Filomat 37:30 (2023), 10249–10264
https://doi.org/10.2298/FIL2330249L

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

Two algorithms for solving generalized coupled Sylvester tensor equations

Tao Li[a,*], Chi-Hua Feng[a], Xin-Fang Zhang[a]

[a] *Department of Mathematics, Key Laboratory of Engineering Modeling and Statistical Computation of Hainan Province, Hainan University, Haikou 570228, P. R. China*

Abstract. In this paper, we consider the generalized coupled Sylvester tensor equations by the tensor forms of the biconjugate A-orthogonal residual and the conjugate A-orthogonal residual squared algorithms. With the absence of round-off errors, we show that our methods converge to the exact solution group within finite steps when they are consistent. Finally, we provide some numerical examples to demonstrate the effectiveness of the proposed methods, including when testing the algorithms by color image restoration problems and randomly generated data.

1. Introduction

Throughout this paper, we denote the tensors as Euler script letters, e.g., $\mathcal{X}$. An order n dimension $I_1 \times I_2 \times \cdots \times I_n$ tensor $\mathcal{X} = (x_{i_1 i_2 \cdots i_n})$ is a multidimensional array consisting of $I_1 I_2 \cdots I_n$ entries with $1 \le i_j \le I_j, 1 \le j \le n$. A mode-1 fiber of $\mathcal{X} = (x_{i_1 i_2 \cdots i_n})$ is defined by fixing every index except for i_1 ($i_1 = 1, \cdots, I_1$), denoted by $x_{:i_2 \cdots i_n}$. We denote the matrices as capital letters, e.g., A. Let $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ and $\mathbb{R}^{I_1 \times I_2}$ be the sets of all tensors and matrices over the real number field $\mathbb{R}$, respectively. The transpose of a matrix $A \in \mathbb{R}^{I_1 \times I_2}$ is denoted by $A^T \in \mathbb{R}^{I_2 \times I_1}$. A tensor $\mathcal{X} = (x_{i_1 i_2 \cdots i_n})$ is called the zero tensor if $x_{i_1 i_2 \cdots i_n} = 0$, denoted by $\mathcal{O}$. The k-mode product of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ with a matrix $A \in \mathbb{R}^{J \times I_k}$ [1, 2], denoted by $\mathcal{X} \times_k A$, is a tensor of size $I_1 \times \cdots \times I_{k-1} \times J \times I_{k+1} \times \cdots \times I_n$ with its entries given by

$$(\mathcal{X} \times_k A)_{i_1 \times \cdots \times i_{k-1} \times j \times i_{k+1} \times \cdots \times i_n} = \sum_{i_k=1}^{I_k} x_{i_1 i_2 \cdots i_n} a_{j i_k}, \qquad j = 1, 2, \cdots, J.$$

Given two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, their inner product is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_n} y_{i_1 i_2 \cdots i_n}.$$

2020 *Mathematics Subject Classification*. Primary 15A69; Secondary 65F10.
Keywords. Tensor form; Generalized coupled Sylvester tensor equations; Biconjugate A-orthogonal residual algorithm; Conjugate A-orthogonal residual squared algorithm.
Received: 07 February 2023; Accepted: 30 June 2023
Communicated by Dragana Cvetković-Ilić
Research supported by the Hainan Provincial Natural Science Foundation of China [grant numbers 122QN214 and 122MS001], and the Academic Programs project of Hainan University [grant numbers KYQD(ZR)-21119 and KYQD(ZR)-21151].
* Corresponding author: Tao Li
Email addresses: tli@hainanu.edu.cn (Tao Li), fengchihua@163.com (Chi-Hua Feng), 995272@hainanu.edu.cn (Xin-Fang Zhang)

So the Frobenius norm over $\mathbb{R}^{I_1\times I_2\times\cdots\times I_n}$ is naturally defined as

$$\|\mathcal{X}\| = \sqrt{\langle\mathcal{X},\mathcal{X}\rangle} = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_n=1}^{I_n} x_{i_1 i_2\cdots i_n}^2}.$$

If two tensors $\mathcal{X}$ and $\mathcal{Y}$ yield to $\langle\mathcal{X},\mathcal{Y}\rangle = 0$, then they are said to be orthogonal. Let $\mathcal{X}\in\mathbb{R}^{I_1\times\cdots\times I_{k-1}\times P\times I_{k+1}\times\cdots\times I_n}$, $\mathcal{Y}\in\mathbb{R}^{I_1\times\cdots\times I_{k-1}\times Q\times I_{k+1}\times\cdots\times I_n}$ and $M\in\mathbb{R}^{Q\times P}$. Then

$$\langle\mathcal{X}\times_k M,\mathcal{Y}\rangle = \langle\mathcal{X},\mathcal{Y}\times_k M^T\rangle.$$

Let $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_n}$, $M_1,\ M_k\in\mathbb{R}^{J_1\times I_k},\ M_2\in\mathbb{R}^{P\times J_1}$ and $M_l\in\mathbb{R}^{J_2\times I_l}$ $(k\neq l)$. Then

$$\begin{cases}\mathcal{X}\times_k M_1\times_k M_2 = \mathcal{X}\times_k (M_2M_1),\\ \mathcal{X}\times_k M_k\times_l M_l = \mathcal{X}\times_l M_l\times_k M_k.\end{cases}$$

For more details about the properties of tensors, see [3–8].

In this paper, we are interested in solving the following generalized coupled Sylvester tensor equations (GCSTEs)

$$\begin{cases}\mathcal{X}_1\times_1 A_{11}+\mathcal{X}_2\times_2 A_{12}+\cdots+\mathcal{X}_{n-1}\times_{n-1}A_{1(n-1)}+\mathcal{X}_n\times_n A_{1n}=\mathcal{B}_1,\\ \mathcal{X}_2\times_1 A_{21}+\mathcal{X}_3\times_2 A_{22}+\cdots+\mathcal{X}_n\times_{n-1}A_{2(n-1)}+\mathcal{X}_1\times_n A_{2n}=\mathcal{B}_2,\\ \quad\vdots\qquad\qquad\qquad\vdots\qquad\qquad\qquad\vdots\\ \mathcal{X}_n\times_1 A_{n1}+\mathcal{X}_1\times_2 A_{n2}+\cdots+\mathcal{X}_{n-2}\times_{n-1}A_{n(n-1)}+\mathcal{X}_{n-1}\times_n A_{nn}=\mathcal{B}_n,\end{cases}\tag{1}$$

where $A_{ij}\in\mathbb{R}^{I_j\times I_j}$, $\mathcal{B}_i\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_n}$ are given, and tensors $\mathcal{X}_j\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_n}$ are required to be determined, $i,j=1,\cdots,n$. When $i=1$ and $\mathcal{X}_1=\mathcal{X}_2=\cdots=\mathcal{X}_n=\mathcal{X}$, Eqs.(1) reduce to the Sylvester tensor equation (STE)

$$\mathcal{X}\times_1 A_1+\mathcal{X}\times_2 A_2+\cdots+\mathcal{X}\times_n A_n=\mathcal{B}.\tag{2}$$

It was not merely applied to system and control theory [9–13], but also extensively penetrated the heat transfer [14], partial differential equation [15], finite difference [16], finite element [17]. One application of STE is to solve the convection-diffusion equation [18, 20]

$$\begin{aligned}-v\Delta u+c^T\nabla u &= f \ \text{ in }\ \Gamma=[0,1]^n,\\ u &= 0 \ \text{ on }\ \partial\Gamma.\end{aligned}\tag{3}$$

From the finite difference method, Eq.(3) is discreted as Eq.(2) with parameters

$$A_i=\frac{v}{h^2}\begin{bmatrix}2&-1&&&\\-1&2&-1&&\\&\ddots&\ddots&\ddots&\\&&-1&2&-1\\&&&2&-1\end{bmatrix}_{m\times m}+\frac{c_i}{4h}\begin{bmatrix}3&-5&1&&\\1&3&-5&&\\&\ddots&\ddots&\ddots&1\\&&1&3&-5\\&&&1&3\end{bmatrix}_{m\times m},\tag{4}$$

where $h=\frac{1}{m+1}$ is the mesh size, $i=1,\cdots,n$. Indeed, the GCSTEs, as an extension of STE, are quite general and have many important potential applications. However, there is little literature on studying the iterative solution of GCSTEs. It motivates us to establish efficient algorithms for solving the original tensor equations.

Various effective algorithms for solving the STE have been developed in the last decade. From the hierarchical identification principle, Chen and Lv [18] proposed a gradient-based iterative method and its modification version for solving the third-order Sylvester tensor equation when it has a unique solution.

They also [19] developed a generalized minimum residual method and its preconditioned version in their tensor forms for solving Eq.(2). After that, from the Arnoldi process, Ali Beik et al. [20] introduced the conjugate gradient and nested conjugate gradient methods to search for a solution of Eq.(2), respectively. A new method [21] based on the global Bidiag 1 process was also proposed for obtaining its solutions of Eq.(2). Huang and Li [22] derived some Krylov subspace methods, including the conjugate residual, generalized conjugate residual, biconjugate gradient, conjugate gradient squared, and biconjugate gradient stabilized methods in their tensor forms for solving a tensor equation. In [23], Najafi-Kalyani et al. formulated some effective methods for solving the tensor equation (2), which are based on the tensor form of the global Hessenberg process. According to the CP decomposition, Bentbib et al. [24] proposed the block and global Arnoldi-based methods for solving the tensor equation (2) efficiently, which are promising methods. Heyouni et al. [25] established a new iterative method in its tensor form for solving Eq.(2). Besides, Wang and He [26–34] studied the solvability of a series of Sylvester quaternion tensor equations, and also presented their general solution expressions. To the best of our knowledge, only Lv and Ma [35] presented a modified conjugate gradient (MCG) method for investigating the iterative solutions of GCSTEs. So we in this paper intend to develop a biconjugate A-orthogonal residual method in its tensor form for solving the GCSTEs (1). With the absence of round-off errors, we prove that the method mentioned converges to the solutions for any initial value at most finite iteration steps. A transpose-free variant of the proposed method is also established to improve its performance further.

