

Article

A Modified Conjugate Residual Method and Nearest Kronecker Product Preconditioner for the Generalized Coupled Sylvester Tensor Equations

Tao Li ¹, Qing-Wen Wang ^{2,3*}  and Xin-Fang Zhang ¹

¹ Department of Mathematics, Key Laboratory of Engineering Modeling and Statistical Computation of Hainan Province, Hainan University, Haikou 570228, China; tli@hainanu.edu.cn (T.L.); 995272@hainanu.edu.cn (X.-F.Z.)

² Department of Mathematics, Shanghai University, Shanghai 200444, China

³ Collaborative Innovation Center for the Marine Artificial Intelligence, Shanghai 200444, China

* Correspondence: wqw@t.shu.edu.cn or wqw@shu.edu.cn

Abstract: This paper is devoted to proposing a modified conjugate residual method for solving the generalized coupled Sylvester tensor equations. To further improve its convergence rate, we derive a preconditioned modified conjugate residual method based on the Kronecker product approximations for solving the tensor equations. A theoretical analysis shows that the proposed method converges to an exact solution for any initial tensor at most finite steps in the absence of round-off errors. Compared with a modified conjugate gradient method, the obtained numerical results illustrate that our methods perform much better in terms of the number of iteration steps and computing time.

Keywords: generalized coupled Sylvester tensor equations; modified conjugate residual method; Kronecker product approximations; preconditioned modified conjugate residual method

MSC: 15A69; 65F08; 65F10



Citation: Li, T.; Wang, Q.-W.; Zhang, X.-F. A Modified Conjugate Residual Method and Nearest Kronecker Product Preconditioner for the Generalized Coupled Sylvester Tensor Equations. *Mathematics* **2022**, *10*, 1730. <https://doi.org/10.3390/math10101730>

Academic Editors: Zhongshan Li and
Michael Voskoglou

Received: 17 April 2022

Accepted: 13 May 2022

Published: 18 May 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the Creative Commons
Attribution (CC BY) license ([https://
creativecommons.org/licenses/by/
4.0/](https://creativecommons.org/licenses/by/4.0/)).

1. Introduction

As is well known, the tensor equations as generalizations of matrix equations have been intensively applied in the field of theory and applications. Particularly, the generalized coupled Sylvester tensor equations have drawn significant attention because they play an increasingly important role in control theory [1–8], finite difference [9], finite element [10] and information retrieval [11], etc. This motivated us to propose two effective iterative methods for solving the tensor equations.

In this paper, we are interested in developing two iterative algorithms to solve the following generalized coupled Sylvester tensor equations:

$$\begin{cases} \mathcal{X}_1 \times_1 A_{11} + \mathcal{X}_2 \times_2 A_{12} + \cdots + \mathcal{X}_{n-1} \times_{n-1} A_{1(n-1)} + \mathcal{X}_n \times_n A_{1n} = B_1, \\ \mathcal{X}_2 \times_1 A_{21} + \mathcal{X}_3 \times_2 A_{22} + \cdots + \mathcal{X}_n \times_{n-1} A_{2(n-1)} + \mathcal{X}_1 \times_n A_{2n} = B_2, \\ \quad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ \mathcal{X}_n \times_1 A_{n1} + \mathcal{X}_1 \times_2 A_{n2} + \cdots + \mathcal{X}_{n-2} \times_{n-1} A_{n(n-1)} + \mathcal{X}_{n-1} \times_n A_{nn} = B_n, \end{cases} \quad (1)$$

where the matrices $A_{ij} \in \mathbb{R}^{I_j \times I_j}$, the right-hand side tensors $\mathcal{B}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$ are given ($i, j = 1, \dots, n$), and $\mathcal{X}_j \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$ must be unknown. As a special case of Equation (1) ($\mathcal{X}_j = \mathcal{X}$ for $j = 1, 2, 3$), the third-order Sylvester tensor equation (known as 3D matrix equation) is

$$\mathcal{X} \times_1 A_1 + \mathcal{X} \times_2 A_2 + \mathcal{X} \times_3 A_3 = \mathcal{B}, \quad (2)$$

which arises frequently from the discretization of radiative discrete ordinates equations [12,13] and high-dimensional partial differential equation [14]. One application of Equation (2) in heat transfer is to solve the three-dimensional microscopic heat transport problem [15]

$$\frac{\rho C_p}{\kappa} \left(\frac{\partial U}{\partial t} + \tau_q \frac{\partial^2 U}{\partial t^2} \right) = \nabla^2 U + \tau_q \left(\frac{\partial^3 U}{\partial t \partial x^2} + \frac{\partial^3 U}{\partial t \partial y^2} \right) + \tau_U \frac{\partial^3 U}{\partial t \partial z^2} + \frac{Q + \tau_q \partial Q / \partial t}{\kappa}, \quad (3)$$

where Q is a heat source, U is temperature, C_p is specific heat, ρ is density, κ is conductivity, and τ_q and τ_U are the time lags of the heat flux and temperature gradient, respectively. From the mixed collocation-finite difference method, Equation (3) can be discretized as Equation (2) with its coefficient matrices given by

$$\begin{aligned} (A_1)_{ji} &= \begin{cases} -\tau_q \left(\frac{\pi}{\delta_x} \right)^2 \left(\frac{I_x^2 + 2}{3} \right) - \frac{\Delta t + \tau_q}{\alpha \Delta t}, & \text{if } i = j \\ -2\tau_q \left(\frac{\pi}{\delta_x} \right)^2 \frac{(-1)^{i+j}}{\sin^2 \left[\frac{1}{2} \left(\frac{2\pi\beta_j}{\delta_x} - x_i \right) \right]}, & \text{if } i \neq j, \end{cases} \\ (A_2)_{sk} &= \begin{cases} -\tau_q \left(\frac{\pi}{\delta_y} \right)^2 \left(\frac{I_y^2 + 2}{3} \right), & \text{if } k = s \\ -2\tau_q \left(\frac{\pi}{\delta_y} \right)^2 \frac{(-1)^{k+s}}{\sin^2 \left[\frac{1}{2} \left(\frac{2\pi\gamma_s}{\delta_y} - y_k \right) \right]}, & \text{if } k \neq s, \end{cases} \\ (A_3)_{qp} &= \begin{cases} -\tau_U \left(\frac{\pi}{\delta_z} \right)^2 \left(\frac{I_z^2 + 2}{3} \right), & \text{if } p = q \\ -2\tau_U \left(\frac{\pi}{\delta_z} \right)^2 \frac{(-1)^{p+q}}{\sin^2 \left[\frac{1}{2} \left(\frac{2\pi\eta_q}{\delta_z} - z_p \right) \right]}, & \text{if } p \neq q, \end{cases} \end{aligned} \quad (4)$$

where $x_i = \frac{2\pi i}{I_x}$, $y_k = \frac{2\pi k}{I_y}$ and $z_p = \frac{2\pi p}{I_z}$, and $\beta_j = \frac{j\delta_x}{I_x}$, $\gamma_s = \frac{s\delta_y}{I_y}$ and $\eta_z = \frac{q\delta_z}{I_z}$ for $i, j = 0, \dots, I_x - 1$, $k, s = 0, \dots, I_y - 1$ and $p, q = 0, \dots, I_z - 1$. Some effective iterative algorithms have been proposed for solving Equation (2). For example, according to the hierarchical identification principle, Chen and Lv proposed a gradient-based iterative method and its modification version [16] for solving the tensor equation when it has a unique solution. To accelerate the rate of convergence of the GI method, Ali Beik and Ahmadi Asl [17] derived a so-called residual norm steepest descent algorithm for solving the tensor equation. If $\mathcal{X}_j = \mathcal{X}$ for $j = 1, \dots, n$, then Equation (1) reduces to the generalized Sylvester tensor equation

$$\mathcal{X} \times_1 A_1 + \mathcal{X} \times_2 A_2 + \dots + \mathcal{X} \times_n A_n = \mathcal{B}, \quad (5)$$

which has been widely applied in the signal image, blind source separation, restoration of color and hyperspectral images. For more details, one may refer to [18–23] and the references therein.

In the last decade, some efficient algorithms for solving Equation (5) have been developed well. For example, Chen and Lv [24] proposed a generalized minimum residual (GMRES) method and its preconditioned version in their tensor forms for solving Equation (5). By employing the Arnoldi process and the full orthogonalization method, Ali Beik et al. [25] introduced the conjugate gradient and nested conjugate gradient methods to search the solution of the generalized Sylvester tensor equation, respectively. Moreover, Karimi and Dehghan [26] presented a new method based on the global Bidiag 1 process [27] for finding an approximate solution of (5), which is a promising method. Wang et al. [28] studied the performance of an iterative algorithm that obtains the least square solution of Equation (5) over the quaternion algebra. Additionally, Xu and Wang [29] extended the BiCG and BiCR methods to solve the Stein tensor equation, which is a general form of the Lyapunov matrix equation. Li et al. [30] developed five numerical algorithms for solving the discrete Lyapunov tensor equation. Huang and Li [31] proposed 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. Hajarian [32] gave the matrix form of the biconjugate residual algorithm for studying the generalized reflexive and anti-reflexive solutions of a generalized Sylvester matrix equation. Then, Hajarian [33] derived four efficient iterative methods for obtaining the reflexive periodic solutions of general periodic matrix equations. Additionally, Hajarian [34] presented a conjugate gradient-like method for solving a general tensor equation with respect to the Einstein product. In [35], Najafi-Kalyani et al. derived some effective iterative methods for solving the tensor Equation (5), which are based on the tensor form of global Hessenberg process. Liu et al. [36] investigated the solvability of a quaternion matrix equation. Mehany and Wang [37] studied some necessary and sufficient conditions for the solution of the three symmetrical systems of coupled Sylvester-like quaternion matrix equations. Combining the standard Tikhonov regularization technique, a new iterative method to solve the ill-posed tensor Equation (5), was also established. By CP decomposition, Bentbib et al. [38] developed Arnoldi-based methods (block and global), which possess the global convergence for solving the tensor Equation (5) efficiently. Additionally, Heyouni et al. [39] established the generalized Hessenberg method in its tensor form for solving (5). However, so far, there has been little research on solving the generalized coupled Sylvester tensor Equation (1), numerically. Only Lv and Ma [40] presented a modified conjugate gradient (MCG) method for investigating its iterative solutions.

