Iranian Journal of Numerical Analysis and Optimization
Vol. 14, No. 3, 2024, pp 938–969
https://doi.org/10.22067/ijnao.2024.87481.1418
https://ijnao.um.ac.ir/

Research Article

Extending quasi-GMRES method to solve generalized Sylvester tensor equations via the Einstein product

M.M. Izadkhah*,

Abstract

This paper aims to extend a Krylov subspace technique based on an incomplete orthogonalization of Krylov tensors (as a multidimensional extension of the common Krylov vectors) to solve generalized Sylvester tensor equations via the Einstein product. First, we obtain the tensor form of the quasi-GMRES method, and then we lead to the direct variant of the proposed algorithm. This approach has the great advantage that it uses previous data in each iteration and has a low computational cost. Moreover, an upper bound for the residual norm of the approximate solution is found. Finally, several experimental problems are given to show the acceptable accuracy and efficiency of the presented method.

AMS subject classifications (2020): 15A69,6 5F08, 65F10.

---

*Corresponding author

Received 04 Aprail 2024; revised 01 June 2024; accepted 16 June 2024

Mohammad Mahdi Izadkhah
Department of Computer Science, Faculty of Computer and Industrial Engineering, Birjand University of Technology, Birjand, Iran. e-mail: izadkhah@birjandut.ac.ir

How to cite this article
Izadkhah, M.M., Extending quasi-GMRES method to solve generalized Sylvester tensor equations via the Einstein product. *Iran. J. Numer. Anal. Optim.*, 2024; 14(3): 938-969. https://doi.org/10.22067/ijnao.2024.87481.1418

Keywords: Generalized Sylvester tensor equations; Einstein product; Quasi-GMRES method; Convergence analysis.

1 Introduction

As a common notation in the research literature, tensors are written in calligraphic font, for example, $\mathcal{A}$. For a positive integer N, an Nth order tensor (in some literature N-mode tensor, e.g., [6]) $\mathcal{A} = (a_{i_1\cdots i_N})(1 \le i_j \le I_j, j = 1,\ldots,N)$ is a multidimensional N-way array with I $(I = I_1 I_2 \cdots I_N)$ entries [25]. Let $\mathbb{R}^{I_1\times\cdots\times I_N}$ be the set of Nth order tensors of size $I_1 \times \cdots \times I_N$ over the real field $\mathbb{R}$. The tensor $\mathcal{O} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ with all entries zero denotes the zero tensor.

In this paper, we suggest an efficient iterative method to solve the generalized Sylvester tensor equation

$$\mathcal{A} \star_N \mathcal{X} \star_M \mathcal{B} + \mathcal{C} \star_N \mathcal{X} \star_M \mathcal{D} = \mathcal{F}, \tag{1}$$

where $\mathcal{A},\mathcal{C} \in \mathbb{R}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$, $\mathcal{B},\mathcal{D} \in \mathbb{R}^{K_1\times\cdots\times K_M\times K_1\times\cdots\times K_M}$, $\mathcal{F} \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$ are known tensors, and $\mathcal{X} \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$ is an unknown tenor to be determined. We denote the Einstein product by $\star_N$, which will be described in detail in Section 2.

Tensor equations arise from various fields of science and engineering multidimensional applications, including signal processing, data mining, thermal radiation, information retrieval, and three-dimensional (3D) microscopic heat transfer problems in heat transfer, and so many other modern applications in machine learning [31, 32, 33, 34, 39, 44].

Tensor equations involving the Einstein product have been studied in [7, 15, 38], which have many applications in continuum physics, engineering, isotropic, and anisotropic elastic models [26]. For example, Wang and Xu [41] introduced some iterative methods for solving different types of these tensor equations. Huang, Xie, and Ma [23] proposed the Krylov subspace methods to solve a class of tensor equations via the Einstein product. Huang and Ma [22] presented an iterative algorithm to solve the generalized Sylvester tensor equation. In [21], they also presented the global least squares methods based on tensor form to solve the tensor equation (1). Liang, Zheng, and Zhao [30] discussed the tensor inversion and its applications for solving the tensor equations via the Einstein product.

The high order Sylvester tensor equation via the Tucker product of tensors is as follows:

$$\mathcal{X} \times_1 A_1 + \mathcal{X} \times_2 A_2 + \cdots + \mathcal{X} \times_N A_N = \mathcal{D}, \tag{2}$$

where $A_j \in \mathbb{R}^{I_j \times I_j}$, $j = 1, 2, \ldots, N$, $\mathcal{D} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ are known, and $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is unknown. The product $\times_k$ will be defined in the next Section.

Recently, Li, Wang, and Zhang [29] proposed a modified conjugate residual method to solve the generalized coupled variant of (2), and Dehdezi and Karimi [11] extended the conjugate gradient squared method and the conjugate residual squared method to obtain their iterative solutions. Zhang, Ding, and Li [45] mainly focused on proposing the tensor form of the generalized product-type biconjugate gradient method to solve the generalized Sylvester quaternion tensor equations (2). Heyouni, Movahed, and Tajaddini [20] used the Hessenberg process instead of the Arnoldi process to generate a basis of the Krylov subspace and then proposed an iterative method to solve the real tensor equation. In addition, Zhang and Wang [44] introduced the CGNR and CGNE methods for the third-order Sylvester tensor equation (2).

Let us contemplate the following partial differential equation (see, e.g., [3, 21]):

$$\begin{cases} -\Delta u + c^T \nabla u = f, & \text{in } \Omega = [0,1]^N, \\ u = 0, & \text{on } \partial\Omega. \end{cases}$$

The use of the finite-difference discretization together with a second-order convergent scheme for the convection term leads us to a linear system that is expressed in the form (2). Chen and Lu [9] established the projection method to solve the tensor equation (2). They also applied the Kronecker product preconditioner to accelerate the convergence of the iterative method. Later, Beik, Movahed, and Ahmadi-Asl [6] derived the Krylov subspace methods to solve the Sylvester tensor equation (2) in the case of 3-mode tensors. Shi, Wei, and Ling [37] investigated the backward error and perturbation bounds for the tensor equation (2) for the 3-mode tensors.

The high order Sylvester tensor equation, which uses the Einstein product, is defined in [38] and is given by

$$\mathcal{A} \star_N \mathcal{X} + \mathcal{X} \star_M \mathcal{B} = \mathcal{C}, \tag{3}$$

where $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}, \mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_M \times J_1 \times \cdots \times J_M}$, $\mathcal{C} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$ and $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$. It is noteworthy that the Sylvester tensor equation given in (3) comes from the discretization of the linear partial differential equation by the finite difference, finite element, and spectral methods in high dimension [19, 27, 28, 26].

Recently, Sun et al. [38] investigated the generalized inverses of tensors via the Einstein product. Using the generalized inverses of tensors, they also gave the general solutions of the tensor equation (3). Behera and Mishra [4] derived further results on generalized inverses of tensors via the Einstein product. Later, Wang and Xu [41] considered the iterative algorithms for solving the tensor equation (3). Moreover, Dehdezi and Karimi [12] presented an extended version of a gradient-based iterative method for solving large multilinear systems via the Einstein product. They introduced a new preconditioner to accelerate the convergence rate of the new iterative methods. As the gradient-based and the gradient-based least-squares algorithms, Dehdezi [10] derived iterative methods for the Sylvester-transpose tensor equation as (1). Erfanifar and Hajarian [16] also proposed a method for solving the nonlinear tensor equation

$$\mathcal{X} + \mathcal{A}^T \star_M \mathcal{X}^{-1} \star_N \mathcal{A} = \mathcal{I}$$

along with the Einstein product.

Brown and Hindmarsh [8] and then Jia [24] analyzed an incomplete generalized minimal residual method for solving large unsymmetric linear systems with low computational cost, which is a truncated version of the generalized minimal residual method (GMRES) [35]. Later, Saad and Wu [36] extracted a direct form of the incomplete generalized minimal residual method, abbreviated by DQGMRES, using QR decomposition of the Hessenberg matrix that appeared in the incomplete GMRES method. This motivates us to present an effective high order iterative algorithm as the DQGMRES method based on the tensor format to solve the generalized Sylvester tensor equation (1) via the Einstein product.

The outline of this paper is as follows. In Section 2, we concisely recall some definitions and properties of tensor operators that are useful in the rest of the paper. In Section 3, we derive the tensor form of the DQGMRES method for solving the generalized Sylvester tensor equation (1) via the Einstein product. In Section 4, we analyze the convergence properties of the proposed method and find an upper bound for the residual norm of the approximate solution. Moreover, in Section 5, we report some numerical experiments on solving (1) using the presented method to illustrate its effectiveness and accuracy. Finally, a conclusion is drawn in Section 6.

2 Preliminaries

In this section, some preliminary definitions, and a number of technical lemmas are given, which will be used in what follows.

Definition 1. [38] Let N, M, L be the positive integers, let $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$, and let $\mathcal{B} \in \mathbb{R}^{K_1\times\cdots\times K_M\times J_1\times\cdots\times J_L}$. The Einstein product of two tensors $\mathcal{A}$ and $\mathcal{B}$ is defined by the operation $\star_M$ via

$$(\mathcal{A} \star_M \mathcal{B})_{i_1\cdots i_N j_1\cdots j_L} = \sum_{k_M=1}^{K_M}\cdots\sum_{k_1=1}^{K_1} a_{i_1\cdots i_N k_1\cdots k_M} b_{k_1\cdots k_M j_1\cdots j_L}. \tag{4}$$

Thus $\mathcal{A} \star_M \mathcal{B} \in \mathbb{R}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_L}$ and the associative law of this tensor product holds.

For $\mathcal{A} = (a_{i_1\cdots i_N j_1\cdots j_M}) \in \mathbb{R}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_M}$, let $\mathcal{B} = (b_{i_1\cdots i_M j_1\cdots j_N}) \in \mathbb{R}^{J_1\times\cdots\times J_M\times I_1\times\cdots\times I_N}$ be a tensor with $b_{i_1\cdots i_M j_1\cdots j_N} = a_{j_1\cdots j_N i_1\cdots i_M}$. We call $\mathcal{B}$ the transpose of $\mathcal{A}$ and denote it by $\mathcal{A}^T$.