The organization of this paper is as follows. In Section 2, we propose the biconjugate A-orthogonal residual method based on the tensor format (BiCOR_BTF) for solving the generalized coupled Sylvester tensor equations, the convergence of which is also established under certain assumptions. Moreover, we develop the tensor form of the conjugate A-orthogonal residual squared method, which converges much faster than BiCOR_BTF. We provide some numerical examples to show the efficiency of our methods compared with MCG in Section 3. Numerical results of the proposed methods applied to some color image restoration problems are also recorded in this section. Finally, we give a brief conclusion in Section 4.

2. Two iterative algorithms for solving GCSTEs (1)

This section proposes the tensor forms of the biconjugate A-orthogonal residual method and its transpose-free variant for solving the GCSTEs (1). As stated in [35], by the properties of k-mode product and Kronecker product [1, 4, 36], the GCSTEs (1) can be rewritten equivalently as the following linear system

$$Ax = b, \tag{5}$$

where

$$A = \begin{bmatrix} I^{I_n} \otimes \cdots \otimes I^{I_2} \otimes A_{11} & I^{I_n} \otimes \cdots \otimes A_{12} \otimes I^{I_1} & \cdots & A_{1n} \otimes I^{I_{n-1}} \otimes \cdots \otimes I^{I_1} \\ A_{2n} \otimes I^{I_{n-1}} \otimes \cdots \otimes I^{I_1} & I^{I_n} \otimes \cdots \otimes I^{I_2} \otimes A_{21} & \cdots & I^{I_n} \otimes A_{2(n-1)} \otimes \cdots \otimes I^{I_1} \\ \vdots & \vdots & \ddots & \vdots \\ I^{I_n} \otimes \cdots \otimes A_{n2} \otimes I^{I_1} & I^{I_n} \otimes \cdots \otimes A_{n3} \otimes I^{I_2} \otimes I^{I_1} & \cdots & I^{I_n} \otimes \cdots \otimes I^{I_2} \otimes A_{n1} \end{bmatrix},$$

$$x = \begin{bmatrix} \mathrm{vec}(\mathcal{X}_1) \\ \mathrm{vec}(\mathcal{X}_2) \\ \vdots \\ \mathrm{vec}(\mathcal{X}_n) \end{bmatrix}, \qquad b = \begin{bmatrix} \mathrm{vec}(\mathcal{B}_1) \\ \mathrm{vec}(\mathcal{B}_2) \\ \vdots \\ \mathrm{vec}(\mathcal{B}_n) \end{bmatrix},$$

and the operator vec$(\cdot)$ represents a vectorized tensor by stacking its mode-1 fiber to form a vector.

We call that the tensor equations (1) are consistent if and only if Eq.(5) is consistent. It is well known that there are many effective Krylov subspace methods for solving Eq.(5). As shown in [37–39], the biconjugate A-orthogonal residual (BiCOR) and the conjugate A-orthogonal residual squared (CORS) algorithms for solving a linear system outperform some Krylov subspace methods. So we here utilize the BiCOR and CORS algorithms, based on the biconjugate A-orthonormalization procedure, to solve Eq.(5) instead of Eqs.(1).

The procedure is built upon the two-sided Lanczos method and can be regarded as a Petrov-Galerkin projection technique between the dual Krylov subspaces $K_m(A; v_1)$ and $A^T K_m(A^T; \omega_1)$ with v_1 and ω_1 being two column vectors of appropriate dimensions. Nevertheless, the computational work and memory space required to implement the two algorithms are costly with the dimension increases. As a result, we intend to propose the tensor forms of the BiCOR and CORS algorithms for solving Eqs.(1). Unless otherwise stated, we always assume that the tensor equations (1) are consistent. Moreover, we give a useful lemma that plays a key role in the proposed algorithms.

Lemma 2.1. If $\mathcal{L}_i(\mathcal{X}_1, \cdots, \mathcal{X}_n)$ and $\widetilde{\mathcal{L}}_i(\mathcal{Y}_1, \cdots, \mathcal{Y}_n)(i = 1, 2, \cdots, n)$ are linear operators on $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, then

$$\begin{aligned} &\langle \mathcal{Y}_1, \mathcal{L}_1(\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_n)\rangle + \langle \mathcal{Y}_2, \mathcal{L}_2(\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_n)\rangle + \cdots + \langle \mathcal{Y}_n, \mathcal{L}_n(\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_n)\rangle \\ &= \langle \mathcal{X}_1, \widetilde{\mathcal{L}}_1(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_n)\rangle + \langle \mathcal{X}_2, \widetilde{\mathcal{L}}_2(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_n)\rangle + \cdots + \langle \mathcal{X}_n, \widetilde{\mathcal{L}}_n(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_n)\rangle, \end{aligned} \tag{6}$$

where

$$\begin{cases} \mathcal{L}_1(\mathcal{X}_1, \cdots, \mathcal{X}_n) = \mathcal{X}_1 \times_1 A_{11} + \mathcal{X}_2 \times_2 A_{12} + \cdots + \mathcal{X}_n \times_n A_{1n}, \\ \mathcal{L}_2(\mathcal{X}_1, \cdots, \mathcal{X}_n) = \mathcal{X}_2 \times_1 A_{21} + \mathcal{X}_3 \times_2 A_{22} \cdots + \mathcal{X}_1 \times_n A_{2n}, \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \mathcal{L}_n(\mathcal{X}_1, \cdots, \mathcal{X}_n) = \mathcal{X}_n \times_1 A_{n1} + \mathcal{X}_1 \times_2 A_{n2} + \cdots + \mathcal{X}_{n-1} \times_n A_{nn} \end{cases} \tag{7}$$

and

$$\begin{cases} \widetilde{\mathcal{L}}_1(\mathcal{Y}_1, \cdots, \mathcal{Y}_n) = \mathcal{Y}_1 \times_1 A_{11}^T + \mathcal{Y}_2 \times_n A_{2n}^T + \cdots + \mathcal{Y}_n \times_2 A_{n2}^T, \\ \widetilde{\mathcal{L}}_2(\mathcal{Y}_1, \cdots, \mathcal{Y}_n) = \mathcal{Y}_1 \times_2 A_{12}^T + \mathcal{Y}_2 \times_1 A_{21}^T \cdots + \mathcal{Y}_n \times_3 A_{n3}^T, \\ \quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \widetilde{\mathcal{L}}_n(\mathcal{Y}_1, \cdots, \mathcal{Y}_n) = \mathcal{Y}_1 \times_n A_{1n}^T + \mathcal{Y}_2 \times_{n-1} A_{2(n-1)}^T + \cdots + \mathcal{Y}_n \times_1 A_{n1}^T. \end{cases} \tag{8}$$

Proof. For Eq.(6), it follows that

$$\begin{aligned} &\langle \mathcal{Y}_1, \mathcal{X}_1 \times_1 A_{11} + \mathcal{X}_2 \times_2 A_{12} + \cdots + \mathcal{X}_n \times_n A_{1n}\rangle + \langle \mathcal{Y}_2, \mathcal{X}_2 \times_1 A_{21} + \mathcal{X}_3 \times_2 A_{22} \cdots + \mathcal{X}_1 \times_n A_{2n}\rangle \\ &+ \cdots + \langle \mathcal{Y}_n, \mathcal{X}_n \times_1 A_{n1} + \mathcal{X}_1 \times_2 A_{n2} + \cdots + \mathcal{X}_{n-1} \times_n A_{nn}\rangle \\ &= [\langle \mathcal{Y}_1 \times_1 A_{11}^T, \mathcal{X}_1\rangle + \langle \mathcal{Y}_1 \times_2 A_{12}^T, \mathcal{X}_2\rangle + \cdots + \langle \mathcal{Y}_1 \times_n A_{1n}^T, \mathcal{X}_n\rangle] + [\langle \mathcal{Y}_2 \times_1 A_{21}^T, \mathcal{X}_2\rangle + \langle \mathcal{Y}_2 \times_2 A_{22}^T, \mathcal{X}_3\rangle \\ &+ \cdots + \langle \mathcal{Y}_2 \times_n A_{2n}^T, \mathcal{X}_1\rangle] + [\langle \mathcal{Y}_n \times_1 A_{n1}^T, \mathcal{X}_n\rangle + \langle \mathcal{Y}_n \times_2 A_{n2}^T, \mathcal{X}_1\rangle + \cdots + \langle \mathcal{Y}_n \times_n A_{nn}^T, \mathcal{X}_{n-1}\rangle] \\ &= \langle \mathcal{X}_1, \mathcal{Y}_1 \times_1 A_{11}^T + \mathcal{Y}_2 \times_n A_{2n}^T + \cdots + \mathcal{Y}_n \times_2 A_{n2}^T\rangle + \langle \mathcal{X}_2, \mathcal{Y}_1 \times_2 A_{12}^T + \mathcal{Y}_2 \times_1 A_{21}^T \cdots + \mathcal{Y}_n \times_3 A_{n3}^T\rangle \\ &+ \langle \mathcal{X}_n, \mathcal{Y}_n \times_1 A_{n1}^T + \mathcal{Y}_2 \times_{n-1} A_{2(n-1)}^T + \cdots + \mathcal{Y}_1 \times_n A_{1n}^T\rangle \\ &= \langle \mathcal{X}_1, \widetilde{\mathcal{L}}_1(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_n)\rangle + \langle \mathcal{X}_2, \widetilde{\mathcal{L}}_2(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_n)\rangle + \cdots + \langle \mathcal{X}_n, \widetilde{\mathcal{L}}_n(\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_n)\rangle. \end{aligned}$$

□

Next, we develop the BiCOR method in its tensor form for studying the GCSTEs (1) as follows.