For solving an n -by- n symmetric linear system $Ax = b$, the classical conjugate residual (CR) method as a special case of GMRES method was derived in [41]. In this case, the residual vectors are always A -orthogonal, and the vectors Ap_i are also orthogonal for $i = 0, 1, \dots, n$. This fact shows that the method converges fast to an exact solution within finite steps in the absence round-off errors. Compared with the CG method, the computational work and storage costs of CR method may increase slightly, but its convergence behavior is more smooth. Note that the classical CR method cannot be extended directly to solve the tensor Equation (1), because it is nonsymmetric. So, we propose a modified conjugate residual (MCR) method for studying Equation (1). Also note that preconditioning is an effective mean of reaching improved convergence of iterative methods. Preconditioners, such as incomplete LU factorizations preconditioner [42], Neumann preconditioner [43], and nearest Kronecker product (NKP) approximate preconditioner [44], have been confirmed as powerful tools for solving large linear systems. In [45], some iterative methods with the NKP preconditioner performed much better than other preconditioners. In this paper, we present a preconditioned MCR method based on Kronecker product approximations for solving Equation (1), which is superior to the MCR method.

The rest of this paper is organized as follows: In Section 2, we recall some notations and preliminaries that will be used in the sequel. In Section 3, we formulate a modified conjugate residual method to solve the generalized coupled Sylvester tensor Equation (1), and present its preconditioned version to speed up the rate of convergence of MCR method. The established convergence analysis shows that an exact solution to Equation (1) can be given for any initial tensor at most finite steps in the absence of round-off errors. In Section 4, we provide some numerical examples to illustrate the effectiveness of the methods proposed in here. Finally we draw some conclusions in Section 5.

2. Preliminaries

In this section, we recall some definitions and useful results that will be used in the sequel. Let $\mathbb{R}$ denote the real number field. Given a positive integer n , an order n real tensor $\mathcal{X} = (x_{i_1 i_2 \dots i_n}) (1 \leq i_j \leq I_j, j = 1, 2, \dots, n)$ is a multidimensional array consisting of $I_1 I_2 \dots I_n$ entries [46,47]. We denote the set of all real tensors by $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$. If $n = 2$, then we call the tensor $\mathcal{A}$ as an $I_1 \times I_2$ matrix, denoted by A . We denote the set of real matrices by $\mathbb{R}^{I_1 \times I_2}$. An $I_1 \times I_1$ identity matrix is denoted by I^{I_1} . For a matrix $A \in \mathbb{R}^{I_1 \times I_2}$,

$A^T \in \mathbb{R}^{I_2 \times I_1}$ is then said to be the transpose of A . Given matrices $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ and B , their Kronecker product, denoted $A \otimes B$, is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}.$$

For matrices A and B , $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$. The unfolding of a tensor $\mathcal{X}$ along mode 1, denoted by $X_{(1)}$, is an $I_1 \times I_2 I_3 \cdots I_n$ matrix with its column being a column of $\mathcal{X}$ along mode 1. The symbol $\text{vec}(X)$ denotes a vectorized matrix by stacking its column to form a vector. $\mathcal{O}$ is called a zero tensor if its entries are zero. Given tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, their inner product is a scalar given by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \text{vec}(X_{(1)})^T \text{vec}(Y_{(1)}) = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_n} y_{i_1 i_2 \cdots i_n}. \quad (6)$$

If $\langle \mathcal{X}, \mathcal{Y} \rangle = 0$, then $\mathcal{X}$ and $\mathcal{Y}$ are said to be orthogonal. The Frobenius norm over $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m}$ is defined as

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

For a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ and a matrix $A \in \mathbb{R}^{m \times I_k}$, their k -mode product, denoted by $\mathcal{X} \times_k A$, is an n th order $I_1 \times \cdots \times I_{k-1} \times m \times I_{k+1} \times \cdots \times I_n$ dimensional tensor with its entries given by

$$(\mathcal{X} \times_k A)_{i_1 \cdots i_{k-1} j i_{k+1} \cdots i_n} = \sum_{i_k=1}^{I_k} x_{i_1 \cdots i_k \cdots i_n} a_{j i_k}, \quad j = 1, \cdots, m. \quad (7)$$

From the definition of unfolding and the results in [48,49], the following statements hold:

- (1) If $\mathcal{Y} = \mathcal{X} \times_1 A_1 \times_2 A_2 \times_3 \cdots \times_n A_n$, then $Y_{(1)} = A_1 X_{(1)} (A_n \otimes A_{n-1} \otimes \cdots \otimes A_2)$ holds;
- (2) For different mode multiplications ($k \neq n$), it follows that $\mathcal{X} \times_n A \times_k B = \mathcal{X} \times_k B \times_n A$;
- (3) If $k = n$, then $\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA)$;
- (4) The k -mode multiplication commutes with respect to the inner product, that is, $\langle \mathcal{X}, \mathcal{Y} \times_k A \rangle = \langle \mathcal{X} \times_k A^T, \mathcal{Y} \rangle$.

3. A Modified Conjugate Residual Method for Solving the Tensor Equations

In this section, we propose a modified conjugate residual method for solving the tensor equation Equation (1), and establish its precondition version to speed up the rate of convergence. First of all, we present a necessary and sufficient condition for the existence of solution of (1). From the above results in Section 2, Equation (1) can be rewritten as the following linear system of equations

$$Ax = b, \quad (8)$$

where

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

$$x = \begin{bmatrix} \text{vec}(\mathcal{X}_{1(1)}) \\ \text{vec}(\mathcal{X}_{2(1)}) \\ \vdots \\ \text{vec}(\mathcal{X}_{n(1)}) \end{bmatrix}, \quad b = \begin{bmatrix} \text{vec}(\mathcal{B}_{1(1)}) \\ \text{vec}(\mathcal{B}_{2(1)}) \\ \vdots \\ \text{vec}(\mathcal{B}_{n(1)}) \end{bmatrix}.$$

Then, we have the following results.

Lemma 1 ([40]). Equation (1) has a solution if and only if the linear system of Equation (8) is consistent. In this case, its solution is unique if the coefficient matrix A is nonsingular.

Actually, for Equation (8), it is difficult to find its solution by the classical conjugate residual method because the matrix A cannot be guaranteed to be symmetric. Moreover, the above method cannot be achieved in actual implementations because its computational work and storage costs are very expensive for larger $I_j, j = 1, \dots, n$. Thereby we propose a modified conjugate residual method based on tensor form for solving Equation (1). Similar to [40], we define two operators $\mathcal{L}_i$ and $\tilde{\mathcal{L}}_i (i = 1, \dots, n)$ over $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ as

$$\begin{cases} \mathcal{L}_1(\mathcal{X}_1, \dots, \mathcal{X}_n) = \mathcal{X}_1 \times_1 A_{11} + \mathcal{X}_2 \times_2 A_{12} + \cdots + \mathcal{X}_n \times_n A_{1n}, \\ \mathcal{L}_2(\mathcal{X}_1, \dots, \mathcal{X}_n) = \mathcal{X}_2 \times_1 A_{21} + \mathcal{X}_3 \times_2 A_{22} + \cdots + \mathcal{X}_1 \times_n A_{2n}, \\ \vdots \\ \mathcal{L}_n(\mathcal{X}_1, \dots, \mathcal{X}_n) = \mathcal{X}_n \times_1 A_{n1} + \mathcal{X}_1 \times_2 A_{n2} + \cdots + \mathcal{X}_{n-1} \times_n A_{nn} \end{cases} \quad (9)$$

and

$$\begin{cases} \tilde{\mathcal{L}}_1(\mathcal{R}_1, \dots, \mathcal{R}_n) = \mathcal{R}_1 \times_1 A_{11}^T + \mathcal{R}_2 \times_n A_{2n}^T + \cdots + \mathcal{R}_n \times_2 A_{n2}^T, \\ \tilde{\mathcal{L}}_2(\mathcal{R}_1, \dots, \mathcal{R}_n) = \mathcal{R}_1 \times_2 A_{12}^T + \mathcal{R}_2 \times_1 A_{21}^T + \cdots + \mathcal{R}_n \times_3 A_{n3}^T, \\ \vdots \\ \tilde{\mathcal{L}}_n(\mathcal{R}_1, \dots, \mathcal{R}_n) = \mathcal{R}_1 \times_n A_{1n}^T + \mathcal{R}_2 \times_{n-1} A_{2(n-1)}^T + \cdots + \mathcal{R}_n \times_1 A_{n1}^T, \end{cases} \quad (10)$$

where

$$\begin{cases} \mathcal{R}_1 = \mathcal{B}_1 - \mathcal{L}_1(\mathcal{X}_1, \dots, \mathcal{X}_n), \\ \mathcal{R}_2 = \mathcal{B}_2 - \mathcal{L}_2(\mathcal{X}_1, \dots, \mathcal{X}_n), \\ \vdots \\ \mathcal{R}_n = \mathcal{B}_n - \mathcal{L}_n(\mathcal{X}_1, \dots, \mathcal{X}_n). \end{cases} \quad (11)$$

From Equations (9) and (10), we can obtain the following lemma.

