 1.7259 & 0.7574 & 0.9353 & 0.9727 & 1.7796 \\ 1.3744 & 0.8646 & 1.1735 & 0.4966 & 1.4846 \end{pmatrix} \in CJC^{m\times m}(J),$$

$$X_3^{(130)} = \begin{pmatrix} 2.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 2.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 2.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 2.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 2.0000 \end{pmatrix} \in CJC^{m\times m}(J).$$

In Fig.4, according to the Definition 1.2, r_k and e_k of Algorithm 1, MCG algorithm and BCR2O algorithm are given. According to Fig.4(a) and Fig.4(b), Algorithm 1 converges faster and has better convergence accuracy than MCG algorithm and BCR2O algorithm, which means that Algorithm 1 is convergent and effective in solving the central symmetric solution.



(a) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for residual

(b) Comparison of Algorithm 1, MCG algorithm and BCR2O algorithm for relative error

Fig. 4. Comparison of convergence curves for Example 4.4

In Table 7 and Table 8, we give the relationship between the iterative step and the computational time of Algorithm 1, MCG algorithm and BCR2O algorithm on similar residual and relative error. It can be seen more clearly that Algorithm 1 requires fewer iterative steps, and BCR20 requires the largest number of iterative steps both in terms of residual and relative error.

Table 7. Iterative step, residual and computational time of Fig.4

Method						
Algorithm 1	$\log_{10} r_k$	−10.7091	−9.4503	−8.2242	−7.5802	−6.4563
	Steps	79	75	71	69	67
	Time (s)	0.0427	0.0423	0.0423	0.0419	0.0417
MCG algorithm	$\log_{10} r_k$	−10.2239	−9.1721	−8.2741	−7.5080	−6.6601
	Steps	118	108	96	95	88
	Time (s)	0.0234	0.0233	0.0233	0.0232	0.0228
BCR2O algorithm	$\log_{10} r_k$	−10.0122	−9.0263	−8.0236	−7.0083	−6.0066
	Steps	4566	3617	3283	2415	2165
	Time (s)	0.3868	0.2967	0.2555	0.2030	0.1859

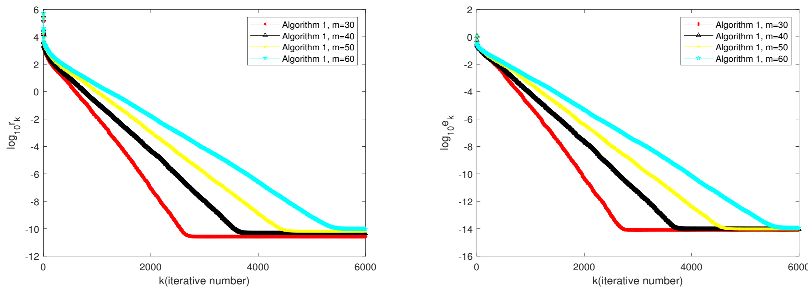
Table 8. Iterative step, relative error and computational time of Fig.4

Method						
Algorithm 1	$\log_{10} e_k$	−13.1816	−12.1682	−11.1107	−10.8798	−9.6489
	Steps	79	74	69	67	65
	Time (s)	0.0481	0.0468	0.0440	0.0419	0.0419
MCG algorithm	$\log_{10} e_k$	−13.0076	−12.1675	−11.0572	−10.7582	−9.7025
	Steps	331	107	95	93	88
	Time (s)	0.0336	0.0260	0.0240	0.0230	0.0230
BCR2O algorithm	$\log_{10} e_k$	−13.0062	−12.0056	−11.0012	−10.0288	−9.1210
	Steps	5598	4562	3539	3287	2349
	Time (s)	0.4584	0.3703	0.2898	0.2782	0.2009

From Fig.5(a) and Fig.5(b), we can clearly see that with the increase of the value of m, the convergence speed of residual and iterative error gradually slows down. With the continuous increase of matrix dimension, when the residual tends to be stable, the number of iterative steps required increases, and the running time of the program becomes longer. How to reduce the calculation time and amount, and how to improve and optimize the algorithm are the problems that we will continue to study in the future.