When $N = M = 1$, the tensor equation (1) reduces to

$$AXB + CXD = F, \tag{5}$$

which is the generalized Sylvester matrix equation and arises frequently from the areas of systems and control theory [13, 14]. According to the representation (5), the tensor equation (1) is called generalized Sylvester tensor equation.

Definition 2. [38] Let $\mathcal{A} = (a_{i_1\cdots i_N i_1 \cdots i_N}) \in \mathbb{R}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$. The trace of $\mathcal{A}$ is defined as

$$\mathrm{tr}(\mathcal{A}) = \sum_{i_N=1}^{I_N} \cdots \sum_{i_1=1}^{I_1} a_{i_1\cdots i_N i_1\cdots i_N}. \tag{6}$$

The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_M}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y}\rangle = \mathrm{tr}(\mathcal{Y}^T \star_N \mathcal{X}) = \sum_{j_M=1}^{J_M} \cdots \sum_{j_1=1}^{J_1} \sum_{i_N=1}^{I_N} \cdots \sum_{i_1=1}^{I_1} x_{i_1\cdots i_N j_1\cdots j_M} y_{i_1\cdots i_N j_1 \cdots j_M}. \tag{7}$$

Therefore, the tensor norm induced by the inner product (7) is acquired as

$$\|\mathcal{A}\| = \sqrt{\langle \mathcal{X}, \mathcal{X}\rangle} = \sqrt{\sum_{j_M=1}^{J_M} \cdots \sum_{j_1=1}^{J_1} \sum_{i_N=1}^{I_N} \cdots \sum_{i_1=1}^{I_1} |x_{i_1\cdots i_N j_1\cdots j_M}|^2}, \tag{8}$$

which is called the tensor Frobenius norm.

Let us set $I = I_1 I_2 \ldots I_N$ and, similarly, $J = J_1 J_2 \ldots J_N$, $K = K_1 K_2 \ldots K_M$, and $L = L_1 L_2 \ldots L_M$.

Definition 3. The transformation $\Phi_{IJ} : \mathbb{R}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_N} \to \mathbb{R}^{I\times J}$ with $\Phi_{IJ}(\mathcal{A}) = A$ is defined component-wisely as

$$(\mathcal{A})_{i_1\cdots i_N j_1\cdots j_N} \to (A)_{st},$$

where $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_N}$, $A \in \mathbb{R}^{I\times J}$, $s = i_N + \sum_{p=1}^{N-1}\Big((i_p - 1)\prod_{q=p+1}^{N} I_q\Big)$, and $t = j_N + \sum_{p=1}^{N-1}\Big((j_p - 1)\prod_{q=p+1}^{N} J_q\Big)$.

Routine computations verify that the tensor equation (1) is equivalent to the following large system of linear equations:

$$\mathcal{M}x = b, \tag{9}$$

with $x = \mathrm{vec}(\Phi_{IK}(\mathcal{X}))$, $b = \mathrm{vec}(\Phi_{IK}(\mathcal{F}))$, and

$$\mathcal{M} = B^T \otimes A + D^T \otimes C,$$

where $A = \Phi_{II}(\mathcal{A}))$, $B = \Phi_{KK}(\mathcal{B}))$, $C = \Phi_{II}(\mathcal{C}))$, and $D = \Phi_{KK}(\mathcal{D}))$.

The notation $\otimes$ represents the Kronecker product and the operator "vec" corresponds to a vector; see [17] for more details. The system of linear equations (9) is consistent if and only if (1) is consistent, which means that the coefficient matrix $\mathcal{M}$ needs to be nonsingular. In this study, it is assumed that the tensor equation (1) has a unique solution.

The jth *frontal slice* of an Nth order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ (also known as the column tensor of $\mathcal{X}$) is denoted by

$$\mathcal{X}_{\underbrace{:: \cdots :}_{(N-1)-\text{times}} j}, \quad \text{for } j = 1, 2, \ldots, I_N,$$

which is a tensor in $\mathbb{R}^{I_1 \times \cdots \times I_{N-1}}$ and is obtained by fixing the last index.

Definition 4. The operator $\times_n$ stands for the n-mode matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with a matrix $A \in \mathbb{R}^{J \times I_n}$ as $\mathcal{X} \times_n A_n$, which is an Nth order tensor of size $I_1 \times I_2 \times \cdots I_{k-1} \times J \times I_{k+1} \cdots \times I_N$. For each element, we have

$$(\mathcal{X} \times_n A)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \cdots i_N} a_{j i_n}. \tag{10}$$

Definition 5. The operator $\bar{\times}_n$ (for $n = 1, 2, \ldots, N$) represents the n-mode (vector) product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with a vector $v \in \mathbb{R}^{I_n}$ is indicated by $\mathcal{X} \bar{\times}_n v$, which is an $(N-1)$th order tensor of size $I_1 \times I_2 \times \cdots I_{n-1} \times I_{n+1} \cdots \times I_N$. The elements are defined as follows:

$$(\mathcal{X} \bar{\times}_n v)_{i_1 i_2 \cdots i_{n-1} i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} v_{i_n}.$$

Based on Definitions 4 and 5, one can establish some simple calculation rules for the matrix and the vector k-mode products representations; see [25] for further details.

Lemma 1. If $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, $A \in \mathbb{R}^{J_k \times I_k}$ and $v \in \mathbb{R}^{J_k}$, then

$$\mathcal{X} \times_k A \bar{\times}_k v = \mathcal{X} \bar{\times}_k (A^T v).$$

We can see the validity of the following proposition in [25], which is useful for our development.

Proposition 1. If $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, $A \in \mathbb{R}^{J_k \times I_k}$, and $B \in \mathbb{R}^{P_k \times J_k}$, then

$$(\mathcal{X} \times_k A) \times_k B = \mathcal{X} \times_k (BA).$$

Proposition 2. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ be an Nth order tensor and $v = e_j$ such that e_j is the jth column of the identity matrix $I^{(I_N)}$. Then

$$\mathcal{X} \bar{\times}_N v = \mathcal{X}_{::\cdots:j}, \quad j = 1, 2, \ldots, I_N.$$

Consider two N-mode tensors $\mathcal{X}$ and $\mathcal{Y}$. We define $\boxtimes^{(N)}$ product for $N = 1, 2, \ldots$, by beginning 1-mode tensor as a vector and developing the 2-mode tensor as a matrix. In point of fact, the $\boxtimes^{(1)}$ and $\boxtimes^{(2)}$ products are naturally written in the following forms:

$$\mathcal{X} \boxtimes^{(1)} \mathcal{Y} = \mathcal{X}^T \mathcal{Y}, \quad \mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1},$$

and

$$\mathcal{X} \boxtimes^{(2)} \mathcal{Y} = \mathcal{X}^T \mathcal{Y}, \quad \mathcal{X} \in \mathbb{R}^{I_1 \times I_2}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \tilde{I}_2}.$$

In general case, the $\boxtimes^{(N)}$ product between two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_{N-1} \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_{N-1} \times \tilde{I}_N}$ is defined as an $I_N \times \tilde{I}_N$ matrix whose (i, j)th element is

$$\left[\mathcal{X} \boxtimes^{(N)} \mathcal{Y}\right]_{ij} = \operatorname{tr}(\mathcal{X}_{::\cdots:i} \boxtimes^{(N-1)} \mathcal{Y}_{::\cdots:j}), \quad N = 2, 3, \ldots.$$

The following proposition from [6] presents some constructive relations for the $\boxtimes^{(N+1)}$ product and the $\bar{\times}_k$ vector product, which are useful for the convergence analysis of the proposed method.

Proposition 3. Suppose that $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N \times m}$ is an $(N+1)$-mode tensor with the N-mode column tensors $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_m \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}$ and $z = (z_1, z_2, \ldots, z_m)^T \in \mathbb{R}^m$. For an arbitrary $(N+1)$-mode tensor $\mathcal{A}$ with N-mode column tensors $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$, we have the following statements:

$$\mathcal{A} \boxtimes^{(N+1)} (\mathcal{B} \bar{\times}_{N+1} z) = \left(\mathcal{A} \boxtimes^{(N+1)} \mathcal{B}\right) z, \tag{11}$$

and

$$(\mathcal{B} \bar{\times}_{N+1} z) \boxtimes^{(N+1)} \mathcal{A} = z^T \left(\mathcal{B} \boxtimes^{(N+1)} \mathcal{A}\right). \tag{12}$$

In the spirit of the fact that $\|\mathcal{X}\|^2 = \mathrm{tr}(\mathcal{X} \boxtimes^{(N)} \mathcal{X}) = \mathcal{X} \boxtimes^{(N+1)} \mathcal{X}$, and also using Proposition 3, the next proposition is acquired.

Definition 6. The set of N-mode tensors $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}$ is called orthonormal if

$$\langle \mathcal{V}_i, \mathcal{V}_j \rangle = 0, \quad i, j = 1, 2, \ldots, m(i \neq j),$$

and $\langle \mathcal{V}_i, \mathcal{V}_i \rangle = 1$ for $i = 1, 2, \ldots, m$.

Remark 1. Suppose that $\mathcal{A}$ is a given $(N+1)$-mode tensor with the column tensors $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}$. If the set of N-mode tensors $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$ is orthonormal, then

$$\mathcal{A} \boxtimes^{(N+1)} \mathcal{A} = I^{(m)}.$$

Proposition 4. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an N-mode tensor, and let $v \in \mathbb{R}^{I_N}$. Then,

$$\|\mathcal{X} \bar{\times}_N v\| \leq \|\mathcal{X}\| \|v\|_2.$$

Remark 2. In the case that the frontal slices of a tensor $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is orthonormal, then for $v \in \mathbb{R}^{I_N}$, Remark 1 concludes

$$\|\mathcal{F} \bar{\times}_N v\| = \|v\|_2.$$

3 Tensor form of the quasi-GMRES method

By using given tensors $\mathcal{A}, \mathcal{C} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$, $\mathcal{B}, \mathcal{D} \in \mathbb{R}^{K_1 \times \cdots \times K_M \times K_1 \times \cdots \times K_M}$, we define the following linear operator:

$$\mathcal{L} : \mathbb{R}^{I_1 \times \cdots \times I_N \times K_1 \times \cdots \times K_M} \to \mathbb{R}^{I_1 \times \cdots \times I_N \times K_1 \times \cdots \times K_M},$$

as

$$\mathcal{X} \mapsto \mathcal{L}(\mathcal{X}) := \mathcal{A} \star_N \mathcal{X} \star_M \mathcal{B} + \mathcal{C} \star_N \mathcal{X} \star_M \mathcal{D}.$$

Based on the above definition, the generalized Sylvester tensor equation (1) is stated as $\mathcal{L}(\mathcal{X}) = \mathcal{F}$.