Lemma 2. If tensors $\mathcal{X}_i, \mathcal{Y}_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}, i = 1, \dots, n$, and scalars $\mu, \nu \in \mathbb{R}$, then

$$\begin{aligned} \mathcal{L}_i(\mu \mathcal{X}_1 + \nu \mathcal{Y}_1, \dots, \mu \mathcal{X}_n + \nu \mathcal{Y}_n) &= \mu \mathcal{L}_i(\mathcal{X}_1, \dots, \mathcal{X}_n) + \nu \mathcal{L}_i(\mathcal{Y}_1, \dots, \mathcal{Y}_n), \\ \tilde{\mathcal{L}}_i(\mu \mathcal{X}_1 + \nu \mathcal{Y}_1, \dots, \mu \mathcal{X}_n + \nu \mathcal{Y}_n) &= \mu \tilde{\mathcal{L}}_i(\mathcal{X}_1, \dots, \mathcal{X}_n) + \nu \tilde{\mathcal{L}}_i(\mathcal{Y}_1, \dots, \mathcal{Y}_n), \end{aligned} \quad (12)$$

i.e., $\mathcal{L}_i$ and $\tilde{\mathcal{L}}_i$ are linear operators from $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ to itself. Moreover,

$$\begin{aligned} &\langle \mathcal{X}_1, \mathcal{L}_1(\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n) \rangle + \langle \mathcal{X}_2, \mathcal{L}_2(\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n) \rangle + \cdots + \langle \mathcal{X}_n, \mathcal{L}_n(\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n) \rangle \\ &= \langle \mathcal{Y}_1, \tilde{\mathcal{L}}_1(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n) \rangle + \langle \mathcal{Y}_2, \tilde{\mathcal{L}}_2(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n) \rangle + \cdots + \langle \mathcal{Y}_n, \tilde{\mathcal{L}}_n(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n) \rangle. \end{aligned} \quad (13)$$

Proof. For the first statement, we only prove that Equation (12) holds for $i = 1$; the rest ($i = 2, \dots, n$) can be deduced similarly. If $\mu, \nu \in \mathbb{R}$, then

$$\begin{aligned}\mathcal{L}_1(\mu\mathcal{X}_1 + \nu\mathcal{Y}_1, \dots, \mu\mathcal{X}_n + \nu\mathcal{Y}_n) &= (\mu\mathcal{X}_1 + \nu\mathcal{Y}_1) \times_1 A_{11} + \dots + (\mu\mathcal{X}_n + \nu\mathcal{Y}_n) \times_n A_{1n}, \\ &= \mu\mathcal{L}_1(\mathcal{X}_1, \dots, \mathcal{X}_n) + \nu\mathcal{L}_1(\mathcal{Y}_1, \dots, \mathcal{Y}_n), \\ \bar{\mathcal{L}}_1(\mu\mathcal{X}_1 + \nu\mathcal{Y}_1, \dots, \mu\mathcal{X}_n + \nu\mathcal{Y}_n) &= (\mu\mathcal{X}_1 + \nu\mathcal{Y}_1) \times_1 A_{11}^T + \dots + (\mu\mathcal{X}_1 + \nu\mathcal{Y}_1) \times_2 A_{n2}^T \\ &= \mu\bar{\mathcal{L}}_1(\mathcal{X}_1, \dots, \mathcal{X}_n) + \nu\bar{\mathcal{L}}_1(\mathcal{Y}_1, \dots, \mathcal{Y}_n).\end{aligned}$$

For Equation (13), it follows that

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

□

Now, we present the following modification of conjugate residual method for solving Equation (1):

From Algorithm 1, one can see that the key computations are the computation of $\mathcal{R}_i^{(k+1)}$ and $\mathcal{P}_i^{(k+1)}$ with $i = 1, 2, \dots, n$. Both of them require $O(I_1 I_2 \dots I_n (\sum_{i=1}^n I_i + 1))$ floating point operations (flops). So, we can obtain that the amount of flops required by each iteration of Algorithm 1 is $O(4n I_1 I_2 \dots I_n (\sum_{i=1}^n I_i + 1))$.

To investigate the convergence of Algorithm 1, we first give some properties of the tensor sequences generated by the algorithm as follows:

Theorem 1. Let $\{\mathcal{R}_l^k\}, \{\mathcal{P}_l^k\}, \{\bar{\mathcal{L}}_l(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\}$ and $\{\mathcal{L}_l(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\}$ ($l = 1, \dots, n, k = 0, 1, \dots$) be the tensor sequences given by Algorithm 1. Then, for $i \neq j, i, j = 0, 1, \dots$, we have

$$\begin{cases} \langle \bar{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i), \bar{\mathcal{L}}_1(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle + \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i), \bar{\mathcal{L}}_n(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle = 0, \\ \langle \mathcal{L}_1(\mathcal{P}_1^i, \dots, \mathcal{P}_n^i), \mathcal{L}_1(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle + \dots + \langle \mathcal{L}_n(\mathcal{P}_1^i, \dots, \mathcal{P}_n^i), \mathcal{L}_n(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle = 0. \end{cases} \quad (14)$$

Moreover, for $j < i$, it follows that

$$\begin{aligned}&\langle \mathcal{R}_1^i, \mathcal{L}_1(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle + \dots + \langle \mathcal{R}_n^i, \mathcal{L}_n(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle \\ &= \langle \mathcal{P}_1^j, \bar{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i) \rangle + \dots + \langle \mathcal{P}_n^j, \bar{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i) \rangle = 0.\end{aligned} \quad (15)$$

Algorithm 1 A modified conjugate residual method for solving Equation (1)

1. Given initial values $\mathcal{X}_i^0, \mathcal{B}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$, $A_{ij} \in \mathbb{R}^{I_j \times I_j}$, $k = 0$, and set $\epsilon > 0$.
2. Compute

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

and

$$\begin{aligned} s_0 &= \|\tilde{\mathcal{L}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2 + \|\tilde{\mathcal{L}}_2(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2 + \dots + \|\tilde{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2, \\ p_0 &= \|\mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \|\mathcal{L}_2(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2, \\ \eta_k &= \|\mathcal{R}_1^k\|^2 + \|\mathcal{R}_2^k\|^2 + \dots + \|\mathcal{R}_n^k\|^2. \end{aligned}$$

3. For $k = 0, 1, 2, \dots$ until $\eta_k < \epsilon$ do:

$$\alpha_k = \frac{s_k}{p_k},$$

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

$$\begin{cases} \tilde{\mathcal{L}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_1 A_{11}^T + \mathcal{R}_2^{k+1} \times_n A_{2n}^T + \dots + \mathcal{R}_n^{k+1} \times_2 A_{n2}^T, \\ \tilde{\mathcal{L}}_2(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_2 A_{12}^T + \mathcal{R}_2^{k+1} \times_1 A_{21}^T \dots + \mathcal{R}_n^{k+1} \times_3 A_{n3}^T, \\ \vdots \\ \tilde{\mathcal{L}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_n A_{1n}^T + \mathcal{R}_2^{k+1} \times_{n-1} A_{2(n-1)}^T + \dots + \mathcal{R}_n^{k+1} \times_1 A_{n1}^T, \end{cases}$$

$$s_{k+1} = \|\tilde{\mathcal{L}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1})\|^2 + \|\tilde{\mathcal{L}}_2(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1})\|^2 + \dots + \|\tilde{\mathcal{L}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1})\|^2.$$

4. Compute $\beta_k = \frac{s_{k+1}}{s_k}$,

$$\begin{cases} \mathcal{P}_1^{k+1} = \tilde{\mathcal{L}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{P}_1^k, \\ \mathcal{P}_2^{k+1} = \tilde{\mathcal{L}}_2(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{P}_2^k, \\ \vdots \\ \mathcal{P}_n^{k+1} = \tilde{\mathcal{L}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{P}_n^k, \end{cases}$$

$$p_{k+1} = \|\mathcal{L}_1(\mathcal{P}_1^{k+1}, \dots, \mathcal{P}_n^{k+1})\|^2 + \|\mathcal{L}_2(\mathcal{P}_1^{k+1}, \dots, \mathcal{P}_n^{k+1})\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^{k+1}, \dots, \mathcal{P}_n^{k+1})\|^2.$$

Proof. The first half of Equation (15) can be obtained by straightforward computation. We now prove that Equations (14) and (15) hold by induction. Suppose $k = 1$, from Lemma 2; it follows that

$$\begin{aligned}
& \langle \bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \bar{\mathcal{L}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) \rangle + \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \bar{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) \rangle \\
&= \langle \bar{\mathcal{L}}_1(\mathcal{R}_1^0 - \alpha_0 \mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0), \dots, \mathcal{R}_n^0 - \alpha_0 \mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)), \bar{\mathcal{L}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) \rangle \\
&+ \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^0 - \alpha_0 \mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0), \dots, \mathcal{R}_n^0 - \alpha_0 \mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)), \bar{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) \rangle \\
&= [\|\bar{\mathcal{L}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2] \\
&- \frac{\|\bar{\mathcal{L}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2}{\|\mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2} [\|\mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2] \\
&= 0, \\
&\langle \mathcal{L}_1(\mathcal{P}_1^1, \dots, \mathcal{P}_n^1), \mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0) \rangle + \dots + \langle \mathcal{L}_n(\mathcal{P}_1^1, \dots, \mathcal{P}_n^1), \mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0) \rangle \\
&= [\langle \mathcal{L}_1(\bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \dots, \bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1)), \mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0) \rangle \\
&+ \dots + \langle \mathcal{L}_n(\bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \dots, \bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1)), \mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0) \rangle] \\
&+ \beta_0 [\langle \mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0), \mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0) \rangle + \dots + \langle \mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0), \mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0) \rangle] \\
&= [\langle \mathcal{L}_1(\bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \dots, \bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1)), \frac{1}{\alpha_0}(\mathcal{R}_1^0 - \mathcal{R}_1^1) \rangle \\
&+ \dots + \langle \mathcal{L}_n(\bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \dots, \bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1)), \frac{1}{\alpha_0}(\mathcal{R}_n^0 - \mathcal{R}_n^1) \rangle] \\
&+ \beta_0 [\|\mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2] \\
&= \frac{1}{\alpha_0} [\langle \bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \bar{\mathcal{L}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) - \bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1) \rangle \\
&+ \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1), \bar{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) - \bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1) \rangle] \\
&+ \beta_0 [\|\mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2] \\
&= -\frac{1}{\alpha_0} [\|\bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1)\|^2] \\
&+ \beta_0 [\|\mathcal{L}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2] \\
&= 0.
\end{aligned}$$