Algorithm 2.1. Tensor form of BiCOR method for solving GCSTEs (1).

1. Input initial guesses $\mathcal{X}_i^0$, $\mathcal{B}_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, $A_{ij} \in \mathbb{R}^{I_j \times I_j}$, $k = 0$, and set $\epsilon > 0$.

2. Calculate

$$\begin{cases}\mathcal{R}_1^0 = \mathcal{B}_1 - \mathcal{L}_1(\mathcal{X}_1^0, \cdots, \mathcal{X}_n^0), \\ \mathcal{R}_2^0 = \mathcal{B}_2 - \mathcal{L}_2(\mathcal{X}_1^0, \cdots, \mathcal{X}_n^0), \\ \quad\vdots \qquad\qquad \vdots \\ \mathcal{R}_n^0 = \mathcal{B}_n - \mathcal{L}_n(\mathcal{X}_1^0, \cdots, \mathcal{X}_n^0), \end{cases} \quad \begin{cases}\mathcal{P}_1^0 = \mathcal{R}_1^0, \\ \mathcal{P}_2^0 = \mathcal{R}_2^0, \\ \quad\vdots \\ \mathcal{P}_n^0 = \mathcal{R}_n^0, \end{cases}$$

$$\begin{cases}\widetilde{\mathcal{R}}_1^0 = \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0), \\ \widetilde{\mathcal{R}}_2^0 = \mathcal{L}_2(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0), \\ \quad\vdots \qquad\qquad \vdots \\ \widetilde{\mathcal{R}}_n^0 = \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0) \end{cases} \quad \begin{cases}\widetilde{\mathcal{P}}_1^0 = \widetilde{\mathcal{R}}_1^0, \\ \widetilde{\mathcal{P}}_2^0 = \widetilde{\mathcal{R}}_2^0, \\ \quad\vdots \\ \widetilde{\mathcal{P}}_n^0 = \widetilde{\mathcal{R}}_n^0, \end{cases}$$

and

$$\begin{aligned}\rho_0 &= \langle \widetilde{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \langle \widetilde{\mathcal{R}}_2^0, \mathcal{L}_2(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle, \\ \sigma_0 &= \langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_1(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle + \langle \widetilde{\mathcal{L}}_2(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_2(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle \\ &\quad + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_n(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle, \\ \eta_k &= \|\mathcal{R}_1^k\| + \|\mathcal{R}_2^k\| + \cdots + \|\mathcal{R}_n^k\|.\end{aligned}$$

3. For $k = 1, 2, \cdots$ until $\eta_k < \epsilon$ do:
$\alpha_k = \frac{\rho_k}{\sigma_k}$,

$$\begin{cases}\mathcal{X}_1^{k+1} = \mathcal{X}_1^k + \alpha_k \mathcal{P}_1^k, \\ \mathcal{X}_2^{k+1} = \mathcal{X}_2^k + \alpha_k \mathcal{P}_2^k, \\ \quad\vdots \qquad \vdots \\ \mathcal{X}_n^{k+1} = \mathcal{X}_n^k + \alpha_k \mathcal{P}_n^k, \end{cases}$$

$$\begin{cases}\mathcal{R}_1^{k+1} = \mathcal{R}_1^k - \alpha_k \mathcal{L}_1(\mathcal{P}_1^k, \cdots, \mathcal{P}_n^k), \\ \mathcal{R}_2^{k+1} = \mathcal{R}_2^k - \alpha_k \mathcal{L}_2(\mathcal{P}_1^k, \cdots, \mathcal{P}_n^k), \\ \quad\vdots \qquad\qquad \vdots \\ \mathcal{R}_n^{k+1} = \mathcal{R}_n^k - \alpha_k \mathcal{L}_n(\mathcal{P}_1^k, \cdots, \mathcal{P}_n^k), \end{cases} \quad \begin{cases}\widetilde{\mathcal{R}}_1^{k+1} = \widetilde{\mathcal{R}}_1^k - \alpha_k \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^k, \cdots, \widetilde{\mathcal{P}}_n^k), \\ \widetilde{\mathcal{R}}_2^{k+1} = \widetilde{\mathcal{R}}_2^k - \alpha_k \widetilde{\mathcal{L}}_2(\widetilde{\mathcal{P}}_1^k, \cdots, \widetilde{\mathcal{P}}_n^k), \\ \quad\vdots \qquad\qquad \vdots \\ \widetilde{\mathcal{R}}_n^{k+1} = \widetilde{\mathcal{R}}_n^k - \alpha_k \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^k, \cdots, \widetilde{\mathcal{P}}_n^k), \end{cases}$$

$$\begin{cases}\mathcal{L}_1(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_1 A_{11} + \mathcal{R}_2^{k+1} \times_2 A_{12} + \cdots + \mathcal{R}_n^{k+1} \times_n A_{1n}, \\ \mathcal{L}_2(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) = \mathcal{R}_2^{k+1} \times_1 A_{21} + \mathcal{R}_3^{k+1} \times_2 A_{22} \cdots + \mathcal{R}_1^{k+1} \times_n A_{2n}, \\ \quad\vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ \mathcal{L}_n(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) = \mathcal{R}_n^{k+1} \times_1 A_{n1} + \mathcal{R}_1^{k+1} \times_2 A_{n2} + \cdots + \mathcal{R}_{n-1}^{k+1} \times_n A_{nn}, \end{cases}$$

$$\rho_{k+1} = \langle \widetilde{\mathcal{R}}_1^{k+1}, \mathcal{L}_1(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1})\rangle + \langle \widetilde{\mathcal{R}}_2^{k+1}, \mathcal{L}_2(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1})\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^{k+1}, \mathcal{L}_n(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1})\rangle.$$

4. Calculate $\beta_k = \frac{\rho_{k+1}}{\rho_k}$,

$$\begin{cases}\mathcal{P}_1^{k+1} = \mathcal{R}_1^{k+1} + \beta_k \mathcal{P}_1^k, \\ \mathcal{P}_2^{k+1} = \mathcal{R}_1^{k+1} + \beta_k \mathcal{P}_2^k, \\ \quad\vdots \qquad\qquad \vdots \\ \mathcal{P}_n^{k+1} = \mathcal{R}_1^{k+1} + \beta_k \mathcal{P}_n^k, \end{cases} \quad \begin{cases}\widetilde{\mathcal{P}}_1^{k+1} = \widetilde{\mathcal{R}}_1^{k+1} + \beta_k \widetilde{\mathcal{P}}_1^k, \\ \widetilde{\mathcal{P}}_2^{k+1} = \widetilde{\mathcal{R}}_1^{k+1} + \beta_k \widetilde{\mathcal{P}}_2^k, \\ \quad\vdots \qquad\qquad \vdots \\ \widetilde{\mathcal{P}}_n^{k+1} = \widetilde{\mathcal{R}}_1^{k+1} + \beta_k \widetilde{\mathcal{P}}_n^k, \end{cases}$$

$$\sigma_{k+1} = \langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^{k+1}, \cdots, \widetilde{\mathcal{P}}_n^{k+1}), \mathcal{L}_1(\mathcal{P}_1^{k+1}, \cdots, \mathcal{P}_n^{k+1})\rangle + \langle \widetilde{\mathcal{L}}_2(\widetilde{\mathcal{P}}_1^{k+1}, \cdots, \widetilde{\mathcal{P}}_n^{k+1}), \mathcal{L}_2(\mathcal{P}_1^{k+1}, \cdots, \mathcal{P}_n^{k+1})\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^{k+1}, \cdots, \widetilde{\mathcal{P}}_n^{k+1}), \mathcal{L}_n(\mathcal{P}_1^{k+1}, \cdots, \mathcal{P}_n^{k+1})\rangle.$$