In Table 9 and Table 10, we give the relationship between the iterative step and the calculation time of Algorithm 1 on the similar residual and relative error in different m values. It can be seen that under the action of Algorithm 1, the number of iterative steps increases with the increase of m value on the residual and relative error.

In this section, four kinds of constrained solutions (symmetric solution, reflexive solution, centrosymmetric solution and anti-centrosymmetric solution) of coupled operator matrix equations are solved respectively. From the corresponding figures of the above four numerical examples obtained through Algorithm 1, it can be clearly concluded that with the increase of the number of steps, the residual and the relative error are gradually tend to be stable, which verifies the convergence and



(a) Comparison of m=30,40,50,60 in Algorithm 1 for residual (b) Comparison of m=30,40,50,60 in Algorithm 1 for relative error

Fig. 5. Comparison of convergence curves for different values of m in Example 4.4

Table 9. Iterative step, residual and computational time of Fig.5

Method						
Algorithm 1 with $m = 30$	$\log_{10} r_k$	-10.0033	-9.0165	-8.0114	-7.0040	-6.0003
	Steps	2548	2375	2175	1996	1810
	Time (s)	0.7812	0.7505	0.6611	0.6126	0.5773
Algorithm 1 with $m = 40$	$\log_{10} r_k$	-10.0018	-9.0024	-8.0000	-7.0020	-6.0001
	Steps	3558	3292	3015	2735	2474
	Time (s)	1.8870	1.6815	1.5084	1.3714	1.2546
Algorithm 1 with $m = 50$	$\log_{10} r_k$	-10.0070	-9.0009	-8.0109	-7.0326	-6.0042
	Steps	4419	4035	3707	3377	3025
	Time (s)	4.6694	4.1888	3.8699	3.5581	3.1489
Algorithm 1 with $m = 60$	$\log_{10} r_k$	-10.0019	-9.0022	-8.0029	-7.0006	-6.0007
	Steps	5563	4943	4548	4156	3786
	Time (s)	13.5856	11.9349	10.8805	9.7150	8.4794

Table 10. Iterative step, relative error and computational time of Fig.5

Method						
Algorithm 1 with $m = 30$	$\log_{10} e_k$	-13.0146	-12.0090	-11.0058	-10.0001	-9.0016
	Steps	2495	2331	2138	1947	1759
	Time (s)	0.9564	0.9261	0.8596	0.7671	0.6586
Algorithm 1 with $m = 40$	$\log_{10} e_k$	-13.0066	-12.0004	-11.0028	-10.0028	-9.0095
	Steps	3451	3196	2911	2629	2373
	Time (s)	1.7897	1.5910	1.4696	1.3549	1.2876
Algorithm 1 with $m = 50$	$\log_{10} e_k$	-13.0002	-12.0035	-11.0029	-10.0081	-9.0054
	Steps	4218	3874	3558	3200	2874
	Time (s)	4.3644	4.1643	3.7216	3.4211	3.0674
Algorithm 1 with $m = 60$	$\log_{10} e_k$	-13.0045	-12.0001	-11.0039	-10.0022	-9.0027
	Steps	5140	4723	4321	3944	3561
	Time (s)	11.7156	10.7456	9.4401	8.1944	7.1374

effectiveness of Algorithm 1. From Table 1 to Table 8, the convergence speed of Algorithm 1 is faster than that of MCG algorithm and BCR2O algorithm in terms of similar residual and relative error.

5. Concluding remark

In this present work, we provide a biconjugate residual (BCR) method for obtaining coupled matrix equation with submatrix constraints by introducing operators. The presented new algorithm can solve many matrix equations and many constraints solutions, for example symmetric solution, reflective solution and centro-symmetric solution. Compared with the algortihm in [24], the provided new algorithm can solve the constraint solution of coupled matrix equations in complex field. In additon, the sufficient conditions for the convergence of new BCR algorithm are given. Some errors or typos in [24] have been corrected. Some numerical examples are provided to illustrate the effectiveness and superiority of new algorithm.