Since

$$\begin{cases} \mathcal{P}_1^0 = \bar{\mathcal{L}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0), \\ \mathcal{P}_2^0 = \bar{\mathcal{L}}_2(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0), \\ \vdots \\ \mathcal{P}_n^0 = \bar{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0), \end{cases}$$

it follows that

$$\langle \mathcal{P}_1^0, \bar{\mathcal{L}}_1(\mathcal{R}_1^1, \dots, \mathcal{R}_n^1) \rangle + \dots + \langle \mathcal{P}_n^0, \bar{\mathcal{L}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) \rangle = 0.$$

If Equations (14) and (15) hold for $0 \leq j < i \leq k(k > 1)$, then for the case $0 \leq j < i \leq k + 1$ and $0 \leq j < k$ we have

$$\begin{aligned}
& \langle \bar{\mathcal{L}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}), \bar{\mathcal{L}}_1(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle + \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}), \bar{\mathcal{L}}_n(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle \\
&= [\langle \bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{P}_1^j \rangle + \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{P}_n^j \rangle]
\end{aligned}$$

[illegible]

$$\begin{aligned}
& + \cdots + \langle \mathcal{P}_n^j, \tilde{\mathcal{L}}_n(\mathcal{R}_1^k - \alpha_k \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \dots, \mathcal{R}_n^k - \alpha_k \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)) \rangle \\
& = \langle \mathcal{R}_1^k - \alpha_k \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{L}_1(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle + \cdots + \langle \mathcal{R}_n^k - \alpha_k \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{L}_n(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle \\
& = [\langle \mathcal{R}_1^k, \mathcal{L}_1(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle + \cdots + \langle \mathcal{R}_n^k, \mathcal{L}_n(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle] \\
& \quad - \alpha_k [\langle \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{L}_1(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle + \cdots + \langle \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{L}_n(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle] \\
& = 0, \\
& \langle \mathcal{P}_1^k, \tilde{\mathcal{L}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) \rangle + \cdots + \langle \mathcal{P}_n^k, \tilde{\mathcal{L}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) \rangle \\
& = [\langle \mathcal{R}_1^k, \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k) \rangle + \cdots + \langle \mathcal{R}_n^k, \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k) \rangle] \\
& \quad - \alpha_k [\langle \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k) \rangle + \cdots + \langle \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k) \rangle] \\
& = [\langle \tilde{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{P}_1^k \rangle + \cdots + \langle \tilde{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{P}_n^k \rangle] \\
& \quad - \alpha_k [\|\mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2 + \cdots + \|\mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2] \\
& = [\langle \tilde{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \tilde{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) + \beta_{k-1} \mathcal{P}_1^{k-1} \rangle + \cdots + \langle \tilde{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \tilde{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \\
& \quad + \beta_{k-1} \mathcal{P}_n^{k-1} \rangle] - \alpha_k [\|\mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2 + \cdots + \|\mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2] \\
& = [\|\tilde{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \cdots + \|\tilde{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2] \\
& \quad - \alpha_k [\|\mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2 + \cdots + \|\mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2] \\
& = 0.
\end{aligned}$$

Consequently, Equations (14) and (15) hold for $0 \leq j < i \leq k$. From the definition of the inner product on $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m}$, it also follows that

$$\begin{cases} \langle \tilde{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i), \tilde{\mathcal{L}}_1(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle + \cdots + \langle \tilde{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i), \tilde{\mathcal{L}}_n(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle \\ = \langle \tilde{\mathcal{L}}_1(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j), \tilde{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i) \rangle + \cdots + \langle \tilde{\mathcal{L}}_n(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j), \tilde{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i) \rangle, \\ \langle \mathcal{L}_1(\mathcal{P}_1^i, \dots, \mathcal{P}_n^i), \mathcal{L}_1(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle + \cdots + \langle \mathcal{L}_n(\mathcal{P}_1^i, \dots, \mathcal{P}_n^i), \mathcal{L}_n(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j) \rangle \\ = \langle \mathcal{L}_1(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j), \mathcal{L}_1(\mathcal{P}_1^i, \dots, \mathcal{P}_n^i) \rangle + \cdots + \langle \mathcal{L}_n(\mathcal{P}_1^j, \dots, \mathcal{P}_n^j), \mathcal{L}_n(\mathcal{P}_1^i, \dots, \mathcal{P}_n^i) \rangle. \end{cases}$$

This shows that Equation (14) holds for $i \neq j$. $\square$

Next, we present a useful lemma that can be used to study the convergence of Algorithm 1.

Lemma 3. Assume that the generalized coupled Sylvester tensor Equation (1) is consistent. If it has solutions $\mathcal{X}_1^*, \dots, \mathcal{X}_n^*$, then

$$\begin{aligned}
& \langle \tilde{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{X}_1^* - \mathcal{X}_1^k \rangle + \cdots + \langle \tilde{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{X}_n^* - \mathcal{X}_n^k \rangle \\
& = \|\mathcal{R}_1^k\|^2 + \cdots + \|\mathcal{R}_n^k\|^2.
\end{aligned} \tag{16}$$

Moreover,

$$\begin{aligned}
& \langle \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{R}_1^k \rangle + \cdots + \langle \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{R}_n^k \rangle \\
& = \|\tilde{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \cdots + \|\tilde{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2.
\end{aligned} \tag{17}$$

Proof. It follows from Equation (13) that

$$\begin{aligned}
& \langle \tilde{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{X}_1^* - \mathcal{X}_1^k \rangle + \cdots + \langle \tilde{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k), \mathcal{X}_n^* - \mathcal{X}_n^k \rangle \\
& = \langle \mathcal{R}_1^k, \mathcal{L}_1(\mathcal{X}_1^* - \mathcal{X}_1^k, \dots, \mathcal{X}_n^* - \mathcal{X}_n^k) \rangle + \cdots + \langle \mathcal{R}_n^k, \mathcal{L}_n(\mathcal{X}_1^* - \mathcal{X}_1^k, \dots, \mathcal{X}_n^* - \mathcal{X}_n^k) \rangle \\
& = \langle \mathcal{R}_1^k, \mathcal{L}_1(\mathcal{X}_1^*, \dots, \mathcal{X}_n^*) - \mathcal{L}_1(\mathcal{X}_1^k, \dots, \mathcal{X}_n^k) \rangle \\
& \quad + \cdots + \langle \mathcal{R}_n^k, \mathcal{L}_n(\mathcal{X}_1^*, \dots, \mathcal{X}_n^*) - \mathcal{L}_n(\mathcal{X}_1^k, \dots, \mathcal{X}_n^k) \rangle \\
& = \langle \mathcal{R}_1^k, \mathcal{B}_1 - \mathcal{L}_1(\mathcal{X}_1^k, \dots, \mathcal{X}_n^k) \rangle + \cdots + \langle \mathcal{R}_n^k, \mathcal{B}_n - \mathcal{L}_n(\mathcal{X}_1^k, \dots, \mathcal{X}_n^k) \rangle \\
& = \|\mathcal{R}_1^k\|^2 + \cdots + \|\mathcal{R}_n^k\|^2.
\end{aligned}$$

For Equation (17), we have

$$\begin{aligned}
 & \langle \mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{R}_1^k \rangle + \dots + \langle \mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \mathcal{R}_n^k \rangle \\
 &= \langle \mathcal{P}_1^k, \bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \rangle + \dots + \langle \mathcal{P}_n^k, \bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \rangle \\
 &= \langle \bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) + \beta_{k-1} \mathcal{P}_1^{k-1}, \bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \rangle \\
 &+ \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) + \beta_{k-1} \mathcal{P}_n^{k-1}, \bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \rangle \\
 &= \|\bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 \\
 &+ \beta_{k-1} [\langle \mathcal{P}_1^{k-1}, \bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \rangle + \dots + \langle \mathcal{P}_n^{k-1}, \bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \rangle] \\
 &= \|\bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2.
 \end{aligned}$$

□

From Equation (16), we can obtain that $\mathcal{L}_l(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k) \neq \mathcal{O}$ if $\|\mathcal{R}_1^k\|^2 + \dots + \|\mathcal{R}_n^k\|^2 \neq 0$, ($l = 1, \dots, n$); otherwise, the tensor equations (1) are nonconsistent. Similarly, from Equation (17), if $\|\bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 \neq 0$, then $\mathcal{R}_l^k \neq \mathcal{O}$ and $\mathcal{L}_l(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k) \neq \mathcal{O}$, ($l = 1, \dots, n$). Overall, we have the following results.

Proposition 1. Let $\{\mathcal{R}_l^k\}, \{\mathcal{P}_l^k\}, \{\bar{\mathcal{L}}_l(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\}$ and $\{\mathcal{L}_l(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\}$ be the tensor sequences given by Algorithm 1, $l = 1, \dots, n, k = 0, 1, \dots$. Then

$$\|\mathcal{R}_1^k\|^2 + \dots + \|\mathcal{R}_n^k\|^2 = 0$$

if and only if

$$\|\bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 = 0.$$

According to the foregoing observations, we present the following theorem to show that Algorithm 1 converges to an exact solution at most finite steps in the absence round-off errors.

Theorem 2. Assume that the generalized coupled Sylvester tensor equations, Equation (1), are consistent. If the tensor sequences $\{\mathcal{R}_l^k\}, \{\mathcal{P}_l^k\}, \{\bar{\mathcal{L}}_l(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\}$ and $\{\mathcal{L}_l(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\}$ are given by Algorithm 1, $l = 1, \dots, n, k = 0, 1, \dots$, then we can obtain a solution for any initial tensors $\mathcal{X}_1^0, \dots, \mathcal{X}_n^0$ at most $nI_1I_2 \dots I_n + 1$ steps in the absence round-off errors.