Theorem 2.1. If the tensor sequences $\{\widetilde{\mathcal{R}}_m^k\}$, $\{\mathcal{L}_m(\mathcal{R}_1^k, \cdots, \mathcal{R}_n^k)\}$, $\{\widetilde{\mathcal{L}}_m(\widetilde{\mathcal{P}}_1^k, \cdots, \widetilde{\mathcal{P}}_n^k)\}$ and $\{\mathcal{L}_m(\mathcal{P}_1^k, \cdots, \mathcal{P}_n^k)\}$ ($m = 1, 2, \cdots, n$, $k = 0, 1, \cdots$) are generated by Algorithm 2.1, then

$$\begin{aligned} &\langle \widetilde{\mathcal{R}}_1^i, \mathcal{L}_1(\mathcal{R}_1^j, \cdots, \mathcal{R}_n^j)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^i, \mathcal{L}_n(\mathcal{R}_1^j, \cdots, \mathcal{R}_n^j)\rangle \\ &= \langle \mathcal{R}_1^j, \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^i, \cdots, \widetilde{\mathcal{R}}_n^i)\rangle + \cdots + \langle \mathcal{R}_n^j, \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^i, \cdots, \widetilde{\mathcal{R}}_n^i)\rangle = 0, \end{aligned} \tag{9}$$

$$\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^i, \cdots, \widetilde{\mathcal{P}}_n^i), \mathcal{L}_1(\mathcal{P}_1^j, \cdots, \mathcal{P}_n^j)\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^i, \cdots, \widetilde{\mathcal{P}}_n^i), \mathcal{L}_n(\mathcal{P}_1^j, \cdots, \mathcal{P}_n^j)\rangle = 0 \tag{10}$$

hold for $i \neq j, i, j = 0, 1, \cdots$. Moreover, for $i > j$, it follows that

$$\langle \widetilde{\mathcal{R}}_1^i, \mathcal{L}_1(\mathcal{P}_1^j, \cdots, \mathcal{P}_n^j)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^i, \mathcal{L}_n(\mathcal{P}_n^j, \cdots, \mathcal{P}_n^j)\rangle = 0. \tag{11}$$

Proof. From Lemma 2.1, we can see that the first half part of Eq.(9) is straightforward. For the rest assertions, we first prove that Eqs.(9), (10) and (11) hold for $0 \leq j < i \leq k$ by induction. If $k = 1$, it follows that

$$\begin{aligned} &\langle \widetilde{\mathcal{R}}_1^1, \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^1, \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle \\ =&\langle \widetilde{\mathcal{R}}_1^0 - \alpha_0 \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^0 - \alpha_0 \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle \\ =&\langle \widetilde{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle \\ &- \alpha_0[\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle] \\ =&\langle \widetilde{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle \\ &- \frac{\langle \widetilde{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle}{\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_1(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_n(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle} \\ &[\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_1(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{L}_n(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle] \\ =&0. \end{aligned}$$

Since $\mathcal{P}_1^0 = \mathcal{R}_1^0$, by Lemma 2.1, it follows that

$$\begin{aligned} \langle \widetilde{\mathcal{R}}_1^1, \mathcal{L}_1(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^1, \mathcal{L}_n(\mathcal{P}_n^0, \cdots, \mathcal{P}_n^0)\rangle =&\langle \widetilde{\mathcal{R}}_1^1, \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^1, \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle \\ =&0, \end{aligned}$$

and

$$\begin{aligned} &\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^1, \cdots, \widetilde{\mathcal{P}}_n^1), \mathcal{L}_1(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^1, \cdots, \widetilde{\mathcal{P}}_n^1), \mathcal{L}_n(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0)\rangle \\ =&\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^1, \cdots, \widetilde{\mathcal{R}}_n^1) + \beta_0 \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \frac{1}{\alpha_0}(\mathcal{R}_1^0 - \mathcal{R}_1^1)\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^1, \cdots, \widetilde{\mathcal{R}}_n^1) + \beta_0 \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \frac{1}{\alpha_0}(\mathcal{R}_n^0 - \mathcal{R}_n^1)\rangle \\ =&-\frac{1}{\alpha_0}[\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^1, \cdots, \widetilde{\mathcal{R}}_n^1), \mathcal{R}_1^1\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^1, \cdots, \widetilde{\mathcal{R}}_n^1), \mathcal{R}_n^1\rangle - \beta_0[\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{R}_1^0\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^0, \cdots, \widetilde{\mathcal{P}}_n^0), \mathcal{R}_n^0\rangle]] \\ =&-\frac{1}{\alpha_0}[\langle \widetilde{\mathcal{R}}_1^1, \mathcal{L}_1(\mathcal{R}_1^1, \cdots, \mathcal{R}_n^1)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^1, \mathcal{L}_n(\mathcal{R}_1^1, \cdots, \mathcal{R}_n^1)\rangle \\ &-\frac{\langle \widetilde{\mathcal{R}}_1^1, \mathcal{L}_1(\mathcal{R}_1^1, \cdots, \mathcal{R}_n^1)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^1, \mathcal{L}_n(\mathcal{R}_1^1, \cdots, \mathcal{R}_n^1)\rangle}{\langle \widetilde{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0)\rangle}[\langle \widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^0, \cdots, \widetilde{\mathcal{R}}_n^0), \mathcal{R}_1^0\rangle + \cdots + \langle \widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^0, \cdots, \widetilde{\mathcal{R}}_n^0), \mathcal{R}_n^0\rangle]] \\ =&-\frac{1}{\alpha_0}[\langle \widetilde{\mathcal{R}}_1^1, \mathcal{L}_1(\mathcal{R}_1^1, \cdots, \mathcal{R}_n^1)\rangle + \cdots + \langle \widetilde{\mathcal{R}}_n^1, \mathcal{L}_n(\mathcal{R}_1^1, \cdots, \mathcal{R}_n^1)\rangle \end{aligned}$$

$$-\frac{\langle\widetilde{\mathcal{R}}_1^1,\mathcal{L}_1(\mathcal{R}_1^1,\cdots,\mathcal{R}_1^1)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^1,\mathcal{L}_n(\mathcal{R}_1^1,\cdots,\mathcal{R}_n^1)\rangle}{\langle\widetilde{\mathcal{R}}_1^0,\mathcal{L}_1(\mathcal{R}_1^0,\cdots,\mathcal{R}_n^0)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^0,\mathcal{L}_n(\mathcal{R}_1^0,\cdots,\mathcal{R}_n^0)\rangle}[\langle\widetilde{\mathcal{R}}_1^0,\mathcal{L}_1(\mathcal{R}_1^0,\cdots,\mathcal{R}_n^0)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^0,\mathcal{L}_n(\mathcal{R}_1^0,\cdots,\mathcal{R}_n^0)\rangle]]$$
$$=0.$$

Assume that the assertions hold for $k=m$, then for $k=m+1$, it follows that

$$\begin{aligned}
&\langle\widetilde{\mathcal{R}}_1^{m+1},\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^{m+1},\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m-\alpha_m\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m-\alpha_m\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle\\
&-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle]\\
=&\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\\
&-\beta_{m-1}\mathcal{L}_1(\mathcal{P}_1^{m-1},\cdots,\mathcal{P}_n^{m-1})\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)-\beta_{m-1}\mathcal{L}_n(\mathcal{P}_1^{m-1},\cdots,\mathcal{P}_n^{m-1})\rangle]\\
=&\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle\\
&-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle]\\
=&0,
\end{aligned}$$

and

$$\begin{aligned}
&\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^{m+1},\cdots,\widetilde{\mathcal{P}}_n^{m+1}),\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^{m+1},\cdots,\widetilde{\mathcal{P}}_n^{m+1}),\mathcal{L}_n(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle\\
=&\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1})+\beta_m\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\frac{1}{\alpha_m}(\mathcal{R}_1^m-\mathcal{R}_1^{m+1})\rangle\\
&+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1})+\beta_m\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\frac{1}{\alpha_m}(\mathcal{R}_n^m-\mathcal{R}_n^{m+1})\rangle\\
=&-\frac{1}{\alpha_m}[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\mathcal{R}_1^{m+1}\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\mathcal{R}_n^{m+1}\rangle\\
&-\beta_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{R}_1^m\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{R}_n^m\rangle]]\\
=&-\frac{1}{\alpha_m}[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\mathcal{R}_1^{m+1}\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\mathcal{R}_n^{m+1}\rangle\\
&-\beta_m[\langle\widetilde{\mathcal{P}}_1^m,\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{P}}_n^m,\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle]]\\
=&-\frac{1}{\alpha_m}[\langle\widetilde{\mathcal{R}}_1^{m+1},\mathcal{L}_1(\mathcal{R}_1^{m+1},\cdots,\mathcal{R}_n^{m+1})\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^{m+1},\mathcal{L}_n(\mathcal{R}_1^{m+1},\cdots,\mathcal{R}_n^{m+1})\rangle\\
&-\frac{\langle\widetilde{\mathcal{R}}_1^{m+1},\mathcal{L}_1(\mathcal{R}_1^{m+1},\cdots,\mathcal{R}_n^{m+1})\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^{m+1},\mathcal{L}_n(\mathcal{R}_1^{m+1},\cdots,\mathcal{R}_n^{m+1})\rangle}{\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle}\\
&[\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle]]\\
=&0.
\end{aligned}$$

Moreover, it follows that

$$\begin{aligned}
&\langle\widetilde{\mathcal{R}}_1^{m+1},\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^{m+1},\mathcal{L}_n(\mathcal{P}_n^m,\cdots,\mathcal{P}_n^m)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m-\alpha_m\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m-\alpha_m\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_n^m,\cdots,\mathcal{P}_n^m)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle\\
&-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_n^m,\cdots,\mathcal{P}_n^m)\rangle]
\end{aligned}$$

$$
\begin{aligned}
&=\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{R}_1^m,\cdots,\mathcal{R}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{R}_n^m,\cdots,\mathcal{R}_n^m)\rangle\\
&\quad+\beta_{m-1}\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{P}_1^{m-1},\cdots,\mathcal{P}_n^{m-1})\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{P}_n^{m-1},\cdots,\mathcal{P}_n^{m-1})\rangle\\
&\quad-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^m,\cdots,\mathcal{P}_n^m)\rangle]\\
&=0.
\end{aligned}
$$