Thanks to the above discussion, the kth tensor Krylov subspace associated with the linear operator $\mathcal{L}$ and a tensor $\mathcal{V} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times K_1 \times \cdots \times K_M}$ is defined as

$$\mathcal{K}_k(\mathcal{L},\mathcal{V}) = \text{span}\{\mathcal{V}, \mathcal{L}(\mathcal{V}), \ldots \mathcal{L}^{k-1}(\mathcal{V})\},$$

where $\mathcal{L}^i(\mathcal{V}) = \mathcal{L}(\mathcal{L}^{i-1}(\mathcal{V}))$ and $\mathcal{L}^0(\mathcal{V}) = \mathcal{V}$.

First, we introduce a useful alternative to the well-known Arnoldi process by truncating the orthogonalization process [24]. In this way, we achieve a strategy with low computational cost and a small truncation parameter m. It is emphasized that the truncation parameter m for the kth tensor Krylov subspace must be satisfied $2 \leq m \leq k$. Here, we start with the tensor form of the incomplete orthogonalization process (IOP_BTF), described by Algorithm 2.

Algorithm 2: IOP_BTF

1. **Input:** Given tensors $\mathcal{A}, \mathcal{C} \in \mathbb{R}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$,
   $\mathcal{B}, \mathcal{D} \in \mathbb{R}^{K_1\times\cdots\times K_M\times K_1\times\cdots\times K_M}$ and $\mathcal{V} \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$.
2. Set $\beta = \|\mathcal{V}\|$ and $\mathcal{V}_1 = \mathcal{V}/\beta$
3. For given k, define $(k+1)\times k$ matrix $\bar{H}_k$, and set $\bar{H}_k = 0$;
4. **for** $j = 1, 2, \ldots, k$ **do**
5. Compute $\mathcal{W}_j = \mathcal{L}(\mathcal{V}_j)$
6. **for** $i = \max\{1, j-m+1\}, \ldots, j$ **do**
7. $h_{ij} = \langle \mathcal{W}_j, \mathcal{V}_i \rangle$
8. $\mathcal{W}_j = \mathcal{W}_j - h_{ij}\mathcal{V}_i$
9. **end**
10. Compute $h_{j+1,j} = \|\mathcal{W}_j\|$ and $\mathcal{V}_{j+1} = \frac{\mathcal{W}_j}{h_{j+1,j}}$
11. **end**
12. **Output:** Tensors $\mathcal{V}_j$, for $j = 1, 2, \ldots, k+1$ and matrix $\bar{H}_k$.

It is plain to verify that the IOP_BTF strategy produces the locally orthonormal basis $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k$ (only the last m tensors $\mathcal{V}_i$'s are orthonormal) for the tensor Krylov subspace $\mathcal{K}_k(\mathcal{L},\mathcal{V})$ [35].

Let $\bar{H}_k = [h_{ij}]_{(k+1)\times k}$ be the matrix whose nonzero entries are those computed in lines 7 and 10 of Algorithm 2. We denote H_k as the matrix obtained from $\bar{H}_k$ by deleting its last row. Note that the Hessenberg matrix H_k has a band structure with a bandwidth $m+1$. Assume that $\tilde{\mathcal{V}}_k$ is the $(M+N+1)$-mode tensor with the frontal slices $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k$ obtained by

Algorithm 2 with the truncation parameter m. Beik, Movahed, and Ahmadi-Asl [6] have proven the following statement for the Arnoldi_BTF process

$$[\mathcal{L}(\mathcal{V}_1), \cdots, \mathcal{L}(\mathcal{V}_k)] = \tilde{\mathcal{V}}_{k+1} \times_{(N+M+1)} \bar{H}_k^T, \tag{13}$$

which is also satisfied for the IOP_BTF strategy in Algorithm 2.

Here, we briefly recall how the well-known GMRES method can be extended based on the basis of the tensor form. Let $\mathcal{X}_0 \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$ be a given initial tensor guess for the exact solution of (1) with the corresponding residual tensor $\mathcal{R}_0 = \mathcal{F} - \mathcal{L}(\mathcal{X}_0) \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$. For the approximate solution $\mathcal{X}_k$ computed at the kth iterative step of the GMRES_BTF method [23], we consider

$$\mathcal{X}_k \in \mathcal{X}_0 + \mathcal{K}_k(\mathcal{L}, \mathcal{R}_0),$$

and

$$\|\mathcal{F} - \mathcal{L}(\mathcal{X}_k)\| = \min_{\mathcal{X}\in\mathcal{X}_0+\mathcal{K}_k(\mathcal{L},\mathcal{R}_0)} \|\mathcal{F} - \mathcal{L}(\mathcal{X})\|. \tag{14}$$

So, the quasi-GMRES method (QGMRES) consists of performing the IOP_BTF and constructing $\mathcal{X}_k = \mathcal{X}_0 + \tilde{\mathcal{V}}_k \bar{\times}_{(M+N+1)} y_k$, where y_k is obtained as the condition (14) holds true; see [23] for more details.

As Saad and Wu mentioned in [36], the dimension of the Krylov subspace in the GMRES method increases by one at each step, which makes the procedure impractical for large dimensions. There are two standard remedies to this problem. The first is to restart the algorithm. In a simple way, the dimension is fixed, and the algorithm is restarted as many times as necessary, defining the initial vector defined as the latest approximation from the previous outer iteration. An alternative is to truncate the long-recurrence of the Arnoldi process as described in the IOP_BTF strategy in Algorithm 2.

Following the incomplete GMRES method presented by Brown and Hindmarsh [8], we now describe the QGMRES method based on the tensor format (QGMRES_BTF) in Algorithm 3.

Algorithm 3: QGMRES_BTF

1. **Input:** Given tensors $\mathcal{A}, \mathcal{C} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times I_1 \times \cdots \times I_N}$, $\mathcal{B}, \mathcal{D} \in \mathbb{R}^{K_1 \times \cdots \times K_M \times K_1 \times \cdots \times K_M}$ and $\mathcal{F} \in \mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$, truncation parameter m, and initial guess $\mathcal{X}_0 \in \mathbb{R}^{I_1 \times \cdots \times I_N \times K_1 \times \cdots \times K_M}$.
2. Compute $\mathcal{R}_0 = \mathcal{F} - \mathcal{L}(\mathcal{X}_0)$, set $\beta = \|\mathcal{R}_0\|$ and $\mathcal{V}_1 = \mathcal{R}_0/\beta$
3. For given k, define $(k+1) \times k$ matrix $\bar{H}_k$, and set $\bar{H}_k = 0$
4. **for** $j = 1, 2, \ldots, k$ **do**
5. Compute $\mathcal{W}_j = \mathcal{L}(\mathcal{V}_j)$
6. **for** $i = \max\{1, j-m+1\}, \ldots, j$ **do**
7. $h_{ij} = \langle \mathcal{W}_j, \mathcal{V}_i \rangle$
8. $\mathcal{W}_j = \mathcal{W}_j - h_{ij}\mathcal{V}_i$
9. **end**
10. Compute $h_{j+1,j} = \|\mathcal{W}_j\|$ and $\mathcal{V}_{j+1} = \frac{\mathcal{W}_j}{h_{j+1,j}}$
11. **end**
12. Solve the problem $y_k = \text{argmin}_{y \in \mathbb{R}^k} \|\bar{H}_k y - \beta e_1\|$
13. Compute $\mathcal{X}_k = \mathcal{X}_0 + \tilde{\mathcal{V}}_k \bar{\times}_{(M+N+1)} y_k$
14. **Output:** Approximate solution $\mathcal{X}_k$.

Constructing of $\hat{\mathcal{V}}_{k+1}$ and its first frontal slice as $\mathcal{V}_1 = \mathcal{R}_0/\beta$, yields $\mathcal{R}_0 = \hat{\mathcal{V}}_{k+1} \bar{\times}_{(M+N+1)} (\beta e_1)$, where e_1 is the first column of the identity matrix $I^{(k+1)}$. By using Lemma 1 and also making use of (13), the residual tensor $\mathcal{R}_k$ for the QGMRES_BTF approximate solution $\mathcal{X}_k$ generated by Algorithm 3 is given by

$$\begin{aligned} \mathcal{R}_k &= \mathcal{R}_0 - (\tilde{\mathcal{V}}_{k+1} \times_{(M+N+1)} \bar{H}_k^T) \bar{\times}_{(M+N+1)} y_k, \\ &= \tilde{\mathcal{V}}_{k+1} \bar{\times}_{(M+N+1)} [\beta e_1 - \bar{H}_k y_k]. \end{aligned}$$