Proof. If

$$\|\bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 = 0, \quad (k = 1, \dots, nI_1I_2 \dots I_n - 1),$$

then it follows from Proposition 1 that $\mathcal{R}_l^k = \mathcal{O}$, i.e., $\mathcal{X}_1^k, \dots, \mathcal{X}_n^k$ are solutions of Equation (1). If

$$\|\bar{\mathcal{L}}_1(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^k, \dots, \mathcal{R}_n^k)\|^2 \neq 0 \quad (k = 1, \dots, nI_1I_2 \dots I_n),$$

then we have $\mathcal{L}_l(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k) \neq \mathcal{O}$, ($l = 1, \dots, n$), i.e.,

$$\|\mathcal{L}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2 + \dots + \|\mathcal{L}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k)\|^2 \neq 0.$$

So, we can obtain $\bar{\mathcal{L}}_l(\mathcal{R}_1^{nI_1I_2 \dots I_n+1}, \dots, \mathcal{R}_n^{nI_1I_2 \dots I_n+1})$. Therefore, by Equation (14), it follows that

$$\begin{aligned}
 & \langle \bar{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i), \bar{\mathcal{L}}_1(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle + \dots + \langle \bar{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i), \bar{\mathcal{L}}_n(\mathcal{R}_1^j, \dots, \mathcal{R}_n^j) \rangle \\
 &= \begin{cases} \|\bar{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i)\|^2 + \dots + \|\bar{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i)\|^2, & j = i, \\ 0, & j \neq i. \end{cases} \quad (18)
 \end{aligned}$$

holds for $0 \leq j, i \leq nI_1 I_2 \cdots I_n + 1$. Let

$$\mathcal{M}^i = \begin{bmatrix} \tilde{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i)_{(1)} & & & \\ & \tilde{\mathcal{L}}_2(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i)_{(1)} & & \\ & & \ddots & \\ & & & \tilde{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i)_{(1)} \end{bmatrix},$$

where $\tilde{\mathcal{L}}_1(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i)_{(1)}, \dots, \tilde{\mathcal{L}}_n(\mathcal{R}_1^i, \dots, \mathcal{R}_n^i)_{(1)}$ denote the unfolding of themselves along mode 1. From Equation (18), we have

$$\langle \mathcal{M}^i, \mathcal{M}^j \rangle = \begin{cases} \|\mathcal{M}^i\|^2, & j = i, \\ 0, & j \neq i. \end{cases}$$

Hence, $\mathcal{M}^i (i = 1, \dots, nI_1 I_2 \cdots I_n + 1)$ is an orthogonal basis of matrix space

$$\mathcal{L} = \left\{ \mathcal{S} | \mathcal{S} = \begin{bmatrix} \mathcal{S}_1 & & & \\ & \mathcal{S}_2 & & \\ & & \ddots & \\ & & & \mathcal{S}_n \end{bmatrix} \right\},$$

where $\mathcal{S}_l \in \mathbb{R}^{I_1 \times I_2 \cdots I_n}$, $(l = 1, \dots, n)$. However, it is impossible for $\mathcal{M}^{nI_1 I_2 \cdots I_n + 1} \neq \mathcal{O}$ because the maximum dimension of $\mathcal{L}$ is $nI_1 I_2 \cdots I_n$. Using Proposition 1, it follows that $\mathcal{R}_l^{nI_1 I_2 \cdots I_n + 1} = \mathcal{O}$. This fact shows that $\mathcal{X}_1^{nI_1 I_2 \cdots I_n + 1}, \dots, \mathcal{X}_n^{nI_1 I_2 \cdots I_n + 1}$ are solutions of Equation (1). $\square$

Since the preconditioned technique is used to solve nonsingular linear systems, in the rest of this paper, without specification, we assume that Equation (1) always has a unique solution; i.e., the matrix A of Equation (8) is nonsingular. Now, we want to find a preconditioning matrix P to approximate A^{-1} such that

$$PAx = Pb \quad (19)$$

and Equation (8) has a same solution. Additionally, Equation (19) could be solved more effectively than the original systems.

The nearest Kronecker product (NKP) preconditioner, which is based on the Kronecker product approximations, was firstly described by Pitsianis and Van Loan in [44]. For a general matrix A , they derived a method for finding its nearest Kronecker product $M \otimes N$, which causes $P = M^{-1} \otimes N^{-1}$ to approximate A^{-1} sufficiently. Then, Langville and Stewart [45] extended the above results on matrices with special structure. The goal is to find matrices $Q_1, Q_2, \dots, Q_n$ such that

$$\|A - Q_1 \otimes Q_2 \otimes \cdots \otimes Q_n\|^2 \quad (20)$$

is minimized, where $A = \sum_{j=1}^m \otimes_{i=1}^n A_j^{(i)}$. From [45], we can see that

$$\begin{cases} Q_1 \approx a_{11}A_1^{(1)} + a_{12}A_2^{(1)} + a_{1m}A_m^{(1)}, \\ Q_2 \approx a_{21}A_1^{(2)} + a_{22}A_2^{(2)} + a_{2m}A_m^{(2)}, \\ \vdots \\ Q_n \approx a_{n1}A_1^{(n)} + a_{n2}A_2^{(n)} + a_{nm}A_m^{(n)}. \end{cases}$$

Similarly, we now want to find matrices $R, P_1, P_2, \dots, P_n$ such that

$$\|A - R \otimes P_1 \otimes P_2 \otimes \cdots \otimes P_n\|^2 \quad (21)$$

is minimized. Here, A is the same as in Equation (8). It is easy to see that the matrix A can be written as

$$A = R_{11} \otimes I^{I_n} \otimes \cdots \otimes I^{I_2} \otimes A_{11} + R_{12} \otimes I^{I_n} \otimes \cdots \otimes A_{12} \otimes I^{I_1} + \cdots + R_{1n} \otimes A_{1n} \otimes I^{I_{n-1}} \otimes \cdots \otimes I^{I_1} \\ + R_{21} \otimes A_{2n} \otimes I^{I_{n-1}} \otimes \cdots \otimes I^{I_1} + \cdots + R_{2n} \otimes I^{I_n} \otimes A_{2(n-1)} \otimes \cdots \otimes I^{I_1} + R_{n1} \otimes I^{I_n} \otimes \cdots \otimes A_{n2} \otimes I^{I_1} \\ + \cdots + R_{nn} \otimes I^{I_n} \otimes \cdots \otimes I^{I_2} \otimes A_{n1},$$

where

$$R_{ij} = j \begin{pmatrix} & i \\ \ddots & & \ddots \\ & 1 & \\ \ddots & & \ddots \end{pmatrix}$$

is an $n \times n$ matrix, $i, j = 1, \dots, n$. By Equation (20), it follows that

$$\begin{cases} R \approx r_{11}R_{11} + r_{12}R_{12} + \cdots + r_{1n}R_{1n} + r_{21}R_{21} + \cdots + r_{2n}R_{2n} + \cdots + r_{n1}R_{n1} + \cdots + r_{nn}R_{nn}, \\ P_1 \approx p_{10}I^{I_n} + p_{11}A_{1n} + p_{12}A_{2n} + \cdots + p_{1n}A_{nn}, \\ P_2 \approx p_{20}I^{I_{n-1}} + p_{21}A_{1(n-1)} + p_{22}A_{2(n-1)} + \cdots + p_{2n}A_{n(n-1)}, \\ \vdots \\ P_n \approx p_{n0}I^{I_1} + p_{n1}A_{11} + p_{n2}A_{21} + \cdots + p_{nn}A_{n1}, \end{cases} \quad (22)$$

where r_{ij}, p_{ik} are unknown parameters, $i, j = 1, \dots, n, k = 0, 1, \dots, n$. Moreover, in order to reduce the complexity of finding the inverse of R , we only take $R \approx r_{11}R_{11} + r_{22}R_{22} + \cdots + r_{nn}R_{nn}$, i.e.,

$$R \approx \begin{bmatrix} r_{11} & & & \\ & r_{22} & & \\ & & \ddots & \\ & & & r_{nn} \end{bmatrix}.$$

As stated in [24,45], the optimal parameters then can be solved by the nonlinear optimization software quickly, such as `fminsearch` in MATLAB, `nlp` in SAS, multilevel coordinate search. So, we have $A \approx R \otimes P_1 \otimes P_2 \otimes \cdots \otimes P_n$, and take

$$P = (R \otimes P_1 \otimes P_2 \otimes \cdots \otimes P_n)^{-1} = R^{-1} \otimes P_1^{-1} \otimes P_2^{-1} \otimes \cdots \otimes P_n^{-1} \\ = \begin{bmatrix} r_{11}^{-1}P_1^{-1} \otimes P_2^{-1} \otimes \cdots \otimes P_n^{-1} & & & \\ & r_{22}^{-1}P_1^{-1} \otimes P_2^{-1} \otimes \cdots \otimes P_n^{-1} & & \\ & & \ddots & \\ & & & r_{nn}^{-1}P_1^{-1} \otimes P_2^{-1} \otimes \cdots \otimes P_n^{-1} \end{bmatrix} \quad (23)$$

as a preconditioner for solving Equation (8). Obviously, according to Equation (23) and the fact $(A \otimes B)(C \otimes D) = (AC \otimes BD)$, the tensor equation, Equation (1), can be rewritten as

$$\begin{cases} \mathcal{X}_1 \times_1 P_n^{-1} A_{11} \times_2 \cdots \times_n P_1^{-1} + \mathcal{X}_2 \times_1 P_n^{-1} \times_2 P_{n-1}^{-1} A_{12} \times_3 \cdots \times_n P_1^{-1} \\ + \cdots + \mathcal{X}_n \times_1 P_n^{-1} \times_2 \cdots \times_n P_1^{-1} A_{1n} = \mathcal{B}_1 \times_1 P_n^{-1} \times_2 \cdots \times_n P_1^{-1}, \\ \mathcal{X}_2 \times_1 P_n^{-1} A_{21} \times_2 \cdots \times_n P_1^{-1} + \mathcal{X}_3 \times_1 P_n^{-1} \times_2 P_{n-1}^{-1} A_{22} \times_3 \cdots \times_n P_1^{-1} \\ + \cdots + \mathcal{X}_1 \times_1 P_n^{-1} \times_2 \cdots \times_n P_1^{-1} A_{2n} = \mathcal{B}_2 \times_1 P_n^{-1} \times_2 \cdots \times_n P_1^{-1}, \\ \vdots \\ \mathcal{X}_n \times_1 P_n^{-1} A_{n1} \times_2 \cdots \times_n P_1^{-1} + \mathcal{X}_1 \times_1 P_n^{-1} \times_2 P_{n-1}^{-1} A_{n2} \times_3 \cdots \times_n P_1^{-1} \\ + \cdots + \mathcal{X}_{n-1} \times_1 P_n^{-1} \times_2 \cdots \times_n P_1^{-1} A_{nn} = \mathcal{B}_n \times_1 P_n^{-1} \times_2 \cdots \times_n P_1^{-1}. \end{cases} \quad (24)$$

From the above discussion, we propose a preconditioned modified conjugate residual (PMCR) method for solving Equation (1) (Algorithm 2).