References

[1] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*, Courier Corporation, 2007.

[2] S. Barnett and C. Storey, *Some applications of the Lyapunov matrix equation*, IMA Journal of Applied Mathematics, 1968, 4(1), 33–42.

[3] A. Bouhamidi and K. Jbilou, *A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications*, Applied Mathematics and Computation, 2008, 206(2), 687–694.

[4] D. C. Chen, Y. N. Zhang and S. Li, *Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method*, IEEE Transactions on Industrial Informatics, 2017, 14(7), 3044–3053.

[5] Y. C. Chen, K. Q. Sun, B. Beker and R. Mittra, *Unified matrix presentation of Maxwell's and wave equations using generalized differential matrix operators [EM engineering education]*, IEEE Transactions on Education, 1998, 41(1), 61–69.

[6] M. Dehghan and M. Hajarian, *Finite iterative algorithms for the reflexive and anti-reflexive solutions of the matrix equation* $A_1X_1B_1 + A_2X_2B_2 = C$, Mathematical and Computer Modelling, 2009, 49(9–10),1937–1959.

[7] M. Dehghan and M. Hajarian, *On the reflexive solutions of the matrix equation* $AXB+ CYD= E$, Bulletin of the Korean Mathematical society, 2009, 46(3), 511–519.

[8] M. Dehghan and M. Hajarian. *The general coupled matrix equations over generalized bisymmetric matrices*, Linear Algebra and its Applications, 2010, 432(6), 1531–1552.

[9] M. Delphi and S. Shihab, *Operational matrix basic spline wavelets of derivative for linear optimal control problem*, Electronics Science Technology and Application, 2019, 6(2), 18–24.

[10] Ç. Demir and Ö. Civalek, *A new nonlocal FEM via Hermitian cubic shape functions for thermal vibration of nano beams surrounded by an elastic matrix*, Composite Structures, 2017, 168, 872–884.

[11] A. M. Diwekar and R. K. Yedavalli, *Smart structure control in matrix second-order form*, Smart materials and Structures, 1995, 2442, 24–34.

[12] Z. Hailin, *An iterative method for symmetric solutions of the matrix equation AXB+ CXD= F*, Mathematica Numerica Sinica, 2010. DOI: 10.3788/HPLPB20102206.1351.

[13] M. Hajarian, *Symmetric solutions of the coupled generalized Sylvester matrix equations via BCR algorithm*, Journal of The Franklin Institute, 2016, 353(13), 3233–3248.

[14] M. Hajarian, *Convergence of HS version of BCR algorithm to solve the generalized Sylvester matrix equation over generalized reflexive matrices*, Journal of the Franklin Institute, 2017, 354(5), 2340–2357.

[15] E. J. Haug and K. K. Choi, *Structural design sensitivity analysis with generalized global stiffness and mass matrices*, AIAA journal, 1984, 22(9), 1299–1303.

[16] Z. H. He, W. L. Qin and X. X. Wang, *Some applications of a decomposition for five quaternion matrices in control system and color image processing*, Computational and Applied Mathematics, 2021. DOI: 10.1007/S40314-021-01579-3.

[17] Z. H. He, X. X. Wang and Y. F. Zhao, *Eigenvalues of quaternion tensors with applications to color video processing*, Journal of Scientific Computing, 2022. DOI: 10.1007/S10915-022-02058-5.

[18] L. Jin, S. Li, B. Hu and J. G. Yu, *A noise-suppressing neural algorithm for solving the time-varying system of linear equations: A control-based approach*, IEEE Transactions on Industrial Informatics, 2018, 15(1), 236–246.

[19] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Courier Corporation, 2012.

[20] V. N. Katsikis, S. D. Mourtas, P. S. Stanimirovic and Y. N. Zhang, *Solving complex-valued time-varying linear matrix equations via QR decomposition with applications to robotic motion tracking and on angle-of-arrival localization*, IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(8), 3415–3424.