The norm of the residual tensor $\mathcal{R}_k$ is then formulated as

$$\|\mathcal{R}_k\| = \|\tilde{\mathcal{V}}_{k+1} \bar{\times}_{(M+N+1)} [\beta e_1 - \bar{H}_k y_k]\|, \tag{15}$$

where, as before in Algorithm 3, y_k minimizes the norm $\|\beta e_1 - \bar{H}_k y\|_2$ over all vectors y in $\mathbb{R}^k$. This approach does not minimize the actual norm of the residual tensor over $\mathcal{X}_0 + \mathcal{K}_k(\mathcal{L}, \mathcal{R}_0)$. This idea leads us to minimize the norm $\|\beta e_1 - \bar{H}_k y\|_2$ by the QR factorization method. We implement the direct variant of the QGMRES idea motivated from [36] by using the Givens rotation matrices to transform $\bar{H}_k$ and βe_1, to get

$$\bar{R}_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \text{ and } \bar{g}_k = (\gamma_1, \gamma_2, \ldots, \gamma_{k+1})^T, \tag{16}$$

respectively, in which R_k is an upper triangular matrix. Actually, we construct the following unitary matrix of order $k+1$

$$\mathbf{Q}_k = \Omega_k \cdots \Omega_2 \Omega_1, \tag{17}$$

where the $(k+1) \times (k+1)$ Givens rotation matrices

$$\Omega_i = \begin{pmatrix} I^{(i-1)} & & & \\ & c_i & s_i & \\ & -s_i & c_i & \\ & & & I^{(k-i)} \end{pmatrix} \equiv \begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix}, \quad i = 1, 2, \ldots, k, \tag{18}$$

are used with $c_i^2 + s_i^2 = 1$ in which $I^{(n)}$ indicates the identity matrix of order n. We now construct the following pre-multiplication operations on the kth column of $\bar{H}_k$:

$$\Omega_{k-1}\Omega_{k-2}\cdots\Omega_{k-m} \begin{pmatrix} \vdots \\ 0 \\ 0 \\ h_{k-m+1,k} \\ \vdots \\ h_{kk} \\ h_{k+1,k} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ 0 \\ t_{k-m,k} \\ t_{k-m+1,k} \\ \vdots \\ t_{kk} \\ h_{k+1,k} \\ \vdots \end{pmatrix}. \tag{19}$$

By adopting Ω_k in the kth column of the result vector in (19), we get

$$\begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} \vdots \\ 0 \\ t_{k-m,k} \\ \vdots \\ t_{kk} \\ h_{k+1,k} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ 0 \\ t_{k-m,k} \\ \vdots \\ t_{kk} \\ 0 \\ \vdots \end{pmatrix}, \tag{20}$$

with $c_k = \frac{t_{kk}}{\sqrt{t_{kk}^2 + h_{k+1,k}^2}}$ and $s_k = \frac{h_{k+1,k}}{\sqrt{t_{kk}^2 + h_{k+1,k}^2}}$. For elements of $\bar{g}_k$, we have the recurrence relations $\gamma_{k+1} = -s_k \gamma_k$ and $\gamma_k = c_k \gamma_k$, with the initial term $\gamma_1 = \beta$.

Then, for any vector $y \in \mathbb{R}^k$, one has

$$\begin{aligned} \|\beta e_1 - \bar{H}_k y\|_2^2 &= \|\mathbf{Q}_k(\beta e_1 - \bar{H}_k y)\|_2^2 \\ &= \|\bar{g}_k - \bar{R}_k y\|_2^2 \\ &= |\gamma_{k+1}|^2 + \|g_k - R_k y\|_2^2. \end{aligned} \tag{21}$$

The minimum of the left-hand side is reached when the second term on the right-hand side of (21) has disappeared. Since R_k is nonsingular, the minimum of (21) is obtained by $y_k = R_k^{-1} g_k$, in which g_k is the vector obtained by removing the last element γ_{k+1} from $\bar{g}_k$. We therefore have $\|\beta e_1 - \bar{H}_k y_k\|_2 = |\gamma_{k+1}|$.

Following the above discussion and making use of Lemma 1, we obtain

$$\begin{aligned} \mathcal{X}_k &= \mathcal{X}_0 + \tilde{\mathcal{V}}_k \bar{\times}_{N+1} y_k \\ &= \mathcal{X}_0 + \tilde{\mathcal{V}}_k \bar{\times}_{N+1} (R_k^{-1} g_k) \\ &= \mathcal{X}_0 + (\tilde{\mathcal{V}}_k \times_{N+1} R_k^{-T}) \bar{\times}_{N+1} g_k \\ &= \mathcal{X}_0 + \tilde{\mathcal{P}}_k \bar{\times}_{N+1} g_k \\ &= \mathcal{X}_0 + \tilde{\mathcal{P}}_{k-1} \bar{\times}_{N+1} g_{k-1} + \gamma_k \mathcal{P}_k \\ &= \mathcal{X}_{k-1} + \gamma_k \mathcal{P}_k, \end{aligned}$$

where $\tilde{\mathcal{P}}_k = \tilde{\mathcal{V}}_k \times_{N+1} R_k^{-T}$ with the frontal slices $\mathcal{P}_i$'s. By using Proposition 1, we conclude that $\tilde{\mathcal{V}}_k = \tilde{\mathcal{P}}_k \times_{N+1} R_k^T$, and straightforward computations yield

$$\mathcal{P}_1 = \mathcal{V}_1/t_{11},$$
$$\mathcal{P}_2 = (\mathcal{V}_2 - t_{12}\mathcal{P}_1)/t_{22},$$
$$\vdots$$
$$\mathcal{P}_k = t_{kk}^{-1}\left(\mathcal{V}_k - \sum_{i=k-m}^{k-1} t_{ik}\mathcal{P}_i\right),$$

where t_{ik} for $i = k-m, \dots, k-1, k$ are the elements of the kth column of the upper triangular matrix R_k in (20). We can describe the DQGMRES algorithm based on the tensor format (DQGMRES_BTF) for solving the generalized Sylvester tensor equation (1) via the Einstein product as done in Algorithm 4.

Algorithm 4: DQGMRES_BTF

1. **Input:** Given tensors $\mathcal{A}, \mathcal{C} \in \mathbb{R}^{I_1\times\cdots\times I_N\times I_1\times\cdots\times I_N}$, $\mathcal{B}, \mathcal{D} \in \mathbb{R}^{K_1\times\cdots\times K_M\times K_1\times\cdots\times K_M}$ and $\mathcal{F} \in \mathbb{R}^{I_1\times\cdots\times I_N\times J_1\times\cdots\times J_M}$, truncation parameter m, and initial guess $\mathcal{X}_0 \in \mathbb{R}^{I_1\times\cdots\times I_N\times K_1\times\cdots\times K_M}$.
2. Compute $\mathcal{R}_0 = \mathcal{F} - (\mathcal{A} \star_N \mathcal{X}_0 \star_M \mathcal{B} + \mathcal{C} \star_N \mathcal{X}_0 \star_M \mathcal{D})$, $\gamma_1 = \|\mathcal{R}_0\|$, $\mathcal{V}_1 = \mathcal{R}_0/\gamma_1$
3. **for** $k = 0, 1, \dots$ until convergence **do**
4. Compute $h_{ik}, i = \max\{1, k-m+1\}, \dots, k$, and $\mathcal{V}_{k+1}$ as in lines 2 to 10 of Algorithm 2
5. Update the QR factorization of $\bar{H}_k$ according (19) and (20): i.e.
6. Apply $\Omega_i, i = k-m, \dots, k-1$, to the kth column of $\bar{H}_k$
7. Compute the rotation coefficients c_k and s_k
8. Apply rotation Ω_k to the last column of $\bar{H}_k$ and to $\bar{g}_k$; i.e. compute
9. $\gamma_{k+1} = -s_k\gamma_k, \quad \gamma_k = c_k\gamma_k$
10. $t_{kk} = \sqrt{h_{k+1,k}^2 + t_{kk}^2}$
11. $\mathcal{P}_k = \left(\mathcal{V}_k - \sum_{i=k-m}^{k-1} t_{ik}\mathcal{P}_i\right)/t_{kk}$
12. $\mathcal{X}_k = \mathcal{X}_{k-1} + \gamma_k\mathcal{P}_k$
13. If $|\gamma_{k+1}|$ is small enough the Stop
14. **End**
15. **Output:** Approximate solution $\mathcal{X}_k$ for (1)

4 Convergence analysis of the QGMRES_BTF method

We prove here some convergence results for the DQGMRES_BTF method. The next theorem provides a representation of the residual tensor $\mathcal{R}_k$ of the DQGMRES_BTF method.

Lemma 2. Let $\tilde{\mathcal{V}}_k$ be an $(M+N+1)$-mode tensor with column tensors $\mathcal{V}_i$ for $i = 1, 2, \ldots, k$ which is generated by Algorithm 2 and $\mathbf{Q}_k$ the unitary matrix specified in (17). The residual tensor $\mathcal{R}_k$ of the DQGMRES_BTF method is then given by

$$\mathcal{R}_k = \gamma_{k+1}\tilde{\mathcal{V}}_{k+1} \times_{(M+N+1)} \mathbf{Q}_k \bar{\times}_{(M+N+1)} e_{k+1}, \tag{22}$$

where γ_{k+1} is the last element of $\bar{g}_k$ in (16).

Proof. As discussed earlier in (16), the kth residual iterate of the DQGMRES_BTF method has the following form:

$$\begin{aligned}\mathcal{R}_k &= \mathcal{R}_0 - \mathcal{L}(\tilde{\mathcal{V}}_k)\bar{\times}_{(M+N+1)} y_k \\ &= \tilde{\mathcal{V}}_{k+1}\bar{\times}_{(M+N+1)}(\beta e_1 - \bar{H}_k y_k) \\ &= \tilde{\mathcal{V}}_{k+1}\bar{\times}_{(M+N+1)}(\mathbf{Q}_k^T(\bar{g}_k - \bar{R}_k y_k)).\end{aligned}$$

In view of (21), one can see that y_k minimizes the 2-norm of $\bar{g}_k - \bar{R}_k y$ over y and thus annihilates all components of the right-hand side $\bar{g}_k$ except the last one, which is equal to $\gamma_{k+1}e_{k+1}$. Now, it follows that

$$\mathcal{R}_k = \tilde{\mathcal{V}}_{k+1}\bar{\times}_{(M+N+1)}(\mathbf{Q}_k^T(\gamma_{k+1}e_{k+1})).$$

Finally, making use of Lemma 1 completes the proof of the lemma. □

Next, we present a suitable upper bound for the residual norm of the DQGMRES_BTF method, which depends on the specific parameter computed in the proposed Algorithm 4 in a cost-effective way. For this purpose, we prove the following Lemma.