For $0\le j<m$, we have

$$
\begin{aligned}
&\langle\widetilde{\mathcal{R}}_1^{m+1},\mathcal{L}_1(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^{m+1},\mathcal{L}_n(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m-\alpha_m\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m-\alpha_m\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle\\
&-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{R}_1^j,\cdots,\mathcal{R}_n^j)\rangle]\\
=&-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle\\
&-\beta_{j-1}[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^{j-1},\cdots,\mathcal{P}_n^{j-1})\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^{j-1},\cdots,\mathcal{P}_n^{j-1})\rangle]]\\
=&0-0\\
=&0,
\end{aligned}
$$

and

$$
\begin{aligned}
&\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^{m+1},\cdots,\widetilde{\mathcal{P}}_n^{m+1}),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^{m+1},\cdots,\widetilde{\mathcal{P}}_n^{m+1}),\mathcal{L}_n(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle\\
=&\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1})+\beta_m\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle\\
&+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1})+\beta_m\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle\\
=&\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle\\
&+\beta_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle]\\
=&\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\frac{1}{\alpha_j}(\mathcal{R}_1^j-\mathcal{R}_1^{j+1})\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^{m+1},\cdots,\widetilde{\mathcal{R}}_n^{m+1}),\frac{1}{\alpha_j}(\mathcal{R}_n^j-\mathcal{R}_n^{j+1})\rangle\\
=&\frac{1}{\alpha_j}[\langle\widetilde{\mathcal{R}}_1^{m+1},\mathcal{L}_1(\mathcal{R}_1^{j+1},\cdots,\mathcal{R}_n^{j+1})\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^{m+1},\mathcal{L}_n(\mathcal{R}_1^{j+1},\cdots,\mathcal{R}_n^{j+1})\rangle]\\
=&0.
\end{aligned}
$$

Moreover, it follows that

$$
\begin{aligned}
&\langle\widetilde{\mathcal{R}}_1^{m+1},\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^{m+1},\mathcal{L}_n(\mathcal{P}_n^j,\cdots,\mathcal{P}_n^j)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m-\alpha_m\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m-\alpha_m\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_n^j,\cdots,\mathcal{P}_n^j)\rangle\\
=&\langle\widetilde{\mathcal{R}}_1^m,\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{R}}_n^m,\mathcal{L}_n(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle\\
&-\alpha_m[\langle\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_1(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle+\cdots+\langle\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{P}}_1^m,\cdots,\widetilde{\mathcal{P}}_n^m),\mathcal{L}_n(\mathcal{P}_1^j,\cdots,\mathcal{P}_n^j)\rangle]\\
=&0-0\\
=&0.
\end{aligned}
$$

According to the above discussion, one can see that Eqs.(9),(10) and (11) hold for $0\le j<i\le k$. On the other hand, for the case $0\le i<j\le k$, we can similarly prove them and consequently complete the proof. □

Based on the above observations, we present the following theorem to show that Algorithm 2.1 converges to the solutions at most finite steps for any initial values without round-off errors.

Theorem 2.2. Let $\{\mathcal{X}_l^k\}(l=1,\cdots,n)$ be the iterative sequences generated by Algorithm 2.1. If Algorithm 2.1 does not break down by zero division, then $(\mathcal{X}_1^k,\mathcal{X}_2^k,\cdots,\mathcal{X}_n^k)$ converges to an exact solution group

$(\mathcal{X}_1^*, \mathcal{X}_2^*, \cdots, \mathcal{X}_n^*)$ within at most $nI_1I_2\cdots I_n + 1$ iteration steps for any initial values in the absence round-off errors.

Proof. Set $m = nI_1I_2\cdots I_n$, if $\mathcal{R}_l^k = \mathcal{O}, (l = 1, \cdots, n; k < m)$, then $(\mathcal{X}_1^k, \mathcal{X}_2^k, \cdots, \mathcal{X}_n^k)$ is a solution group of Eqs.(1). If $\mathcal{R}_l^k \neq \mathcal{O}(l = 1, \cdots, n; k = 1, \cdots, m)$, by Algorithm 2.1 does not break down, then we can compute $\mathcal{R}_l^{m+1}$. Let

$$S^i = \begin{bmatrix} \mathrm{vec}(\mathcal{R}_1^i) & & & \\ & \mathrm{vec}(\mathcal{R}_2^i) & & \\ & & \ddots & \\ & & & \mathrm{vec}(\mathcal{R}_n^i) \end{bmatrix},$$

$$W^i = \begin{bmatrix} \mathrm{vec}(\widetilde{\mathcal{L}}_1(\widetilde{\mathcal{R}}_1^i, \cdots, \widetilde{\mathcal{R}}_n^i)) & & & \\ & \mathrm{vec}(\widetilde{\mathcal{L}}_2(\widetilde{\mathcal{R}}_1^i, \cdots, \widetilde{\mathcal{R}}_n^i)) & & \\ & & \ddots & \\ & & & \mathrm{vec}(\widetilde{\mathcal{L}}_n(\widetilde{\mathcal{R}}_1^i, \cdots, \widetilde{\mathcal{R}}_n^i)) \end{bmatrix},$$

where $\mathrm{vec}(\mathcal{R}_l^i)$, $\mathrm{vec}(\widetilde{\mathcal{L}}_2(\widetilde{\mathcal{R}}_1^i, \cdots, \widetilde{\mathcal{R}}_n^i)) \in \mathbb{R}^{I_1I_2\cdots I_n}(i = 1, \cdots, m; l = 1, \cdots, n)$ denote the vectors consisting of themselves mode-1 fiber. Suppose that $S^1, S^2, \cdots, S^m$ are linearly dependent. There exist scalars $\lambda_1, \lambda_2, \cdots, \lambda_{nI_1I_2\cdots I_n}$, not all zero, such that

$$\lambda_1 S^1 + \lambda_2 S^2 + \cdots + \lambda_m S^m = O,$$

where O represents an m-by-m identity matrix. Taking the inner product with W_i on both sides, by Theorem 2.1, it follows that

$$\begin{aligned} 0 = \langle W^i, O\rangle &= \langle W^i, \lambda_1 S^1 + \lambda_2 S^2 + \cdots + \lambda_m S^m\rangle \\ &= \lambda_i \langle W^i, S^i\rangle, \quad i = 1, \cdots, m. \end{aligned}$$

Since $\mathcal{R}_l^{m+1}$ can be computed, one can see that $\langle W^i, S^i\rangle \neq 0$ resulting in $\lambda_i = 0$, which contradicts to the assumption. Therefore, we can obtain that $S^1, S^2, \cdots, S^m$ is a basis of the subspace of

$$H = \left\{ G | G = \begin{bmatrix} G_1 & & & \\ & G_2 & & \\ & & \ddots & \\ & & & G_n \end{bmatrix}, \text{where } G_l \in \mathbb{R}^{I_1I_2\cdots I_n}, l = 1, 2, \cdots, n. \right\}.$$

It follows that $S^{m+1} = 0$, i.e., $\mathcal{R}_l^{m+1} = \mathcal{O}$. This fact shows that $(\mathcal{X}_1^{I_1I_2\cdots I_n+1}, \mathcal{X}_2^{I_1I_2\cdots I_n+1}, \cdots, \mathcal{X}_n^{I_1I_2\cdots I_n+1})$ is an exact solution group of Eqs.(1), and consequently completes the proof. □

Similar to the ingenious derivation of the conjugate gradient method, we extend a transpose-free variant of Algorithm 2.1, i.e., the conjugate A-orthogonal residual squared method based on tensor format, for solving Eqs.(1).

Algorithm 2.2. Tensor form of CORS method for solving GCSTEs (1).

1. Input initial guesses $\mathcal{X}_i^0$, $\mathcal{B}_i \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_n}$, $A_{ij} \in \mathbb{R}^{I_j\times I_j}$, $k = 0$, and set $\epsilon > 0$.
2. Calculate

$$\begin{cases} \mathcal{R}_1^0 = \mathcal{B}_1 - \mathcal{L}_1(\mathcal{X}_1^0, \cdots, \mathcal{X}_n^0), \\ \mathcal{R}_2^0 = \mathcal{B}_2 - \mathcal{L}_2(\mathcal{X}_1^0, \cdots, \mathcal{X}_n^0), \\ \quad\vdots \qquad\qquad \vdots \\ \mathcal{R}_n^0 = \mathcal{B}_n - \mathcal{L}_n(\mathcal{X}_1^0, \cdots, \mathcal{X}_n^0), \end{cases} \quad \begin{cases} \mathcal{E}_1^0 = \mathcal{R}_1^0, \\ \mathcal{E}_2^0 = \mathcal{R}_2^0, \\ \quad\vdots \\ \mathcal{E}_n^0 = \mathcal{R}_n^0, \end{cases}$$