Algorithm 2 A preconditioned modified conjugate residual method for solving Equation (1)

1. Given initial values $\mathcal{X}_i^0, \mathcal{B}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$, $A_{ij} \in \mathbb{R}^{I_j \times I_j}$, $k = 0$, and set $\epsilon > 0$, $i, j = 1, \dots, n$.
2. Compute $\bar{\mathcal{B}}_i = \mathcal{B}_i \times_1 P_n^{-1} \times_2 \dots \times_n P_1^{-1}$, and set

$$\begin{cases} \mathcal{M}_1(\mathcal{X}_1^0, \dots, \mathcal{X}_n^0) = \mathcal{X}_1^0 \times_1 P_n^{-1} A_{11} \times_2 \dots \times_n P_1^{-1} + \dots + \mathcal{X}_n^0 \times_1 P_n^{-1} \times_2 \dots \times_n P_1^{-1} A_{1n}, \\ \mathcal{M}_2(\mathcal{X}_1^0, \dots, \mathcal{X}_n^0) = \mathcal{X}_2^0 \times_1 P_n^{-1} A_{21} \times_2 \dots \times_n P_1^{-1} + \dots + \mathcal{X}_1^0 \times_1 P_n^{-1} \times_2 \dots \times_n P_1^{-1} A_{2n}, \\ \vdots \\ \mathcal{M}_n(\mathcal{X}_1^0, \dots, \mathcal{X}_n^0) = \mathcal{X}_n^0 \times_1 P_n^{-1} A_{n1} \times_2 \dots \times_n P_1^{-1} + \dots + \mathcal{X}_{n-1}^0 \times_1 P_n^{-1} \times_2 \dots \times_n P_1^{-1} A_{nn}, \\ \overline{\mathcal{M}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) = \mathcal{R}_1^0 \times_1 (P_n^{-1} A_{11})^T \times_2 \dots \times_n (P_1^{-1})^T + \dots + \mathcal{R}_n^0 \times_1 (P_n^{-1})^T \times_2 \dots \times_n (P_1^{-1} A_{2n})^T, \\ \overline{\mathcal{M}}_2(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) = \mathcal{R}_1^0 \times_1 (P_n^{-1})^T \times_2 (P_{n-1}^{-1} A_{12})^T \times_3 \dots \times_n (P_1^{-1})^T + \dots \\ + \mathcal{R}_n^0 \times_1 \dots \times_3 (P_{n-2}^{-1} A_{n3})^T \times_4 \dots \times_n (P_1^{-1})^T, \\ \vdots \\ \overline{\mathcal{M}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0) = \mathcal{R}_1^0 \times_1 (P_n^{-1})^T \times_2 \dots \times_n (P_1^{-1} A_{1n})^T + \dots + \mathcal{R}_n^0 \times_1 (P_n^{-1} A_{n1})^T \times_2 \dots \times_n (P_1^{-1})^T, \end{cases}$$

3. Define $\mathcal{R}_i^0 = \bar{\mathcal{B}}_i - \mathcal{M}_i(\mathcal{X}_1^0, \dots, \mathcal{X}_n^0)$, $\mathcal{P}_i^0 = \overline{\mathcal{M}}_i(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)$, and compute

$$\begin{aligned} s_0 &= \|\overline{\mathcal{M}}_1(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2 + \dots + \|\overline{\mathcal{M}}_n(\mathcal{R}_1^0, \dots, \mathcal{R}_n^0)\|^2, \\ p_0 &= \|\mathcal{M}_1(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2 + \dots + \|\mathcal{M}_n(\mathcal{P}_1^0, \dots, \mathcal{P}_n^0)\|^2, \\ \eta_k &= \|\mathcal{R}_1^k\|^2 + \dots + \|\mathcal{R}_n^k\|^2. \end{aligned}$$

4. For $k = 1, 2, \dots$ until $\eta_k < \epsilon$ do:

$$\begin{cases} \mathcal{R}_1^{k+1} = \mathcal{R}_1^k - \alpha_k \mathcal{M}_1(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \\ \mathcal{R}_2^{k+1} = \mathcal{R}_2^k - \alpha_k \mathcal{M}_2(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \\ \vdots \\ \mathcal{R}_n^{k+1} = \mathcal{R}_n^k - \alpha_k \mathcal{M}_n(\mathcal{P}_1^k, \dots, \mathcal{P}_n^k), \\ \overline{\mathcal{M}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_1 (P_n^{-1} A_{11})^T \times_2 \dots \times_n (P_1^{-1})^T + \dots \\ + \mathcal{R}_n^{k+1} \times_1 (P_n^{-1})^T \times_2 \dots \times_n (P_1^{-1} A_{2n})^T, \\ \overline{\mathcal{M}}_2(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_1 (P_n^{-1})^T \times_2 (P_{n-1}^{-1} A_{12})^T \times_3 \dots \times_n (P_1^{-1})^T + \dots \\ + \mathcal{R}_n^{k+1} \times_1 \dots \times_3 (P_{n-2}^{-1} A_{n3})^T \times_4 \dots \times_n (P_1^{-1})^T, \\ \vdots \\ \overline{\mathcal{M}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) = \mathcal{R}_1^{k+1} \times_1 (P_n^{-1})^T \times_2 \dots \times_n (P_1^{-1} A_{1n})^T + \dots \\ + \mathcal{R}_n^{k+1} \times_1 (P_n^{-1} A_{n1})^T \times_2 \dots \times_n (P_1^{-1})^T, \\ s_{k+1} = \|\overline{\mathcal{M}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1})\|^2 + \dots + \|\overline{\mathcal{M}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1})\|^2. \end{cases}$$

5. Calculate $\beta_k = \frac{s_{k+1}}{s_k}$,

$$\begin{cases} \mathcal{P}_1^{k+1} = \overline{\mathcal{M}}_1(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{P}_1^k, \\ \mathcal{P}_2^{k+1} = \overline{\mathcal{M}}_2(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{P}_2^k, \\ \vdots \\ \mathcal{P}_n^{k+1} = \overline{\mathcal{M}}_n(\mathcal{R}_1^{k+1}, \dots, \mathcal{R}_n^{k+1}) + \beta_k \mathcal{P}_n^k, \\ p_{k+1} = \|\mathcal{M}_1(\mathcal{P}_1^{k+1}, \dots, \mathcal{P}_n^{k+1})\|^2 + \dots + \|\mathcal{M}_n(\mathcal{P}_1^{k+1}, \dots, \mathcal{P}_n^{k+1})\|^2. \end{cases}$$

4. Numerical Experiments

In this section, we give some numerical experiments to illustrate the effectiveness of the algorithms proposed in here. All codes were written in Matlab 2016b and run on a PC with Inter(R) Core(TM) i7-9750H @2.6 GHz and 8.00 GB memory. Additionally, all operations were based on tensor toolbox (version 2.5) proposed by Bader and Kolda [49].

We compare the MCR and PMCR algorithms with MCG [40] in terms of the number of iteration steps (“IT”), the elapsed CPU time (“CPU”) in seconds and the residual Frobenius norm (“RES”) defined as η_k . In all of the following examples, we take $\mathcal{X} = \mathcal{O}$ as initial tensors, and $\eta_k < 10^{-16}$ or the number of iteration steps exceeding 5000 as the stopping criteria.

Example 1. Consider the third-order Sylvester tensor equation, Equation (2), with its coefficient matrices being the form of Equation (4). Such matrices are scaled by the same parameter $\frac{1}{I_x I_y I_z}$, and $\mathcal{B} = \text{tenrand}(I_x, I_y, I_z)$. From [15], we let $\tau_q = \frac{1}{\pi^2} + 10^3$, $\tau_U = \frac{1}{\pi^2} - 1.99 \times 10^{-5}$, $\alpha = \frac{\rho_{C_p}}{\kappa} = 1$, $\Delta t = 10^{-3}$, $\delta_x = \delta_y = 0.1$ and $\delta_z = 10^{-5}$.

In this case, we implement the proposed algorithms and MCG for finding the iterative solution of Equation (2), and reported the numerical results in Table 1. From this Table, we can observe that our algorithms in here are feasible and effective, and the elapsed CPU time corresponding to MCR algorithm is slightly less than MCG. The number of iteration steps to MCR algorithm is no more than $I_x I_y I_z + 1$, which coincides with the results of Theorem 2. It is also shown that the PMCR algorithm outperforms other algorithms, and its convergence rate is about four times that of the MCG method. Additionally, from Figure 1, we can see that the convergence behavior of MCR is more stable than that of MCG.

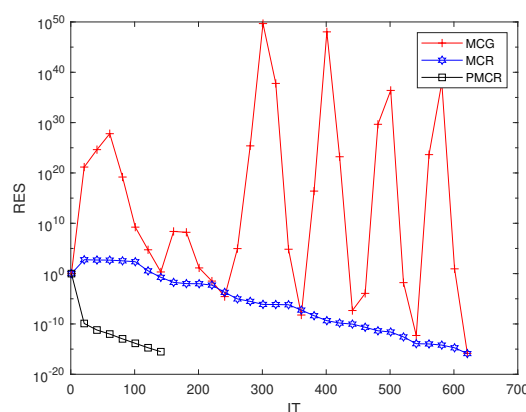


Figure 1. Convergence history of Example 1, $I_1 = 15$, $I_2 = 15$, $I_3 = 20$.

Example 2. As the second example, we consider the following convection-diffusion equation [9,16,25,29]

$$\begin{aligned} -v\Delta u + c^T \nabla u &= f \text{ in } \Gamma = [0,1]^n, \\ u &= 0 \text{ on } \partial\Gamma. \end{aligned} \quad (25)$$

Table 1. Comparison results of Example 1.