Lemma 3. The residual $\mathcal{R}_k$ obtained by the DQGMRES_BTF algorithm with the truncation parameter m for the generalized Sylvester tensor equations of the form (1) satisfies the following inequality:

$$\|\mathcal{R}_k\| \leq |\gamma_{k+1}|\sqrt{k-m+1}.$$

Proof. From Lemmas 1 and 2, we have

$$\mathcal{R}_k = \gamma_{k+1}\tilde{\mathcal{V}}_{k+1}\bar{\times}_{(M+N+1)}(\mathbf{Q}_k^T e_{k+1}).$$

Let $q = \mathbf{Q}_k^T e_{k+1}$ be the unit vector with components $\eta_1, \eta_2, \ldots, \eta_{k+1}$. Then by using Proposition 4, we get

$$\begin{aligned}
\|\mathcal{R}_k\| &= |\gamma_{k+1}|\|\tilde{\mathcal{V}}_{k+1}\bar{\times}_{(M+N+1)} q\| \\
&\leq |\gamma_{k+1}|\left(\left\|\sum_{i=1}^{m+1}\eta_i\mathcal{V}_i\right\| + \left\|\sum_{i=m+2}^{k+1}\eta_i\mathcal{V}_i\right\|\right) \\
&\leq |\gamma_{k+1}|\left(\left[\sum_{i=1}^{m+1}\eta_i^2\right]^{1/2} + \sum_{i=m+2}^{k+1}|\eta_i|\|\mathcal{V}_i\|\right) \\
&\leq |\gamma_{k+1}|\left(\left[\sum_{i=1}^{m+1}\eta_i^2\right]^{1/2} + \sqrt{k-m}\left[\sum_{i=m+2}^{k+1}\eta_i^2\right]^{1/2}\right) \\
&\leq |\gamma_{k+1}|\sqrt{k-m+1},
\end{aligned}$$

where the last inequality follows by the Cauchy–Schwarz inequality and $\|q\|_2 = 1$. □

The next corollary is come to the conclusion by Lemma 3 together with the useful relation between the last elements γ_{k+1} and γ_k of $\bar{g}_k$ and $\bar{g}_{k-1}$, respectively; that is, $\gamma_{k+1} = -s_k\gamma_k$.

Corollary 1. Let $\mathcal{R}_k$ be the residual tensor of the DQGMRES_BTF kth iterate. Then

$$\|\mathcal{R}_k\| \leq |s_1 s_2 \cdots s_k|\|\mathcal{R}_0\|\sqrt{k-m+1},$$

where s_i's are defined as (18).

An extension of the Gram–Schmidt orthogonalization process based on the tensor format concludes the following lemma for the linear independent tensors $\mathcal{V}_i$ for $i = 1, 2, \ldots, k$, which are generated by Algorithm 2.

Lemma 4. Suppose that $\tilde{\mathcal{V}}_{k+1}$ is an $(M+N+1)$-order tensor with the $k+1$ frontal slices $\mathcal{V}_i$ for $i = 1, 2, \ldots, k+1$ obtained by using the IOP_BTF

Algorithm 2. Then, there is an $(k+1)\times(k+1)$ nonsingular matrix $\mathbf{U}$ such that $\tilde{\mathcal{V}}_{k+1} = \tilde{\mathcal{F}}_{k+1} \times_{(M+N+1)} \mathbf{U}$, where $\tilde{\mathcal{F}}_{k+1}$ is an $(M+N+1)$-order tensor with the $k+1$ orthonormal frontal slices $\mathcal{F}_i$ for $i = 1, 2, \ldots, k+1$; that is,

$$\tilde{\mathcal{F}}_{k+1} \boxtimes^{(M+N+1)} \tilde{\mathcal{F}}_{k+1} = I^{(k+1)}. \tag{23}$$

Proof. The proof is a direct result of the tensor form of the Gram–Schmidt orthogonalization process described in [35]. □

In the last theorem of this section, an inequality is found that can be usefully applied in the convergence analysis of the DQGMRES_BTF method. This is a comparison of the residual tensor obtained after k steps of using the DQGMRES_BTF method with that of the GMRES_BTF method [9].

Theorem 1. Assume that $\tilde{\mathcal{V}}_{k+1}$ is an $(M+N+1)$-order tensor with $k+1$ frontal slices $\mathcal{V}_i$ for $i = 1, 2, \ldots, k+1$ obtained by using the IOP_BTF Algorithm 2, $\tilde{\mathcal{V}}_{k+1} = \tilde{\mathcal{F}}_{k+1} \times_{(M+N+1)} \mathbf{U}^T$, where $\tilde{\mathcal{F}}_{k+1}$ is satisfied (23) and $\mathbf{U}^T$ is nonsingular. Let $\mathcal{R}_k^Q$ and $\mathcal{R}_k^G$ be the residual obtained after k steps of using DQGMRES_BTF and GMRES_BTF methods, respectively. Then

$$\|\mathcal{R}_k^Q\| \leq \kappa_2(\mathbf{U})\|\mathcal{R}_k^G\|, \tag{24}$$

where $\kappa_2(\mathbf{U})$ is the condition number of the matrix $\mathbf{U}$.

Proof. Consider the subset of $\mathcal{K}_{k+1}(\mathcal{L}, \mathcal{V}_1)$ given by

$$\mathcal{N} = \{\mathcal{R} : \mathcal{R} = \tilde{\mathcal{V}}_{k+1} \bar{\times}_{(M+N+1)} t;\ t = \beta e_1 - \bar{H}_k y;\ y \in \mathbb{R}^k\}.$$

Denote by $\mathbf{y}_k$ the minimizer of $\|\beta e_1 - \bar{H}_k y\|_2$ over y and $\mathbf{t}_k = \beta e_1 - \bar{H}_k \mathbf{y}_k$. Thus, Lemma 2 concludes that $\mathcal{R}_k^Q = \tilde{\mathcal{V}}_{k+1} \bar{\times}_{(M+N+1)} \mathbf{t}_k$. For any member $\mathcal{R} \in \mathcal{N}$, there exists $\mathbf{t}$ such that $\mathcal{R} = \tilde{\mathcal{F}}_{k+1} \times_{(M+N+1)} \mathbf{U}^T \bar{\times}_{N+1} \mathbf{t}$, which is defined by Lemma 1, it is equivalent to $\mathcal{R} = \tilde{\mathcal{F}}_{k+1} \bar{\times}_{(M+N+1)} (\mathbf{U}\mathbf{t})$. Hence, Proposition 3 yields $\mathbf{U}\mathbf{t} = \tilde{\mathcal{F}}_{k+1} \boxtimes^{(M+N+1)} \mathcal{R}$. Since $\mathbf{U}$ is nonsingular, thus

$$\mathbf{t} = \mathbf{U}^{-1}(\tilde{\mathcal{F}}_{k+1} \boxtimes^{(M+N+1)} \mathcal{R}).$$

From the unitary property of $\tilde{\mathcal{F}}_{k+1}$, we deduce that

$$\|\mathcal{R}_k^Q\| = \|\mathbf{U}\mathbf{t}_k\|_2 \leq \|\mathbf{U}\|_2 \|\mathbf{t}_k\|_2. \tag{25}$$

Note that $\|\mathbf{t}_k\|_2$ is the minimum of the 2-norm of $\beta e_1 - \bar{H}_k y$ over y. Therefore,

$$\begin{aligned}\|\mathbf{t}_k\|_2 &= \|\mathbf{U}^{-1}(\tilde{\mathcal{F}}_{k+1} \boxtimes^{(M+N+1)} \mathcal{R}_k^Q)\| \\ &\leq \|\mathbf{U}^{-1}(\tilde{\mathcal{F}}_{k+1} \boxtimes^{(M+N+1)} \mathcal{R})\| \\ &\leq \|\mathbf{U}^{-1}\|_2 \|\tilde{\mathcal{F}}_{k+1} \boxtimes^{(M+N+1)} \mathcal{R}\|.\end{aligned}$$

It is convenient to obtain $\|\tilde{\mathcal{F}}_{k+1} \boxtimes^{(M+N+1)} \mathcal{R}\| = \|\mathcal{R}\|$, and then

$$\begin{aligned}\|\mathbf{t}_k\|_2 &\leq \|\mathbf{U}^{-1}\|_2 \|\mathcal{R}\|, \quad \text{for all } \mathcal{R} \in \mathcal{N} \\ &\leq \|\mathbf{U}^{-1}\|_2 \|\mathcal{R}_k^G\|.\end{aligned}$$

Consequently, equation (25) is revealed as

$$\begin{aligned}\|\mathcal{R}_k^Q\| &\leq \|\mathbf{U}\|_2 \|\mathbf{U}^{-1}\|_2 \|\mathcal{R}_k^G\| \\ &= \kappa_2(\mathbf{U}) \|\mathcal{R}_k^G\|.\end{aligned}$$

The result is now concluded. □

5 Numerical results

In this section, we present some numerical results to illustrate the effectiveness and accuracy of the proposed DQGMRES_BTF method for solving several types of the generalized Sylvester tensor equation (1) via the Einstein product. To this end, we compare the DQGMRES_BTF method with the CGNR_BTF method given in [10], the CGNE_BTF method proposed in [12] as the tensor format of the CGNR and CGNE algorithms in [35], respectively. We compare also our results with those of the RNSD_BTF method proposed in [5]. All computations were performed using double-precision floating-point arithmetic in MATLAB codes. The computer we used is a system with the specification Intel(R) Core(TM) i3 CPU 2.13GHz, 4G RAM, and 64-bit operating system. In all examples, we choose zero tensor $\mathcal{X}_0 = \mathcal{O}$ as the initial guess. It must be emphasized that no preconditioning was used for any of the test problems. We consider the stopping criterion

$$ERR \equiv \frac{\|\mathcal{R}_k\|}{\|\mathcal{R}_0\|} \leq 10^{-6},$$

where $\mathcal{R}_k$ is the residual tensor corresponding to the approximate solution $\mathcal{X}_k$; that is,

$$\mathcal{R}_k = \mathcal{F} - \mathcal{A} \star_N \mathcal{X}_k \star_M \mathcal{B} - \mathcal{C} \star_N \mathcal{X}_k \star_M \mathcal{D}.$$

If the stopping criterion mentioned above does not apply, then we consider the maximum number of iterations Max-Iter = 1000 in each example. In all tensor computations, we get help from the MATLAB Tensor Toolbox, developed by Bader and Kolda [1, 2] to implement MATLAB.