[21] L. Lapidus and G. F. Pinder, *Numerical solution of partial differential equations in science and engineering*, John Wiley & Sons, 1999.

[22] W. B. Li, *A recurrent neural network with explicitly definable convergence time for solving time-variant linear matrix equations*, IEEE Transactions on Industrial Informatics, 2018, 14(12), 5289–5298.

[23] Z. Li and Y. N. Zhang, *Improved Zhang neural network model and its solution of time-varying generalized linear matrix equations*, Expert Systems with Applications, 2010, 37(10), 7213–7218.

[24] C. Q. Lv and C. F. Ma, *BCR method for solving generalized coupled Sylvester equations over centrosymmetric or anti-centrosymmetric matrix*, Computers & Mathematics with Applications, 2018, 75(1), 70–88.

[25] V. Mehrmann, *Matrix analysis for scientists and engineers [Book Review]*, IEEE Control Systems Magazine, 2006, 26(2), 94–95.

[26] N. Mikaeilvand, *On solvability of fuzzy system of linear matrix equations*, J. Appl. Sci. Res., 2011, 7(2), 141–153.

[27] Y. X. Peng, X. Y. Hu and L. Zhang, *An iteration method for the symmetric solutions and the optimal approximation solution of the matrix equation $AXB=C$*, Applied Mathematics and Computation, 2005, 160(3), 763–777.

[28] Z. H. Peng, *The reflexive least squares solutions of the matrix equation $A_1X_1B_1+A_2X_2B_2+\cdots+A_lX_lB_l=C$ with a submatrix constraint*, Numerical Algorithms, 2013, 64(3), 455–480.

[29] Z. Y. Peng, *An iterative method for the least squares symmetric solution of the linear matrix equation $AXB=C$*, Applied Mathematics and Computation, 2005, 170(1), 711–723.

[30] S. Sana and V. S. Rao, *Application of linear matrix inequalities in the control of smart structural systems*, Journal of intelligent material systems and structures, 2000, 11(4), 311–323.

[31] C. Q. Song, *Iterative method to the coupled operator matrix equations with submatrix constraint and its application in control*, Transactions of the Institute of Measurement and Control, 2021, 43(3), 597–611.

[32] M. T. Vespucci and C. G. Broyden, *Implementation of different computational variations of biconjugate residual methods*, Computers & Mathematics with Applications, 2001, 42(8–9), 1239–1253.

[33] L. Xiao, B. L. Liao, S. Li and K. Chen, *Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations*, Neural Networks, 2018, 98, 102–113.

[34] Y. J. Xie, N. Huang and C. F. Ma, *Iterative method to solve the generalized coupled Sylvester-transpose linear matrix equations over reflexive or anti-reflexive matrix*, Computers and Mathematics with Applications, 2014, 67(11), 2071–2084.

[35] Y. J. Xie and C. F. Ma, *Iterative methods to solve the generalized coupled Sylvester-conjugate matrix equations for obtaining the centrally symmetric (centrally antisymmetric) matrix solutions*, Journal of Applied Mathematics, 2014, 2014(1), 1–17.

[36] C. F. Yi, Y. H. Chen and Z. L. Lu, *Improved gradient-based neural networks for online solution of Lyapunov matrix equation*, Information processing letters, 2011, 111(16), 780–786.

[37] S. W. Yu, Z. H. He, T. C. Qi and X. X. Wang, *The equivalence canonical form of five quaternion matrices with applications to imaging and Sylvester-type equations*, Journal of Computational and Applied Mathematics, 2021. DOI: 10.1016/J.CAM.2021.113494.

[38] H. M. Zhang, *A finite iterative algorithm for solving the complex generalized coupled Sylvester matrix equations by using the linear operators*, Journal of the Franklin Institute, 2017, 354(4), 1856–1874.

[39] J. F. Zhang, *Optimal control for mechanical vibration systems based on second-order matrix equations*, Mechanical Systems and Signal Processing, 2002, 16(1), 61–67.