$$\begin{cases} \overline{\mathcal{R}}_1^0 = \widehat{\mathcal{R}}_1^0 = \mathcal{L}_1(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0), \\ \overline{\mathcal{R}}_2^0 = \widehat{\mathcal{R}}_2^0 = \mathcal{L}_2(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0), \\ \quad \vdots \qquad\qquad \vdots \\ \overline{\mathcal{R}}_n^0 = \widehat{\mathcal{R}}_n^0 = \mathcal{L}_n(\mathcal{R}_1^0, \cdots, \mathcal{R}_n^0) \end{cases} \quad \begin{cases} \mathcal{P}_1^0 = \mathcal{D}_1^0 = \overline{\mathcal{R}}_1^0, \\ \mathcal{P}_2^0 = \mathcal{D}_2^0 = \overline{\mathcal{R}}_2^0, \\ \quad \vdots \\ \mathcal{P}_n^0 = \mathcal{D}_n^0 = \overline{\mathcal{R}}_n^0, \end{cases}$$

and

$$\begin{aligned} \rho_0 &= \langle \overline{\mathcal{R}}_1^0, \widehat{\mathcal{R}}_1^0 \rangle + \langle \overline{\mathcal{R}}_2^0, \widehat{\mathcal{R}}_2^0 \rangle + \cdots + \langle \overline{\mathcal{R}}_n^0, \widehat{\mathcal{R}}_n^0 \rangle, \\ \sigma_0 &= \langle \overline{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0) \rangle + \langle \overline{\mathcal{R}}_2^0, \mathcal{L}_2(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0) \rangle + \cdots + \langle \overline{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{P}_1^0, \cdots, \mathcal{P}_n^0) \rangle, \\ \eta_k &= \|\mathcal{R}_1^k\| + \|\mathcal{R}_2^k\| + \cdots + \|\mathcal{R}_n^k\|. \end{aligned}$$

3. For $k = 0, 1, 2, \cdots$ until $\eta_k < \epsilon$ do:
$\alpha_k = \frac{\rho_k}{\sigma_k}$,

$$\begin{cases} \mathcal{H}_1^k = \mathcal{E}_1^k - \alpha_k \mathcal{P}_1^k, \\ \mathcal{H}_2^k = \mathcal{E}_2^k - \alpha_k \mathcal{P}_2^k, \\ \quad \vdots \quad \vdots \\ \mathcal{H}_n^k = \mathcal{E}_n^k - \alpha_k \mathcal{P}_n^k, \end{cases} \quad \begin{cases} \mathcal{X}_1^{k+1} = \mathcal{X}_1^k + \alpha_k(\mathcal{E}_1^k + \mathcal{H}_1^k), \\ \mathcal{X}_2^{k+1} = \mathcal{X}_2^k + \alpha_k(\mathcal{E}_2^k + \mathcal{H}_2^k), \\ \quad \vdots \quad \vdots \\ \mathcal{X}_n^{k+1} = \mathcal{X}_n^k + \alpha_k(\mathcal{E}_n^k + \mathcal{H}_n^k), \end{cases}$$

$$\begin{cases} \mathcal{F}_1^k = \mathcal{D}_1^k - \alpha_k \mathcal{L}_1(\mathcal{P}_1^k, \cdots, \mathcal{P}_n^k), \\ \mathcal{F}_2^k = \mathcal{D}_2^k - \alpha_k \mathcal{L}_2(\mathcal{P}_1^k, \cdots, \mathcal{P}_n^k), \\ \quad \vdots \qquad\qquad \vdots \\ \mathcal{F}_n^k = \mathcal{D}_n^k - \alpha_k \mathcal{L}_n(\mathcal{P}_1^k, \cdots, \mathcal{P}_n^k), \end{cases} \quad \begin{cases} \mathcal{R}_1^{k+1} = \mathcal{R}_1^k - \alpha_k(\mathcal{D}_1^k + \mathcal{F}_1^k), \\ \mathcal{R}_2^{k+1} = \mathcal{R}_2^k - \alpha_k(\mathcal{D}_2^k + \mathcal{F}_2^k), \\ \quad \vdots \qquad\qquad \vdots \\ \mathcal{R}_n^{k+1} = \mathcal{R}_n^k - \alpha_k(\mathcal{D}_n^k + \mathcal{F}_n^k), \end{cases}$$

$$\begin{cases} \mathcal{L}_1(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_1 A_{11} + \mathcal{R}_2^{k+1} \times_2 A_{12} + \cdots + \mathcal{R}_n^{k+1} \times_n A_{1n}, \\ \mathcal{L}_2(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) = \mathcal{R}_2^{k+1} \times_1 A_{21} + \mathcal{R}_3^{k+1} \times_2 A_{22} \cdots + \mathcal{R}_1^{k+1} \times_n A_{2n}, \\ \quad \vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ \mathcal{L}_n(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) = \mathcal{R}_n^{k+1} \times_1 A_{n1} + \mathcal{R}_1^{k+1} \times_2 A_{n2} + \cdots + \mathcal{R}_{n-1}^{k+1} \times_n A_{nn}, \end{cases}$$

$$\rho_{k+1} = \langle \overline{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) \rangle + \langle \overline{\mathcal{R}}_2^0, \mathcal{L}_2(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) \rangle + \cdots + \langle \overline{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) \rangle.$$

4. Calculate $\beta_k = \frac{\rho_{k+1}}{\rho_k}$,

$$\begin{cases} \mathcal{E}_1^{k+1} = \mathcal{R}_1^{k+1} + \beta_k \mathcal{H}_1^k, \\ \mathcal{E}_2^{k+1} = \mathcal{R}_2^{k+1} + \beta_k \mathcal{H}_2^k, \\ \quad \vdots \quad \vdots \\ \mathcal{E}_n^{k+1} = \mathcal{R}_n^{k+1} + \beta_k \mathcal{H}_n^k, \end{cases} \quad \begin{cases} \mathcal{D}_1^{k+1} = \mathcal{L}_1(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{F}_1^k, \\ \mathcal{D}_2^{k+1} = \mathcal{L}_2(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{F}_2^k, \\ \quad \vdots \qquad\qquad \vdots \\ \mathcal{D}_n^{k+1} = \mathcal{L}_n(\mathcal{R}_1^{k+1}, \cdots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{F}_n^k, \end{cases}$$

$$\begin{cases} \mathcal{P}_1^{k+1} = \mathcal{D}_1^{k+1} + \beta_k(\mathcal{F}_1^k + \beta_k \mathcal{P}_1^k), \\ \mathcal{P}_2^{k+1} = \mathcal{D}_2^{k+1} + \beta_k(\mathcal{F}_2^k + \beta_k \mathcal{P}_2^k), \\ \quad \vdots \qquad\qquad \vdots \\ \mathcal{P}_n^{k+1} = \mathcal{D}_n^{k+1} + \beta_k(\mathcal{F}_n^k + \beta_k \mathcal{P}_n^k), \end{cases}$$

$$\sigma_{k+1} = \langle \overline{\mathcal{R}}_1^0, \mathcal{L}_1(\mathcal{P}_1^{k+1}, \cdots, \mathcal{P}_n^{k+1}) \rangle + \langle \overline{\mathcal{R}}_2^0, \mathcal{L}_2(\mathcal{P}_1^{k+1}, \cdots, \mathcal{P}_n^{k+1}) \rangle + \cdots + \langle \overline{\mathcal{R}}_n^0, \mathcal{L}_n(\mathcal{P}_1^{k+1}, \cdots, \mathcal{P}_n^{k+1}) \rangle.$$

Table 1: Comparison results of Example 3.1.

Algorithms	**MCG** [35]			**Algorithm 2.1**			**Algorithm 2.2**		
$[m, m, m, m]$	IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
$[10, 10, 10, 10]$	171	1.5633	9.9249e-08	54	0.6256	7.1313e-08	33	0.3329	2.3007e-08
$[20, 20, 20, 20]$	648	13.9178	9.3671e-08	107	3.3791	8.4555e-08	67	1.7091	2.5300e-08
$[30, 30, 30, 30]$	1434	99.2302	9.7902e-08	157	16.1341	8.5114e-08	98	7.9482	4.5514e-08
$[40, 40, 40, 40]$	†	†	†	215	64.6975	7.9150e-08	131	33.2806	4.4276e-08

3. Numerical experiments

In this section, we provide some numerical experiments to illustrate the efficiency of the proposed algorithms here. All codes were written via the tensor toolbox (version 3.2.1) in Matlab 2020b. The numerical experiments were done with an XPS 7590 with Inter(R) Core(TM) i7-9750H @2.6 GHz and 8 GB of RAM. To show the superiority of proposed algorithms, we evaluate and compare Algorithms 2.1 and 2.2 against MCG [35] in terms of iterations and computing time. For clarity, we denote the computing time in seconds, the iteration steps, and the residual norm η_k in Algorithm 2.1 as "CPU", "IT", and "RES", respectively. The stopping criteria is RES $< \epsilon := 10^{-7}$, or the iteration steps exceeds $k_{max} := 3000$ with k_{max} being the maximum iterations. We denote "†" as the residual that does not yield to the tolerance after reaching k_{max}.

Example 3.1. Consider a 4th-order Sylvester tensor equation Eq.(2) corresponding to Eq.(3), and set its parameters as Eq.(4) with $v = 3$, $c_i = i (i = 1, 2, 3, 4)$ and $\mathcal{B} = tenrand(m, m, m, m)$.

We performed the proposed algorithms and MCG for solving Eq.(2) and recorded the comparison results in Table 1. Observe from this table that our algorithms outperform MCG both in iterations and computing time. On the other hand, we can see that the convergence rate of Algorithm 2.2 is about twice Algorithm 2.1, and thus the algorithm is the fastest solver concerning CPU time. We depicted the convergence cures of all algorithms for the case $m = 10$ in Fig.1. It shows that the number of iterations corresponding to MCG is prohibitively more than our algorithms to achieve the required accuracy. This fact implies that our algorithms are more competitive.