Example 1. [7, 23] Consider the 3D Poisson problem

$$\begin{cases} -\nabla^2 v = f, & \text{in } \Omega = \{(x,y,z), 0 < x,y,z < 1\}, \\ v = 0, & \text{on } \partial\Omega \end{cases} \tag{26}$$

where f is a given function and

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}.$$

Several problems in physics and mechanics are modeled by (26), where the solution v means, for example, temperature, electromagnetic potential, or displacement of an elastic membrane fixed at the boundary. Now, we consider an approximation of the unknown function $v(x,y,z)$ in (26) corresponding to the uniform mesh step sizes, namely, Δx in the x-direction, Δy in the y-direction, and Δz in the z-direction, satisfy $\Delta x = \Delta y = \Delta z = h = \frac{1}{N+1}$. By the standard central finite difference formulas for the three dimensions, we obtain the following difference relationship:

$$6v_{ijk} - v_{i-1,j,k} - v_{i+1,j,k} - v_{i,j-1,k} - v_{i,j+1,k} - v_{i,j,k-1} - v_{i,j,k+1} = h^3 f_{ijk}. \tag{27}$$

Hence, the higher order tensor representation of the 3D discretized Poisson problem (26) as described in (27) is given by

$$\bar{\mathcal{A}}_N \star_3 \mathcal{V} = \mathcal{F}, \tag{28}$$

where the Laplacian tensor $\bar{\mathcal{A}}_N \in \mathbb{R}^{N\times N\times N\times N\times N\times N}$ and $\mathcal{V}, \mathcal{F} \in \mathbb{R}^{N\times N\times N}$. Both $\mathcal{V}$ and $\mathcal{F}$ are discretized on the unit cube. The entries on the tensor block $(\bar{\mathcal{A}}_N)^{(2,4,6)}_{l,m,n}$ of $\bar{\mathcal{A}}_N$ in (28) follow a seven-point stencil as

$$
\begin{aligned}
((\bar{\mathcal{A}}_N)^{(2,4,6)}_{\alpha,\beta,\gamma})_{\alpha,\beta,\gamma} &= \frac{6}{h^3},\\
((\bar{\mathcal{A}}_N)^{(2,4,6)}_{\alpha,\beta,\gamma})_{\alpha-1,\beta,\gamma} = ((\bar{\mathcal{A}}_N)^{(2,4,6)}_{\alpha,\beta,\gamma})_{\alpha+1,\beta,\gamma} &= -\frac{1}{h^3},\\
((\bar{\mathcal{A}}_N)^{(2,4,6)}_{\alpha,\beta,\gamma})_{\alpha,\beta-1,\gamma} = ((\bar{\mathcal{A}}_N)^{(2,4,6)}_{\alpha,\beta,\gamma})_{\alpha,\beta+1,\gamma} &= -\frac{1}{h^3},\\
((\bar{\mathcal{A}}_N)^{(2,4,6)}_{\alpha,\beta,\gamma})_{\alpha,\beta,\gamma-1} = ((\bar{\mathcal{A}}_N)^{(2,4,6)}_{\alpha,\beta,\gamma})_{\alpha,\beta,\gamma+1} &= -\frac{1}{h^3},
\end{aligned}
$$

for $\alpha, \beta, \gamma = 2, \ldots, N-1$. We use the notation $(\bar{\mathcal{A}}_N)^{(2,4,6)}_{l,m,n} = \bar{\mathcal{A}}_N(:, l, :, m, :, n)$ for the block tensors of $\bar{\mathcal{A}}_N$. For different grids $N = 4, 6, 8$, the iteration number and the CPU time of the CGNR_BTF and the CGNE_BTF methods are reported in Table 1, compared with the proposed DQGMRES_BTF method with the truncation parameter $m = 5$. The corresponding convergence histories of the numerical results are depicted in Figures 1 and 2 with the truncation parameter $m = 10$ of the DQGMRES_BTF method for $N = 8$ and $N = 10$, respectively. These results show that the DQGMRES_BTF algorithm is more effective and less expensive than the other solvers.

Table 1: Results of the iteration number (Iter) and CPU time (Time) for Example 1 with different Grids and the truncation parameter $m = 10$.

Methods	CGNE_BTF		CGNR_BTF		DQGMRES_BTF	
Grid	Time	Iter	Time	Iter	Time	Iter
$4 \times 4 \times 4$	0.1635	6	0.1626	6	0.0126	6
$6 \times 6 \times 6$	5.9706	19	5.9789	19	0.1311	19
$8 \times 8 \times 8$	67.7331	38	67.5701	38	0.5693	26

Example 2. Let us consider Sylvester tensor equation,

$$
\mathcal{A} \star_N \mathcal{X} + \mathcal{X} \star_M \mathcal{B} = \mathcal{C}, \tag{29}
$$

where various cases for the coefficient tensors $\mathcal{A}$ and $\mathcal{B}$ are given by

$$
\begin{aligned}
&(a) \quad \mathcal{A} = \text{tenrand}([4\ 2\ 4\ 2]), && \mathcal{B} = \text{tenrand}([5\ 3\ 5\ 3]),\\
&(b) \quad \mathcal{A} = \text{tenrand}([6\ 4\ 6\ 4]), && \mathcal{B} = \text{tenrand}([8\ 5\ 8\ 5]),\\
&(c) \quad \mathcal{A} = \text{tenrand}([10\ 5\ 10\ 5]), && \mathcal{B} = \text{tenrand}([12\ 6\ 12\ 6]).
\end{aligned}
$$

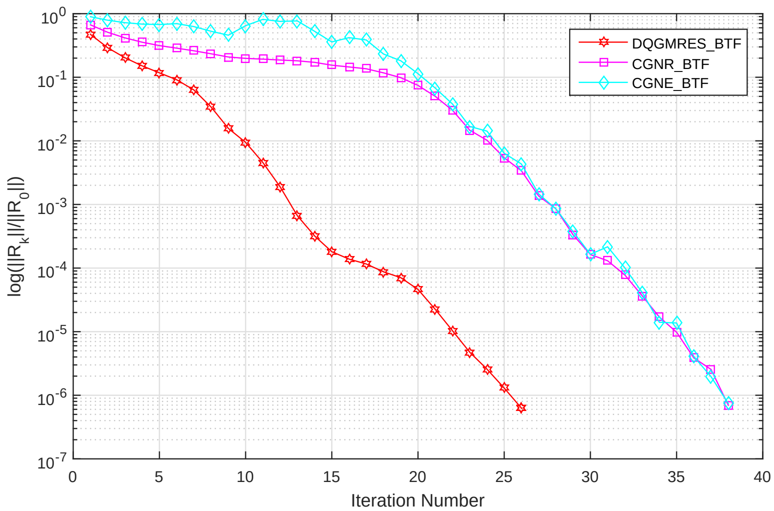

Figure 1: Comparison of convergence histories for Example 1 with Grid $N = 8$ and the truncation parameter $m = 10$.

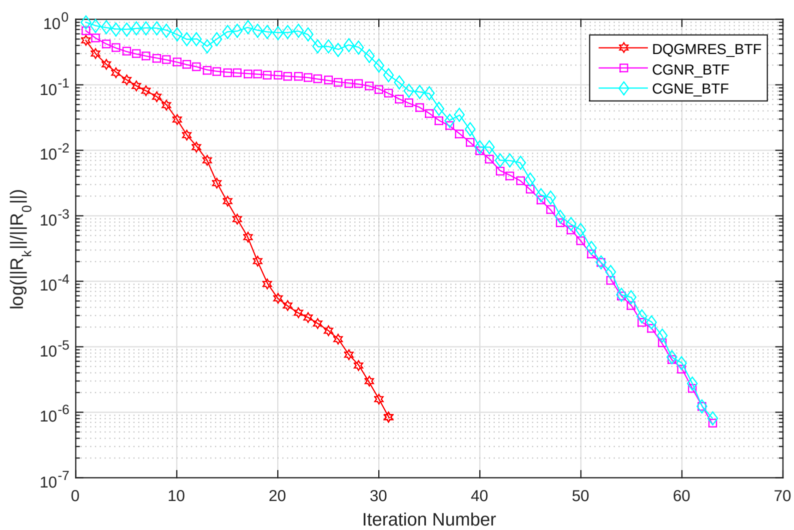

Figure 2: Comparison of convergence histories for Example 1 with Grid $N = 10$ and the truncation parameter $m = 10$.

The iteration number and CPU time of the CGNR_BTF, CGNE_BTF, and RNSD_BTF methods are reported in Table 2, compared with the proposed DQGMRES_BTF method with the truncation parameter $m = 5$ for cases (a)-(b). The corresponding convergence histories of the numerical results are depicted in Figure 3 with the truncation parameter $m = 5$ of the DQGMRES_BTF method. Theses results show that the DQGMRES_BTF algorithm is more effective and less expensive than the other methods. If we apply the DQGMRES_BTF algorithm, we obtain the more efficient approximate solution of Example 2. The result curves for case (c) are depicted in Figure 4. These results confirm the acceptable convergence of the proposed DQGMRES_BTF method. In other words, we can say that the proposed method is efficient for solving this type of tensor equation equipped with the Einstein product for small truncation parameters. The corresponding convergence histories of the numerical results for large-size $\mathcal{A} = tenrand([20\ 10\ 20\ 10])$ and $\mathcal{B} = tenrand([10\ 10\ 10\ 10])$ are depicted in Figure 5 with the truncation parameter $m = 5$ of the DQGMRES_BTF method and superior property of the DQGMRES_BTF method is observed compared to those of the CGNR_BTF method.

Table 2: Results of iteration number (Iter) and CPU time (Time) for Example 2.