Algorithms [I_x, I_y, I_z]	MCG [40]			MCR			PMCR		
	IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
[5, 5, 5]	15	0.1463	5.9784×10^{-18}	14	0.1396	3.6714×10^{-18}	5	0.0313	6.2438×10^{-19}
[10, 10, 15]	301	0.9928	9.6023×10^{-17}	296	0.9434	8.8758×10^{-17}	62	0.2183	6.2891×10^{-17}
[15, 15, 20]	632	2.7739	9.6572×10^{-17}	624	2.6599	9.5403×10^{-17}	149	0.5866	8.6559×10^{-17}
[15, 20, 25]	1189	8.0489	6.4185×10^{-17}	1152	7.8583	4.5580×10^{-17}	359	2.2154	1.5868×10^{-17}
[20, 20, 30]	1696	10.2321	9.2562×10^{-17}	1648	9.7157	9.1825×10^{-17}	478	3.0135	4.8919×10^{-17}

From the standard finite difference discretization on equidistant nodes and a second-order convergent scheme (Fromm's scheme), Equation (25) can be discretized as Equation (5) with

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

where $h = \frac{1}{m+1}$ is the mesh size, $i = 1, \dots, n$. Set $n = 5, m = 10$, and

- (1). $v = 100, c_1 = 1, c_2 = 1, c_3 = 1, c_4 = 1, c_5 = 1$,
- (2). $v = 100, c_1 = 10, c_2 = 30, c_3 = 50, c_4 = 70, c_5 = 90$,
- (3). $v = 1, c_1 = 1, c_2 = 1, c_3 = 1, c_4 = 1, c_5 = 1$,
- (4). $v = 1, c_1 = 10, c_2 = 30, c_3 = 50, c_4 = 70, c_5 = 90$,
- (5). $v = 0.01, c_1 = 1, c_2 = 1, c_3 = 1, c_4 = 1, c_5 = 1$,
- (6). $v = 0.01, c_1 = 10, c_2 = 30, c_3 = 50, c_4 = 70, c_5 = 90$,

and $\mathcal{B} = \text{tenrand}(10, 10, 10, 10, 10)$.

In this example, we utilize the proposed algorithms and MCG to solve Equation (5). The obtained numerical results are reported in Table 2. From this table, one can see that the number of iteration steps to MCR is less than $10^5 + 1$. On the other hand, the performance of MCR is slightly better than MCG in terms of the elapsed CPU time. It is also shown that PMCR algorithm converges faster than that of MCG and MCR.

Table 2. Comparison results of Example 2.

Algorithms	MCG [40]			MCR			PMCR		
	IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
(1)	232	3.1112	9.7305×10^{-17}	226	3.0947	8.6237×10^{-17}	58	0.8934	8.1465×10^{-17}
(2)	321	4.2594	8.8469×10^{-17}	299	3.9742	8.9632×10^{-17}	66	1.0423	7.7125×10^{-17}
(3)	283	3.9432	9.9765×10^{-17}	278	3.9099	9.3321×10^{-17}	63	0.9578	9.0356×10^{-17}
(4)	400	5.2723	8.9944×10^{-17}	382	5.2150	9.1298×10^{-17}	82	2.1283	7.9046×10^{-17}
(5)	376	5.0942	9.4951×10^{-17}	358	4.9285	9.8407×10^{-17}	79	1.9201	8.9957×10^{-17}
(6)	487	6.0027	9.3902×10^{-17}	462	5.9755	9.4925×10^{-17}	124	2.4562	9.1929×10^{-17}

Example 3. Consider the following generalized coupled Sylvester tensor equations

$$\begin{cases} \mathcal{X}_1 \times_1 A_{11} + \mathcal{X}_2 \times_2 A_{12} + \mathcal{X}_3 \times_3 A_{13} = \mathcal{B}_1, \\ \mathcal{X}_2 \times_1 A_{21} + \mathcal{X}_3 \times_2 A_{22} + \mathcal{X}_1 \times_3 A_{23} = \mathcal{B}_2, \\ \mathcal{X}_3 \times_1 A_{31} + \mathcal{X}_1 \times_2 A_{32} + \mathcal{X}_2 \times_3 A_{33} = \mathcal{B}_3, \end{cases} \quad (26)$$

where A_{11} and A_{22} are the matrices A_1 and A_2 of Equation (4), respectively, and the remaining matrices and tensors are given by

$\alpha = 100$;

$A_{12} = \text{rand}(I_2, I_2) + \text{diag}(\text{ones}(I_2, 1)) * \alpha$, $A_{13} = \text{rand}(I_3, I_3) + \text{diag}(\text{ones}(I_3, 1)) * \alpha$;

$A_{21} = \text{rand}(I_1, I_1) + \text{diag}(\text{ones}(I_1, 1)) * \alpha$, $A_{23} = \text{rand}(I_3, I_3) + \text{diag}(\text{ones}(I_3, 1)) * \alpha$;

$A_{31} = \text{rand}(I_1, I_1) + \text{diag}(\text{ones}(I_1, 1)) * \alpha$, $A_{32} = \text{rand}(I_2, I_2) + \text{diag}(\text{ones}(I_2, 1)) * \alpha$;

$A_{33} = \text{rand}(I_3, I_3) + \text{diag}(\text{ones}(I_3, 1)) * \alpha$, $\mathcal{B}_i = \text{tenrand}(I_1, I_2, I_3)$, $i = 1, 2, 3$.

Our tested experimental results in Table 3 show that the elapsed CPU time to all of the algorithms are increasing as the dimension increases, and the number of iteration steps to MCR algorithm does not exceed $I_1 I_2 I_3 + 1$. It also reflects that the elapsed CPU time to MCR with nearest

Kronecker product preconditioner is much lower than that of MCR and MCG. From Figure 2, one can see that the MCG algorithm has a quiet irregular convergence behavior but MCR is rather stable. Additionally, we can see that the MCR algorithm is less than MCG in terms of the elapsed CPU time.

Table 3. Comparison results of Example 3.

Algorithms [I_1, I_2, I_3]	MCG [40]			MCR			PMCR		
	IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
[5, 10, 15]	76	0.5844	3.1415×10^{-17}	74	0.5179	4.3350×10^{-17}	20	0.1248	2.5513×10^{-17}
[10, 15, 20]	465	3.4385	8.1439×10^{-17}	442	3.1806	6.1823×10^{-17}	102	0.6026	7.6829×10^{-17}
[15, 15, 15]	130	0.9715	1.1913×10^{-17}	128	0.9237	1.0795×10^{-17}	42	0.2556	6.4513×10^{-17}
[20, 25, 30]	1310	9.5945	9.1953×10^{-17}	1289	9.1217	8.3788×10^{-17}	363	2.0159	7.2917×10^{-17}
[25, 25, 25]	1255	9.0559	8.2675×10^{-17}	1210	8.8567	7.4489×10^{-17}	325	1.8219	7.6519×10^{-17}

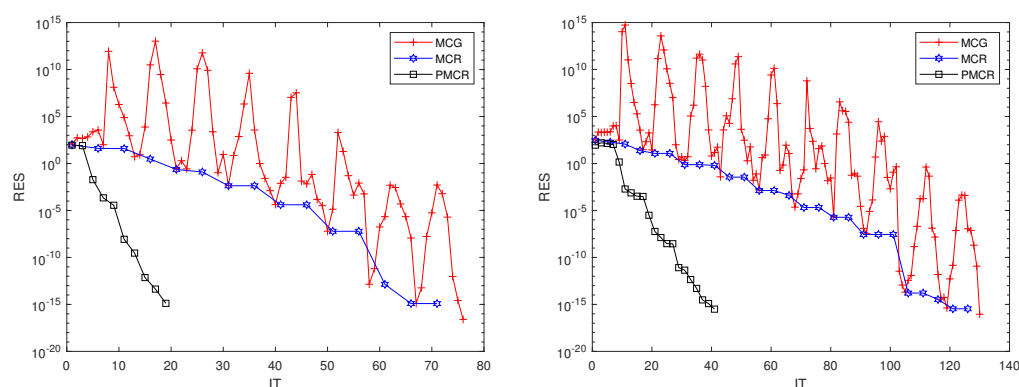


Figure 2. Convergence history of Example 3: $I_1 = 5, I_2 = 10, I_3 = 15$ (left), $I_1 = 15, I_2 = 15$, and $I_3 = 15$ (right).

5. Conclusions

In this paper, we proposed a modified conjugate residual algorithm and its preconditioned version based on the Kronecker product approximations for solving the generalized coupled Sylvester tensor Equation (1). Our convergence analysis showed that the MCR algorithm is convergent to an exact solution for any initial tensors at most finite steps in the absence round-off errors, and its convergence behavior is more smooth than MCG for solving some problems. From the Kronecker product approximations, we developed the PMCR method for solving the original tensor Equation (1). The performance of PMCR outperforms other algorithms in terms of the elapsed CPU time and the number of iteration steps. Note that the nearest Kronecker product technique can also be applied to some existing methods, such as the biconjugate gradient method and biconjugate residual method. So, the preconditioned technique is very promising.

Author Contributions: All authors have equal contributions for Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing an original draft, Writing a review, and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by National Natural Science Foundation of China [grant numbers 11971294 and 12171369], and Hainan Provincial Natural Science Foundation of China [grant numbers 122QN214 and 122MS001].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous for their valuable suggestions, comments and National Natural Science Foundation of China [grant numbers 11971294 and

12171369], and Hainan Provincial Natural Science Foundation of China [grant numbers 122QN214 and 122MS001].

Conflicts of Interest: The authors declare no conflict of interest.