Example 3.2. Consider the system of Sylvester tensor equations

$$\begin{cases} \mathcal{X}_1 \times_1 A_{11} + \mathcal{X}_2 \times_2 A_{12} + \mathcal{X}_3 \times_3 A_{13} = \mathcal{B}_1, \\ \mathcal{X}_2 \times_1 A_{21} + \mathcal{X}_3 \times_2 A_{22} + \mathcal{X}_1 \times_1 A_{23} = \mathcal{B}_2, \\ \mathcal{X}_3 \times_1 A_{31} + \mathcal{X}_1 \times_2 A_{32} + \mathcal{X}_2 \times_3 A_{33} = \mathcal{B}_3, \end{cases} \tag{12}$$

with its parameters given by

$$\begin{aligned} A_{11} &= M + 2rN + \frac{100}{(I_1+1)^2} eye(I_1),\ A_{12} = eye(I_2),\ A_{13} = eye(I_3), \\ A_{21} &= eye(I_1),\ A_{22} = M + 2rN + \frac{100}{(I_2+1)^2} eye(I_2),\ A_{23} = eye(I_3), \\ A_{31} &= eye(I_1),\ A_{32} = eye(I_2),\ A_{33} = M + 2rN + \frac{100}{(I_3+1)^2} eye(I_3), \\ \mathcal{B}_i &= tenrand(I_1, I_2, I_3),\ i = 1, 2, 3, \end{aligned}$$

where

$$M = tridiag(-1, 2, -1),\ N = tridiag(0.5, 0, -0.5).$$

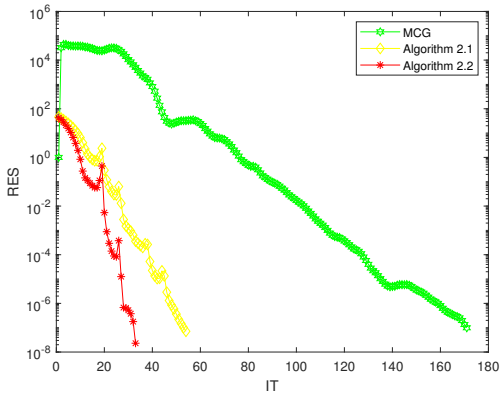

Figure 1: Convergence curves of Example 3.1 with $m = 10$.

Table 2: Comparison results of Example 3.2.

Algorithms	**MCG** [35]			**Algorithm 2.1**			**Algorithm 2.2**		
$[I_1, I_2, I_3]$	IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
$[3, 4, 5]$	37	0.3106	7.1819e-08	29	0.2265	2.4183e-09	10	0.1389	2.4964e-08
$[5, 7, 9]$	141	1.1514	8.3830e-08	71	0.7592	2.8053e-08	54	0.5073	8.6670e-08
$[7, 10, 9]$	271	2.1072	9.3842e-08	117	1.2020	6.0249e-08	85	0.8645	6.0355e-08
$[10, 10, 10]$	442	4.0603	8.8554e-08	155	1.6497	5.7904e-08	117	1.1648	9.5092e-08
$[15, 15, 15]$	†	†	†	696	7.1946	9.6477e-08	535	5.2679	6.0540e-08

In this test, we set the initial guesses as $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}_3 = tenzeros(I_1, I_2, I_3)$, and $r = 0.5$. The obtained numerical results were reported in Table 3 and plotted the convergence curves of all the compared algorithms for $I_1 = 7$, $I_2 = 10$, $I_3 = 9$ in Fig.2. As seen from this table, the number of iteration steps to Algorithm 2.1 does not exceed $I_1 I_2 I_3 + 1$ when the acquired accuracy is reached, which coincides with Theorem 2.2. Also, we can observe that the iterations and computing time corresponding to Algorithms 2.1 and 2.2 are commonly less than that of MCG, in which Algorithm 2.2 performs at their best. The curves in Fig.2 show that our algorithms converge much faster than MCG.

Example 3.3. (Lv and Ma, 2020 [35].) Consider the system of Sylvester tensor equations (12) whose parameters are defined as

$$A_{11} = -tril(rand(I_1, I_1), 1) + diag(1 + diag(rand(I_1))),\ A_{12} = tril(rand(I_2, I_2), 1) + diag(1.5 + diag(rand(I_2))),$$
$$A_{13} = triu(rand(I_3, I_3), 1) + diag(2.5 + diag(rand(I_3))),\ A_{21} = tril(rand(I_1, I_1), 1) + diag(1 + diag(rand(I_1))),$$
$$A_{22} = tril(rand(I_2, I_2), 1) - diag(2 + diag(rand(I_2))),\ A_{23} = tril(rand(I_3, I_3), 1) + diag(3 + diag(rand(I_3))),$$
$$A_{31} = triu(rand(I_1, I_1), 1) + diag(1 + diag(rand(I_1))),\ A_{32} = triu(rand(I_2, I_2), 1) + diag(2 + diag(rand(I_2))),$$
$$A_{33} = triu(rand(I_3, I_3), 1) - diag(1.5 + diag(rand(I_3))),\ \mathcal{B}_i = tenrand(I_1, I_2, I_3), i = 1, 2, 3.$$

We set $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}_3 = tenrand(I_1, I_2, I_3)$ to be the initial values, and applied the proposed algorithms and MCG to deal with Eqs.(12). The obtained comparison results were reported in Table 3. We see from this table that the performance of the proposed algorithms is still better than that of MCG in terms of iterations and computing time.

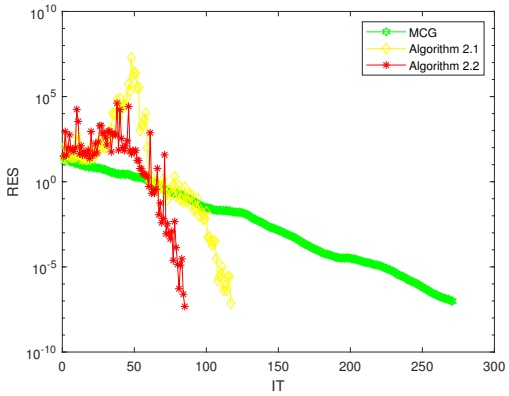

Figure 2: Convergence curves of Example 3.2 with $I_1 = 7, I_2 = 10, I_3 = 9$.

Table 3: Comparison results of Example 3.3.

Algorithms	**MCG** [35]			**Algorithm 2.1**			**Algorithm 2.2**		
$[I_1, I_2, I_3]$	IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
[5, 5, 5]	254	2.1077	9.3305e-08	178	1.4362	5.0525e-08	109	0.9536	3.2942e-08
[5, 10, 15]	491	3.7641	9.6892e-08	215	2.2212	8.2790e-08	148	1.3571	4.6815e-08
[10, 10, 10]	402	4.1542	9.0112e-08	249	2.6857	7.5969e-08	160	1.5032	9.3366e-08
[10, 15, 10]	1682	12.9351	9.8441e-08	622	6.4348	9.4728e-08	454	4.0409	8.4176e-08
[15, 15, 15]	†	†	†	1745	18.8484	9.6751e-08	1171	10.1202	8.2844e-08

In the following example, we test the efficiency of Algorithms 2.1 and 2.2 applied to color image restoration problems, and compare them numerically with MCG. Color image, denoted by $\mathcal{X}$, is a tensor of order $m \times n \times 3$, and each of its frames is a gray image consisting of $m \times n$ pixel values in the range $[0, d]$. The parameter $d = 255$ denotes the maximum possible pixel value. In this test, we generate a blurred and noisy-free color image $\mathcal{B}$ by

$$\mathcal{X} \times_1 A_1 + \mathcal{X} \times_2 A_2 = \mathcal{B}, \tag{13}$$

where $\mathcal{X} \in \mathbb{R}^{m \times n \times 3}$ represents the original image, and A_1, A_2 are the burring matrices of appropriate dimensions. Denote $\mathcal{X}_{restored}$ as the restored color image. We evaluate the performance of all algorithms by the peak signal-to-noise ratio (PSNR) in decibels (dB):

$$\mathrm{PSNR}\,(\mathcal{X}) = 10 \log_{10}\left(\frac{3mnd^2}{\|\mathcal{X} - \mathcal{X}_{restored}\|^2}\right),$$

and compute the relative error:

$$\mathrm{RRE}\,(\mathcal{X}) = \frac{\|\mathcal{X} - \mathcal{X}_{restored}\|}{\|\mathcal{X}\|}.$$

Example 3.4. In this example, we test the popular color image 'Lena' of size $256 \times 256 \times 3$, i.e., $\mathcal{X}$ in Eq.(13), and the blurring matrices $A_1 = A_2 = F \otimes G \in \mathbb{R}^{256 \times 256}$ [40, 41], where $F = (f_{ij}^{(1)})_{1 \le i,j \le 16}$ and $G = (g_{ij}^{(2)})_{1 \le i,j \le 16}$ are the Toeplitz matrices of dimension 256×256 with their entries given by

$$a_{ij}^{(1)} = \begin{cases} \dfrac{1}{\sigma\sqrt{2\pi}} \exp\left(-\dfrac{(i-j)^2}{2\sigma^2}\right), & |i-j| \le r, \\ 0, \quad \text{otherwise}, \end{cases} \qquad a_{ij}^{(2)} = \begin{cases} \dfrac{1}{2s-1}, & |i-j| \le s, \\ 0, \quad \text{otherwise}. \end{cases}$$

Here we depict the original and the blurred and noisy-free images in Fig.3 and set $r = s = 3$ and $\sigma = 1$.