Methods	RNSD_BTF		CGNE_BTF		CGNR_BTF		DQGMRES_BTF	
	Time	Iter	Time	Iter	Time	Iter	Time	Iter
case (a)	1.8173	†	0.9166	200	0.6954	151	0.1387	29
case (b)	17.7077	†	14.6543	416	6.8674	195	1.0540	29
case (c)	54.3223	†	41.6364	121	16.6119	303	2.8529	21

According to Definition 3, one can reduce the Sylvester tensor equation (29) to the associated Sylvester matrix equation and then solve it by the block QGMRES method [18]. We present the numerical results of applying the block QGMRES method to the reduced matrix equation and compare them with those of the DQGMRES_BTF method with the small truncation parameter $m = 5$. The advantage of the DQGMRES_BTF method is the short elapsed CPU time. We use the global conjugate gradient method [18] to solve the minimization problem in the block QGMRES method with inner max-iter=1000. As the conclusion, in the case $I_1 = 10, I_2 = 6, J_1 = 10, J_2 =$

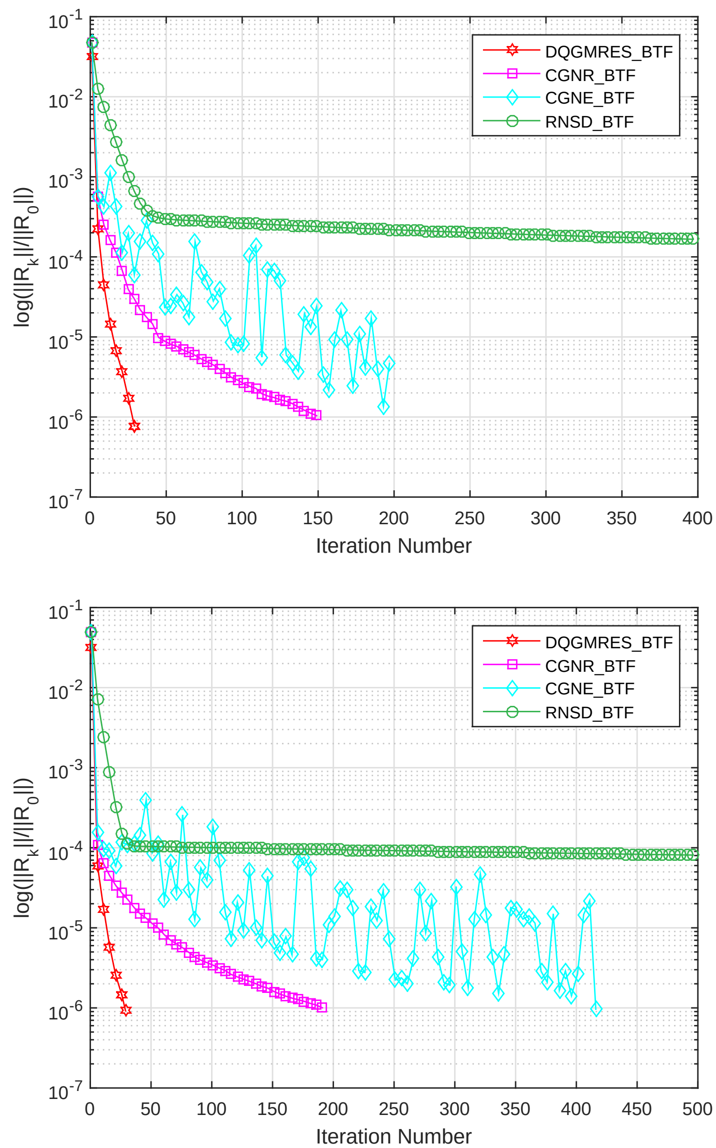

Figure 3: Comparison of convergence histories for case (a) (Up) and case (b) (Down) in Example 2 with the truncation parameter $m = 5$.

4, the results of Figure 6 (Up) have been obtained. It seems we have a better number of iterations for the block QGMRES, but the elapsed time is worse than the DQGMRES-BTF methods as depicted in Figure 6 (Down).

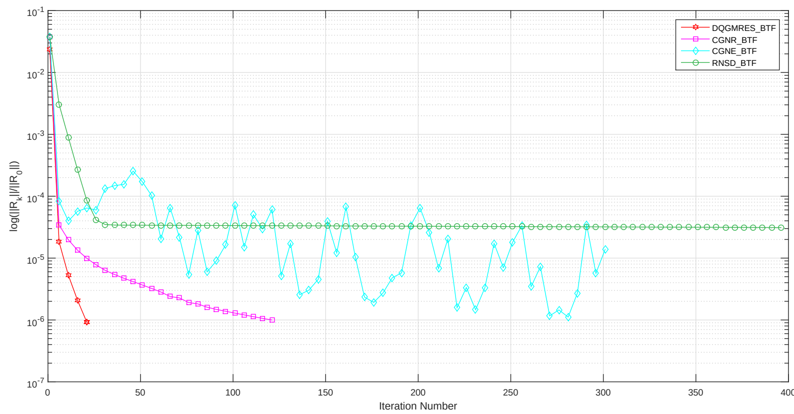

Figure 4: Comparison of convergence histories for case (c) in Example 2 with the truncation parameter $m = 5$.

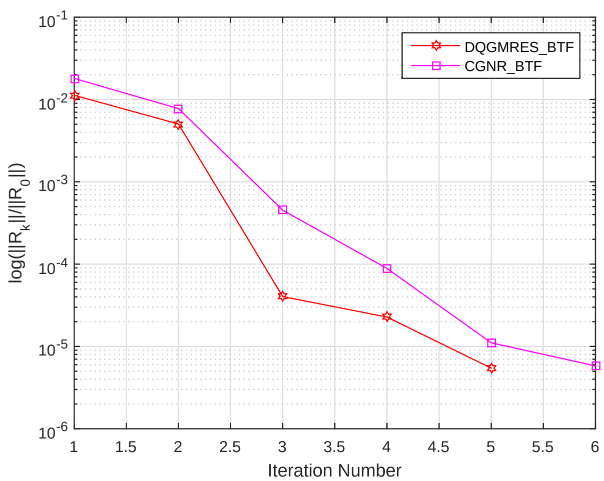

Figure 5: Comparison of convergence histories in Example 2 with the truncation parameter $m = 5$.

Example 3. Consider generalized Sylvester tensor equation

$$\mathcal{A} \star_N \mathcal{X} \star_M \mathcal{B} + \mathcal{C} \star_N \mathcal{X} \star_M \mathcal{D} = \mathcal{F},$$

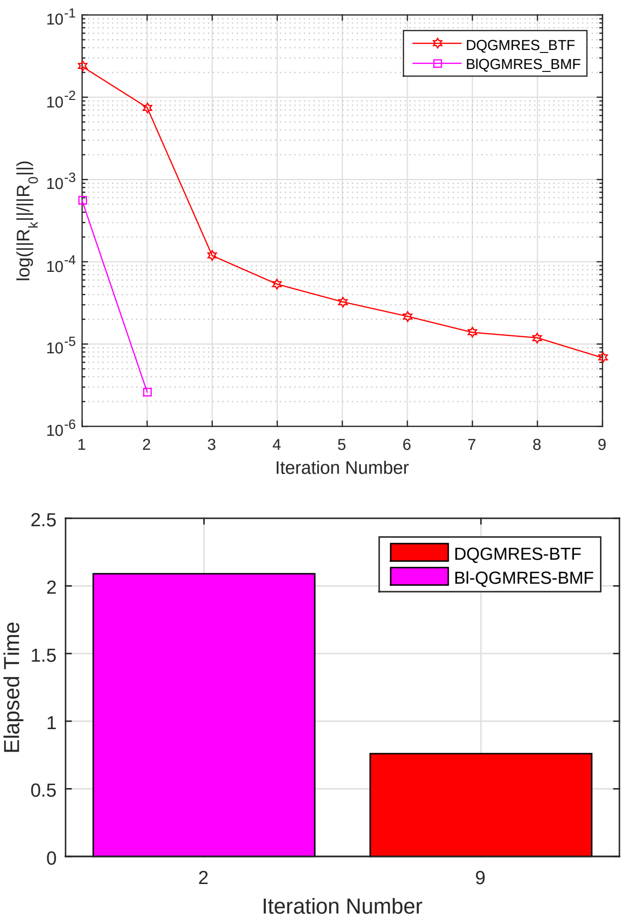

Figure 6: Comparison of the convergence behavior (Up) and the elapsed times (Down) for the block QGMRES and DQGMRES_BTF methods in Example 2 with the truncation parameter $m = 5$.

where $\mathcal{A} = \text{tenrand}([6\ 6\ 6\ 6])$, $\mathcal{B} = \text{tenrand}([8\ 8\ 8\ 8])$, $\mathcal{C} = \text{tenrand}([6\ 6\ 6\ 6])$, $\mathcal{D} = \text{tenrand}([8\ 8\ 8\ 8])$.

In Table 3, we report the numerical results of the iteration number and the CPU time of the CGNR_BTF and CGNE_BTF methods, compared to the proposed DQGMRES_BTF method with the different truncation parameters m. The convergence curves of the numerical results are depicted in Figure 7 with the truncation parameter $m = 5$ of the DQGMRES_BTF method. The effectiveness of the DQGMRES_BTF method and the less elapsed time are shown in Table 3 and Figure 7.

Table 3: Results of iteration number (Iter) and CPU time (Time) for Example 3 with various truncation parameters m in the DQGMRES_BTF method.