References

- Weaver, J. Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues, and eigenvectors. *Am. Math. Month.* **1985**, *92*, 711–717. [\[CrossRef\]](#)
- Datta, L.; Morgera, S. Some results on matrix symmetries and a pattern recognition application. *IEEE Trans. Signal Process.* **1986**, *34*, 992–994. [\[CrossRef\]](#)
- Datta, L.; Morgera, S. On the reducibility of centrosymmetric matrices-applications in engineering problems. *Circ. Syst. Signal Process.* **1989**, *8*, 71–96. [\[CrossRef\]](#)
- Datta, B.N.; Lin, W.; Wang, J.N. Robust partial pole assignment for vibrating systems with aerodynamic effects. *IEEE Trans. Automat. Control* **2006**, *51*, 1979–1984. [\[CrossRef\]](#)
- Liang, M.L.; Zheng, B.; Zhao, R.J. Alternating iterative methods for solving tensor equations with applications. *Numer. Algorithms* **2018**, *80*, 1437–1465. [\[CrossRef\]](#)
- Li, Q.; Zheng, B. Scaled three-term derivative-free methods for solving large-scale nonlinear monotone equations. *Numer. Algorithms* **2021**, *87*, 1343–1367. [\[CrossRef\]](#)
- Zhang, F.X.; Wei, M.S.; Li, Y.; Zhao, J. An efficient method for least-squares problem of the quaternion matrix equation. *Linear Multilinear Algebra* **2020**, 1–13. [\[CrossRef\]](#)
- Zhang, F.X.; Wei, M.S.; Li, Y.; Zhao, J. An efficient real representation method for least squares problem of the quaternion constrained matrix equation $AXB + CYD = E$. *Int. J. Comput. Math.* **2021**, *98*, 1408–1419. [\[CrossRef\]](#)
- Grasedyck, L. Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure. *Computing* **2004**, *72*, 247–265. [\[CrossRef\]](#)
- Bai, Z.Z.; Golub, G.H.; Ng, M.K. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM J. Matrix Anal. Appl.* **2003**, *24*, 603–626. [\[CrossRef\]](#)
- Li, X.T.; Ng, M.K. Solving sparse non-negative tensor equations: algorithms and applications. *Front. Math. China* **2015**, *10*, 649–680. [\[CrossRef\]](#)
- Li, B.W.; Sun, Y.S.; Yu, Y. Iterative and direct Chebyshev collocation spectral methods for one-dimensional radiative heat transfer. *Int. J. Heat Mass Transf.* **2008**, *51*, 5887–5894. [\[CrossRef\]](#)
- Sun, Y.S.; Ma, J.; Li, B.W. Chebyshev collocation spectral method for three-dimensional transient coupled radiative conductive heat transfer. *J. Heat Transfer* **2012**, *134*, 092701. [\[CrossRef\]](#)
- Li, B.W.; Tian, S.; Sun, Y.S.; Hu, Z.M. Schur-decomposition for 3D matrix equations and its application in solving radiative discrete ordinates equations discretized by Chebyshev collocation spectral method. *J. Comput. Phys.* **2010**, *229*, 1198–1212. [\[CrossRef\]](#)
- Malek, A.; Momeni-Masuleh, S.H. A mixed collocation-finite difference method for 3D microscopic heat transport problem. *J. Comput. Appl. Math.* **2008**, *217*, 137–147. [\[CrossRef\]](#)
- Chen, Z.; Lu, L.Z. A gradient based iterative solutions for Sylvester tensor equations. *Math. Probl. Eng.* **2013**, *2013*, 819479. [\[CrossRef\]](#)
- Beik, F.P.A.; Asl, S.A. Residual norm steepest descent based iterative algorithms for Sylvester tensor equations. *J. Math. Model.* **2015**, *2*, 115–131.
- Li, N.; Navasca, C.; Glenn, C. Iterative methods for symmetric outer product tensor decomposition. *Electron. Trans. Numer. Anal.* **2015**, *44*, 124–139.
- Bentbib, A.H.; Guide, M.E.; Jbilou, K.; Onunwor, E.; Reichel, L. Solution methods for linear discrete ill-posed problems for color image restoration. *BIT Numer. Math.* **2018**, *58*, 555–578. [\[CrossRef\]](#)
- Li, F.; Ng, M.K.; Plemmons, R.J. Coupled segmentation and denoising/deblurring for hyperspectral material identification. *Numer. Linear Algebra Appl.* **2012**, *19*, 153–173. [\[CrossRef\]](#)
- Hajarian, M. Symmetric solutions of the coupled generalized Sylvester matrix equations via BCR algorithm. *J. Franklin Inst.* **2016**, *353*, 3233–3248. [\[CrossRef\]](#)
- Hajarian, M. Computing symmetric solutions of general Sylvester matrix equations via Lanczos version of biconjugate residual algorithm. *Comput. Math. Appl.* **2018**, *76*, 686–700. [\[CrossRef\]](#)
- Wang, W.; Ding, F.; Dai, J. Maximum likelihood least squares identification for systems with autoregressive moving average noise. *Appl. Math. Model.* **2012**, *36*, 1842–1853. [\[CrossRef\]](#)
- Chen, Z.; Lu, L.Z. A projection method and Kronecker product preconditioner for solving Sylvester tensor equations. *Sci. China Math.* **2012**, *55*, 1281–1292. [\[CrossRef\]](#)
- Beik, F.P.A.; Movahed, F.S.; Ahmadi-Asl, S. On the Krylov subspace methods based on tensor format for positive definite Sylvester tensor equations. *Linear Algebra Appl.* **2016**, *23*, 444–466. [\[CrossRef\]](#)
- Karimi, S.; Dehghan, M. Global least squares method based on tensor form to solve linear systems in Kronecker format. *Trans. Inst. Meas. Control* **2018**, *40*, 2378–2386. [\[CrossRef\]](#)

27. Toutounian, F.; Karimi, S. Global least squares method (GILSQR) for solving general linear systems with several right-hand sides. *Appl. Math. Comput.* **2006**, *178*, 452–460.
28. Wang, Q.W.; Xu, X.X.; Duan, X.F. Least squares solution of the quaternion Sylvester tensor equation. *Linear Multilinear Algebra* **2021**, *69*, 104–130. [[CrossRef](#)]
29. Xu, X.J.; Wang, Q.W. Extending BiCG and BiCR methods to solve the Stein tensor equation. *Comput. Math. Appl.* **2019**, *77*, 3117–3127. [[CrossRef](#)]
30. Li, T.; Wang, Q.W.; Duan, X.F. Numerical algorithms for solving discrete Lyapunov tensor equation. *J. Comput. Appl. Math.* **2020**, *370*, 112676. [[CrossRef](#)]
31. Huang, B.H.; Li, W. Numerical subspace algorithms for solving the tensor equations involving Einstein product. *Numer. Linear Algebra Appl.* **2021**, *28*, e2351. [[CrossRef](#)]
32. Hajarian, M. Convergence properties of BCR method for generalized Sylvester matrix equation over generalized reflexive and anti-reflexive matrices. *Linear Multilinear Algebra* **2017**, *66*, 1975–1990. [[CrossRef](#)]
33. Hajarian, M. Reflexive periodic solutions of general periodic matrix equations. *Math. Methods Appl. Sci.* **2019**, *42*, 3527–3548. [[CrossRef](#)]
34. Hajarian, M. Conjugate gradient-like methods for solving general tensor equation with Einstein product. *J. Franklin. Inst.* **2020**, *357*, 4272–4285. [[CrossRef](#)]
35. Najafi-Kalyani, M.; Beik, F.P.A.; Jbilou, K. On global iterative schemes based on Hessenberg process for (ill-posed) Sylvester tensor equations. *J. Comput. Appl. Math.* **2020**, *373*, 112216. [[CrossRef](#)]
36. Liu, L.S.; Wang, Q.W.; Chen, J.F.; Xie, Y.Z. An exact solution to a quaternion matrix equation with an application. *Symmetry* **2022**, *14*, 375. [[CrossRef](#)]
37. Mehany, M.S.; Wang, Q.W. Three symmetrical systems of coupled Sylvester-like quaternion matrix equations. *Symmetry* **2022**, *14*, 350. [[CrossRef](#)]
38. Bentbib, A.H.; El-Halouy, S.; Sadek, E.M. Krylov subspace projection method for Sylvester tensor equation with low rank right-hand side. *Numer. Algorithms* **2020**, *84*, 1411–1430. [[CrossRef](#)]
39. Heyouni, M.; Saberi-Movahed, F.; Tajaddini, A. A tensor format for the generalized Hessenberg method for solving Sylvester tensor equations. *J. Comput. Appl. Math.* **2020**, *377*, 112878. [[CrossRef](#)]
40. Lv, C.Q.; Ma, C.F. A modified CG algorithm for solving generalized coupled Sylvester tensor equations. *Appl. Math. Comput.* **2020**, *365*, 124699. [[CrossRef](#)]
41. Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2003; Volume 82.
42. Saad, Y. ILUT: A dual threshold incomplete LU factorization. *Numer. Linear Algebra Appl.* **1994**, *1*, 387–402. [[CrossRef](#)]
43. Tallec, P.L.; Roeck, Y.H.D.; Vidrascu, M. Domain decomposition methods for large linearly elliptic three-dimensional problems. *J. Comput. Appl. Math.* **1991**, *34*, 93–117. [[CrossRef](#)]
44. Loan, C.F.V.; Pitsianis, N. Approximation with Kronecker products. In *Linear Algebra for Large Scale and Real-Time Applications (Leuven, 1992)*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1993; pp. 293–314.
45. Langville, A.N.; Stewart, W.J. A Kronecker product approximate preconditioner for SANs. *Numer. Linear Algebra Appl.* **2004**, *11*, 723–752. [[CrossRef](#)]
46. Qi, L.Q.; Luo, Z.Y. *Tensor Analysis: Spectral Theory and Special Tensors*; SIAM: Philadelphia, PA, USA, 2017.
47. Qi, L.Q.; Chen, H.; Chen, Y. *Tensor Eigenvalues and Their Applications*; Springer: Berlin/Heidelberg, Germany, 2018.
48. Kolda, T.; Bader, B. The TOPHITS model for higher-order web link analysis, Workshop on Link Analysis, Counterterrorism Security. *Minnesota* **2006**, *7*, 26–29.
49. Kolda, T.; Bader, B. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [[CrossRef](#)]