We implemented all the compared algorithms for solving Eq.(13) and displayed the restored images after executing 20 iterations in Fig.4. From this figure, we can observe that the proposed algorithms are able to restore the color image with higher quality than that of MCG. On the other hand, the performance of Algorithm 2.2 is slightly better than Algorithm 2.1.

Figure 3: *Left* the original 'Lena' image. *Right* the blurred and noisy-free image.

Figure 4: Restored images after executing 20 iterations for all the compared algorithms. *Left* MCG with PSNR($\mathcal{X}$) = 32.9795 and RRE($\mathcal{X}$) = 1.8134e−02. *Middle* Algorithm 2.1 with PSNR($\mathcal{X}$) = 34.3291 and RRE($\mathcal{X}$) = 7.6577e−03. *Right* Algorithm 2.2 with PSNR($\mathcal{X}$) = 34.9094 and RRE($\mathcal{X}$) = 2.5198e−03.

4. Conclusion

In this paper, we focus mainly on the iterative solutions of generalized coupled Sylvester tensor equations, one of whose reduced version applies to color image restoration. We present the biconjugate A-orthogonal residual method in its tensor form for solving the tensor equations. The theoretical analysis illustrates that the mentioned algorithm converges to the exact solution group in the absence of round-off errors at most finite steps. To further improve its performance, we develop a transpose-free variant of the BiCOR_BTF, i.e., the tensor form of the CORS method to solve the tensor equations. The numerical results, including when the proposed algorithms are tested with some randomly generated data and a color image restoration problem, illustrating the superiority of our algorithms compared with MCG in terms of iterations and computing time.

References

[1] L.Q. Qi, Z.Y. Luo, *Tensor analysis: spectral theory and special tensors*, SIAM, Philadelphia, 2017.
[2] L.Q. Qi, *Eigenvalues of a real supersymmetric tensor*, J. Symbolic Comput. **40** (2005), 1302-1324.
[3] L.Q. Qi, H. Chen, Y. Chen, *Tensor eigenvalues and their applications*, Springer, 2018.
[4] T. Kolda, B. Bader, *The TOPHITS model for higher-order web link analysis, Workshop on Link Analysis, Counterterrorism Security*, Minnesota. **7** (2006), 26-29.
[5] T. Kolda, B. Bader, *Tensor decompositions and applications*, SIAM Rev. **51** (2009), 455-500.
[6] T. Li, Q.W. Wang, *Structure preserving quaternion full orthogonalization method with applications*, Numer. Linear Algebra Appl., (2023), e2495.
[7] X.F. Zhang, W. Ding, T. Li, *Tensor form of GPBiCG algorithm for solving the generalized Sylvester quaternion tensor equations*, J. Franklin Inst., **360** (2023), 5929-5946.
[8] X.F. Zhang, T. Li, Y.G. Ou, *Iterative solutions of generalized Sylvester quaternion tensor equations*, Linear and Multilinear Algebra, (2023), 1-20.
[9] J. Weaver, *Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues, and eigenvectors*, Am. Math. Month. **92** (1985), 711-717.
[10] L. Datta, S. Morgera, *Some results on matrix symmetries and a pattern recognition application*, IEEE Trans. Signal Process. **34** (1986), 992-994.
[11] L. Datta, S. Morgera, *On the reducibility of centrosymmetric matrices-applications in engineering problems*, Circ. Syst. Signal Process. **8** (1989), 71-96.
[12] Q. Li, B. Zheng, *Scaled three-term derivative-free methods for solving large-scale nonlinear monotone equations*, Numer. Algorithms **87** (2021), 1343-1367.
[13] F.X. Zhang, M.S. Wei, Y. Li, et al., *An efficient method for least-squares problem of the quaternion matrix equation*, Linear and Multilinear Algebra (2020), 1-13.
[14] A. Malek, S.H. Momeni-Masuleh, *A mixed collocation-finite difference method for 3D microscopic heat transport problem*, J. Comput. Appl. Math. **217** (2008), 137-147.
[15] B.W. Li, S. Tian, Y.S. Sun, et al., *Schur-decomposition for 3D matrix equations and its application in solving radiative discrete ordinates equations discretized by Chebyshev collocation spectral method*, J. Comput. Phys. **229** (2010), 1198-1212.
[16] L. Grasedyck, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing **72** (2004), 247-265.
[17] Z.Z. Bai, G.H. Golub, M.K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl. **24**, (2003) 603-626.
[18] Z. Chen, L.Z. Lu, *A gradient based iterative solutions for Sylvester tensor equations*, Math. Probl. Eng. **2013** (2013).
[19] Z. Chen, L.Z. Lu, *A projection method and Kronecker product preconditioner for solving Sylvester tensor equations*, Sci. China Math. **55** (2012), 1281-1292.
[20] F.P. Ali Beik, F.S. Movahed, S. Ahmadi-Asl, *On the Krylov subspace methods based on tensor format for positive definite Sylvester tensor equations*, Linear Algebra Appl. **23** (2016), 444-466.
[21] F. Toutounian, S. Karimi, *Global least squares method (GlLSQR) for solving general linear systems with several right-hand sides*, Appl. Math. Comput. **178** (2006), 452-460.
[22] B.H. Huang, W. Li, *Numerical subspace algorithms for solving the tensor equations involving Einstein product*, Numer. Linear Algebra Appl. **28** (2021), e2351.
[23] M. Najafi-Kalyani, F.P. Ali Beik, K. Jbilou, *On global iterative schemes based on Hessenberg process for (ill-posed) Sylvester tensor equations*, J. Comput. Appl. Math. **373** (2020), 112216.
[24] A.H. Bentbib, S. El-Halouy, E.M. Sadek, *Krylov subspace projection method for Sylvester tensor equation with low rank right-hand side*, Numer. Algorithms (2020), 1-20.
[25] M. Heyouni, F. Saberi-Movahed, A. Tajaddini, *A tensor format for the generalized Hessenberg method for solving Sylvester tensor equations*, J. Comput. Appl. Math. **377** (2020), 112878.
[26] Q.W. Wang, X.X. Wang, *A system of coupled two-sided Sylvester-type tensor equations over the quaternion algebra*, Taiwanese J. Math. **24** (2020), 1399-1416.
[27] M.S. Mehany, Q.W. Wang, *A new generalization of a system of two-sided coupled Sylvester-like quaternion tensor equations*, arXiv preprint arXiv:2205.14810, 2022.
[28] Z.H. He, *The general solution to a system of coupled Sylvester-type quaternion tensor equations involving η-Hermicity*, Bulletin Iranian Math. Soc. **45** (2019), 1407-1430.
[29] Z.H. He, M. Wang, *Solvability conditions and general solutions to some quaternion matrix equations*, Math. Methods Appl. Sci. **44** (2021), 14274-14291.
[30] Z.H. He, *Some new results on a system of Sylvester-type quaternion matrix equations*, Linear and Multilinear Algebra **69** (2021), 3069-3091.
[31] Z.H. He, C. Navasca, X.X. Wang, *Decomposition for a quaternion tensor triplet with applications*, Adv. Appl. Clifford Algebr. **32** (2022), 1-19.
[32] Z.H. He, X.X. Wang, Y.F. Zhao, *Eigenvalues of quaternion tensors with applications to color video processing*, J. Sci. Comput. **94** (2023), 1.
[33] Z.H. He, X.N. Zhang, Y.F. Zhao, *The solvability of a system of quaternion matrix equations involving ϕ-skew-Hermicity*, Symmetry **14** (2022), 1273.
[34] Z.H. He, W.L. Qin, X.X. Wang, *Some applications of a decomposition for five quaternion matrices in control system and color image processing*, Comput. Appl. Math. **40** (2021), 205.

[35] C.Q. Lv, C.F. Ma, *A modified CG algorithm for solving generalized coupled Sylvester tensor equations*, Appl. Math. Comput. **365** (2020), 124699.
[36] Y. Saad, *Iterative methods for sparse linear systems*, second ed., SIAM, Philadelphia (82) 2003.
[37] Y.F. Jing, T.Z. Huang, Y. Zhang, et al., *Lanczos-type variants of the COCR method for complex nonsymmetric linear systems*, J. Comput. Phys. **228** (2009), 6376-6394.
[38] B. Carpentieri, Y.F. Jing, T.Z. Huang, *The BiCOR and CORS iterative algorithms for solving nonsymmetric linear systems*, SIAM J. Sci. Comput. **33** (2011) 3020-3036.
[39] M. Hajarian, *Developing BiCOR and CORS methods for coupled Sylvester-transpose and periodic Sylvester matrix equations*, Appl. Math. Model. **39** (2015), 6073-6084.
[40] J.F. Li, W. Li, R. Huang, *An efficient method for solving a matrix least squares problem over a matrix inequality constraint*, Comput. Optim. Appl. **63** (2016), 393-423.
[41] A. Bouhamidi, R. Enkhbat, K. Jbilou, *Conditional gradient Tikhonov method for a convex optimization problem in image restoration*, J. Comput. Appl. Math. **255** (2014) 580-592.