Methods	Time(Iter)
CGNR_BTF	11.8351(70)
CGNE_BTF	32.6890(191)
DQGMRES_BTF	
m=5	2.7033(16)
m=10	2.8392(16)
m=15	2.5292(15)

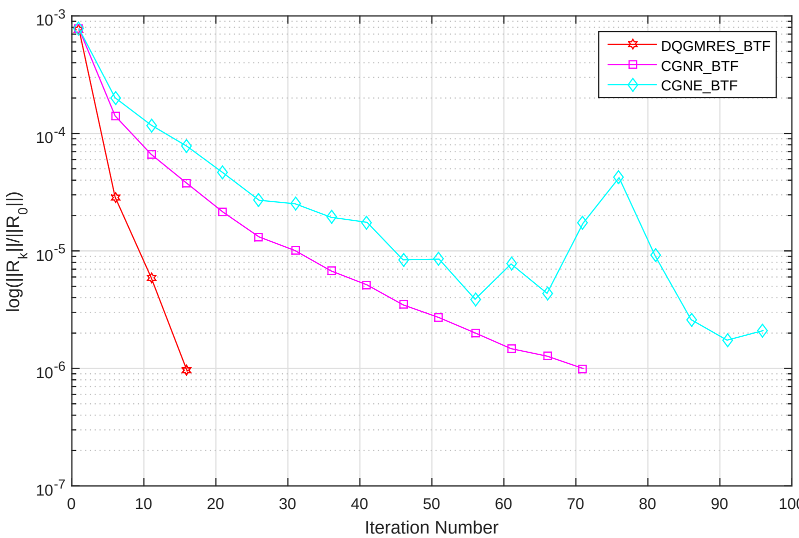

Figure 7: Comparison of convergence histories for Example 3 with the truncation parameter $m = 5$.

6 Conclusion

In this paper, we proposed an iterative method for solving generalized Sylvester tensor equations via the Einstein product using the tensor form of the QGMRES method. We present some useful results on tensor computations and propose a direct variant of the QGMRES method to utilize previous data and practical implementation of the method. Also, some results proved to illustrate a prior convergence behavior of the new method. In the numerical results of the experimental problems, we observed that the presented method has more efficiency and accuracy properties with low computational cost compared to the other tensor equation solvers such as CGNR, CGNE, and RNSD methods.

Acknowledgements

The author is very much indebted to the anonymous referees for their constructive suggestions and helpful comments, which greatly improved the original manuscript of this paper.

Data Availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Bader, B.W. and Kolda, T.G. *Efficient MATLAB computations with sparse and factored tensors*, SIAM J. Sci. Comput. 30(1) (2007), 205–231.

[2] Bader, B.W. and Kolda, G.T. *Tensor Toolbox for MATLAB, Version 3.6*, Available online at https://www.tensortoolbox.org., 2023.

[3] Ballani, J. and Grasedyck, L. *A projection method to solve linear systems in tensor form*, Numer. Linear Algebra Appl. 20 (2013), 27–43.

[4] Behera, R. and Mishra, D. *Further results on generalized inverses of tensors via the Einstein product*, Linear Multilinear Algebra. 65 (2017), 1662–1682.

[5] Beik, F.P.A. and Ahmadi-Asl, S. *Residual norm steepest descent based iterative algorithms for Sylvester tensor equations*, J. Math. Model. 2 (2015), 115–131.

[6] Beik, F.P.A., Movahed, F.S. and Ahmadi-Asl, S. *On the krylov subspace methods based on the tensor format for positive definite sylvester tensor equations*, Numer. Linear Algebra. Appl. 23 (2016), 444–466.

[7] Brazell, M., Li, N., Navasca, C. and Tamon, C. *Solving multilinear systems via tensor inversion*, SIAM J. Matrix Anal. Appl. 34 (2013), 542–570.

[8] Brown, P.N. and Hindmarsh, A.C. *Reduced storage methods in sti ODE systems*, Appl. Math. Comput. 31 (1989), 40.

[9] Chen, Z. and Lu, L.Z. *A projection method and Kronecker product precondetioner for solving Sylvester tensor equations*, Science China 55 (2012), 1281–1292.

[10] Dehdezi, E.K. *Iterative methods for solving Sylvester transpose-tensor equation* $\mathcal{A}\star_N \mathcal{X}\star_M \mathcal{B}+\mathcal{C}\star_M \mathcal{X}^T\star_N \mathcal{D}=\mathcal{E}$, Operations Research Forum. 2 (2021), 64.

[11] Dehdezi, E.K. and Karimi, S. *Extended conjugate gradient squared and conjugate residual squared methods for solving the generalized coupled Sylvester tensor equations*, T. I. Meas. Control. 43 (2021), 519–527.

[12] Dehdezi, E.K. and Karimi, S. *A gradient based iterative method and associated preconditioning technique for solving the large multilinear systems*, Calcolo. 58 (2021), 51.

[13] Ding, F. and Chen. T. *Gradient based iterative algorithms for solving a class of matrix equations*, IEEE Trans Autom Control. 50 (2005), 1216–1221.

[14] Ding, F. and Chen, T. *Iterative least squares solutions of coupled Sylvester matrix equations*, Syst. Control Lett. 54 (2005), 95–107.

[15] Ding, W. and Wei, Y. *Solving multi-linear system with $\mathcal{M}$-tensors*, J Sci Comput. 68 (2016), 689–715.

[16] Erfanifar, R. and Hajarian, M. *Several efficient iterative algorithms for solving nonlinear tensor equation $\mathcal{X}+\mathcal{A}^T\star_M\mathcal{X}^{-1}\star_N\mathcal{A}=\mathcal{I}$ with Einstein product*, Comput. Appl. Math. 43 (2024), 84.

[17] Graham, A. *Kronecker products and matrix calculus: with applications*, Courier Dover Publications, 2018.

[18] Guennouni, A.E., Jbilou, K. and Riquet, A.J. *Block Krylov subspace methods for solving large Sylvester equations*, Numeric. Algorithms. 29 (2002), 75–96.

[19] Grasedyck, L. *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing. 72 (2004), 247–265.

[20] Heyouni, M., Movahed, F.S. and Tajaddini, A. *A tensor format for the generalized hessenberg method for solving sylvester tensor equations*, J. Comput. Appl. Math. 377 (2020), 112878.

[21] Huang, B. and Ma, C. *Global least squares methods based on tensor form to solve a class of generalized Sylvester tensor equations*, Appl. Math. and Comput. 369 (2020), 124892.

[22] Huang, B. and Ma, C. *An iterative algorithm to solve the generalized Sylvester tensor equations*, Linear Multilinear Algebra. 68(6) (2020), 1175–1200.

[23] Huang, B., Xie, Y. and Ma, C. *Krylov subspace methods to solve a class of tensor equations via the Einstein product*, Numer. Linear Algebra Appl. 40(4) (2019), e2254.

[24] Jia, Z. *On IGMRES(q), incomplete generalized minimal residual methods for large unsymmetric linear systems*, Technical Report 94-047, Depart-

ment of Mathematics, University of Bielefeld, Sonderforschungsbereich 343, 1994. Last revision March, 1995.

[25] Kolda, T.G. and Bader, B.W. *Tensor decompositions and applications*, SIAM Rev. 51 (2009), 455–500.

[26] Lai, W.M., Rubin, D. and Krempl, E. *Introduction to continuum mechanics*, Oxford: Butterworth Heinemann, 2009.

[27] Li, B.W., Sun, Y.S. and Zhang, D.W. *Chebyshev collocation spectral methods for coupled radiation and conduction in a concentric spherical participating medium*, ASME J Heat Transfer. 131 (2009), 062701–62709.

[28] Li, B.W., Tian, S., Sun, Y.S. and 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. 229 (2010), 1198–1212.

[29] Li, T., Wang, Q.-W. and Zhang, X.-F. *A Modified Conjugate Residual Method and Nearest Kronecker Product Preconditioner for the Generalized Coupled Sylvester Tensor Equations*, Mathematics. 10 (2022), 1730.

[30] Liang, L., Zheng, B. and Zhao, R.J. *Tensor inversion and its application to the tensor equations with Einstein product*, Linear Multilinear Algebra. 67 (2018), 843–870.

[31] Malek, A. and Masuleh, S.H.M. *Mixed collocation-finite difference method for 3D microscopic heat transport problems, J. Comput. Appl. Math.* 217 (2008), 137–147.

[32] Malek, A., Bojdi, Z.K. and Golbarg, P.N.N. *Solving fully three-dimensional microscale dual phase lag problem using mixed-collocation finite difference discretization*, J. Heat Transfer. 134 (2012), 0945041–0945046.

[33] Masuleh, S.H.M. and Phillips, T.N. *Viscoelastic flow in an undulating tube using spectral methods*, Computers & Fluids 33 (2004), 10751095.

[34] Qi, L. and Luo, Z. *Tensor analysis: Spectral theory and special tensors*, SIAM, Philadelphia, 2017.

[35] Saad, Y. *Iterative methods for sparse linear systems*, PWS press, New York, 1995.

[36] Saad, Y. and Wu, K. *DQGMRES: a direct quasi-minimal residual algorithm based on incomplete orthogonalization*, Numeric. Linear Algebra Appl. 3(4) (1996), 329–343.

[37] Shi, X., Wei, Y. and Ling, S. *Backward error and perturbation bounds for high order Sylvester tensor equation*, Linear Multilinear Algebra. 61 (2013), 1436–1446.

[38] Sun, L., Zheng, B., Bu, C. and Wei, Y. *Moore-Penrose inverse of tensors via Einstein product*, Linear Multilinear Algebra. 64 (2016), 686–698.

[39] Tzou, D.V. *Macro to Micro Heat Transfer*, Taylor & Francis: Washington, 1996.

[40] Wang, Q.W. and Wang, X. *A system of coupled two-sided sylvester-type tensor equations over the quaternion algebra*, Taiwanese J. Math. 24 (2020), 1399–1416.

[41] Wang, Q.W. and Xu, X. *Iterative algorithms for solving some tensor equations*, Linear Multilinear Algebra. 67(7) (2019), 1325–1349.

[42] Wang, Q.W., Xu, X.J. and Duan, X.F. *Least squares solution of the quaternion sylvester tensor equation*, Linear Multilinear Algebra. 69 (2021), 104–130.

[43] Xie, M.Y. and Wang, Q.W. *Reducible solution to a quaternion tensor equation*, Front. Math. China. 15 (2020), 1047–1070.

[44] Zhang, X.-F. and Wang, Q.-W. *Developing iterative algorithms to solve Sylvester tensor equations*, Appl. Math. Comput. 409 (2021), 126403.

[45] Zhang, X.-F., Ding, W. and Li, T. *Tensor form of GPBiCG algorithm for solving the generalized Sylvester quaternion tensor equations*, J. Franklin Institute. 360 (2023), 5929–5946.