Numerical Algorithms (2025) 100:1123–1144
https://doi.org/10.1007/s11075-024-01982-x

RESEARCH

Randomized extended average block Kaczmarz method for inconsistent tensor equations under t-product

Liyuan An[1] · Kun Liang[1] · Han Jiao[1] · Qilong Liu[1]

Received: 27 July 2024 / Accepted: 11 November 2024 / Published online: 16 November 2024

Abstract
The Kaczmarz method is widely recognized as a mainstream technique for solving linear systems. To solve large inconsistent tensor equations under tensor t-product, we propose a tensor randomized extended average block Kaczmarz method. We prove theoretically that it converges to the least-squares norm solution of the equation as expected and demonstrate its convergence and performance through numerical experiments. Compared to existing methods, this approach reduces computation time for solving ill-conditioned equations and enhances convergence stability.

Keywords Tensor systems · Randomized Kaczmarz · T-product

1 Introduction

In recent years, the tensor product (t-product) [1] has been introduced as a linear operator of higher-order tensors. This avoids the process of simplifying tensors into matrices and effectively preserves the multidimensional structure of tensors. The t-product framework allows the extension of linear algebra tools to tensor algebra, thereby greatly promoting the understanding and application of tensors in decomposition [2–5] and completion [6–9] problems. Furthermore, the t-product plays an important role in image processing [10, 11] and signal processing [12, 13].

✉ Qilong Liu
qlliu@gznu.edu.cn

Liyuan An
anly0811@163.com

Kun Liang
kliang8629@163.com

Han Jiao
17385190210@163.com

[1] School of Mathematical Sciences, Guizhou Normal University, Guiyang 550025, People's Republic of China

In image processing [14], the problem of image deblurring can ultimately be formulated as solving the following linear tensor equation

$$\mathcal{A} * \bar{\mathcal{X}} = \bar{\mathcal{B}}, \tag{1}$$

where $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is the known blur tensor, $\bar{\mathcal{X}} \in \mathbb{R}^{N_2 \times K \times N_3}$ is the true image, and $\bar{\mathcal{B}} \in \mathbb{R}^{N_1 \times K \times N_3}$ is the image that has been blurred by $\mathcal{A}$. When (1) represents a consistent linear system, given the tensor $\mathcal{A}$ and $\bar{\mathcal{B}}$, a deblurred true image $\bar{\mathcal{X}}$ can be obtained by solving the equation. However, in practical applications, the image $\bar{\mathcal{B}}$ is often susceptible to noise (denoted as $\varepsilon \in \mathbb{R}^{N_1 \times K \times N_3}$), and the blur tensor $\mathcal{A}$ frequently exhibits ill-conditioned characteristics, leading to the formation of the following inconsistent tensor equation.

$$\mathcal{A} * \mathcal{X} = \mathcal{B}, \mathcal{B} = \bar{\mathcal{B}} + \varepsilon. \tag{2}$$

The ill-conditioned nature of $\mathcal{A}$, combined with the influence of noise ε, complicates the solution of (2). Additionally, because the equation is inconsistent, it does not possess an exact solution. Our objective is to obtain an image that is both deblurred and denoised by solving (2), serving as an approximation for the unattainable true image.

As we know, iterative methods are among the most effective approaches for solving large-scale linear systems. Among these iterative methods, the classic Kaczmarz method, proposed by Kaczmarz [15], has garnered significant attention due to its efficiency and simplicity, leading to its widespread application in fields such as image reconstruction and digital signal processing. To enhance the convergence rate of the classic Kaczmarz method, Stromhmer and Vershynin [16] introduced the idea of randomization, proposing the randomized Kaczmarz (RK) method and proving its exponential convergence. This theoretical proof fills a gap in the explanation of the RK method and provides valuable theoretical foundations for subsequent research on Kaczmarz methods. To further accelerate the convergence speed of the RK method, Needell and Tropp [17] introduced a block variant of the Kaczmarz method, known as the randomized block Kaczmarz method (RBK), which selects a submatrix of the coefficient matrix at each iteration, thus accelerating the convergence speed. The analysis indicates that the convergence performance of RBK varies with the rules of the block partitioning. In [18], the authors proposed a faster randomized block Kaczmarz (RABK) method, which is also a variant of the RK method. Unlike the RBK method, RABK effectively avoids the computation of the matrix pseudoinverse, greatly improving computational speed and making it more suitable for distributed computing.

There are numerous other variants of the Kaczmarz method, such as the greedy Kaczmarz and the block greedy Kaczmarz [19–23], among others. These methods have all improved the performance of the classic Kaczmarz method to varying degrees. Some of the methods mentioned above assume consistency for the system of equations. For linear systems with noise, the author proved in [24] that the RK method converges up to a convergence horizon around the least-squares solution. Following that Zouzias and Freris [25] proposed randomized extend Kaczmarz (REK) method to solve the inconsistent linear system proved that this method converges exponentially in the mean square to the least squares solution of the equation. Based on the REK method and the RABK method, in [26] Du et al. proposed a randomized extended average block Kaczmarz method (REABK) to solve inconsistent linear systems. This approach eliminates the necessity for the computation of the matrix pseudoinverse, thereby facilitating distributed computing. The experimental results demonstrate that this significantly decreases the computational time associated with the REBK method. In summary, the Kaczmarz method demonstrates excellent performance on both consistent and inconsistent linear systems. For further applications and analyses of the Kaczmarz method, please refer to references [27–30].

In the seminal article [31], Ma and Molitor applied the randomized Kaczmarz method to tensor equations based on t-product, introducing the tensor randomized Kaczmarz (TRK) method and theoretically proving its convergence. They made full use of the properties of tensor products to demonstrate that, in the Fourier domain, the TRK method is equivalent to the block Kaczmarz method for matrix systems. Subsequently, Chen and Qin [32] utilized the randomized Kaczmarz method to address tensor recovery problems and proposed the randomized regularized Kaczmarz method. Numerical experiments demonstrated that the Kaczmarz method performs well in various signal/image recovery and image striping tasks. For applications of the randomized Kaczmarz method in tensor recovery and completion problems, please refer to reference [33–36]. The authors of [37] introduced the tensor randomized block Kaczmarz (TRBK) method, which enhances accuracy beyond the TRK method. These methods are all proposed based on the assumption of equation consistency. Recently, for (2), Huang et al. [38] proposed a tensor randomized extended block Kaczmarz method (TREBK) to solve inconsistent equation, which breaks the constraint of equation consistency and can obtain a high-precision solution. However, TREBK requires computing the pseudoinverse of the tensor in each iteration, so it consumes a significant amount of computation time, as shown in Fig. 1.

Inspired by [26], we apply the concept of the average block Kaczmarz method from matrices to tensor equations, thereby avoiding the computation of the tensor pseudoinverse and ultimately reducing computational time. Consequently, we propose a tensor randomized extended average block Kaczmarz (TREABK) method, which uses a series of convex linear combinations of tensors to replace the process of calculating the pseudoinverse of tensors in the original method. Specifically, in each iteration, tensor only need for a simple multiplication and addition operations, and save the calculation time of the algorithm. Numerical experiments show that our proposed method is more time-saving and has more stable convergence than the existing methods.

The rest of the paper is organized as follows. Section 2 introduces some basic properties of symbols and tensors used in the sequel. In Section 3, TREABK algorithm is proposed and its convergence is proved. Section 4 describes numerical experiments and the results. The conclusion is written in Section 5.

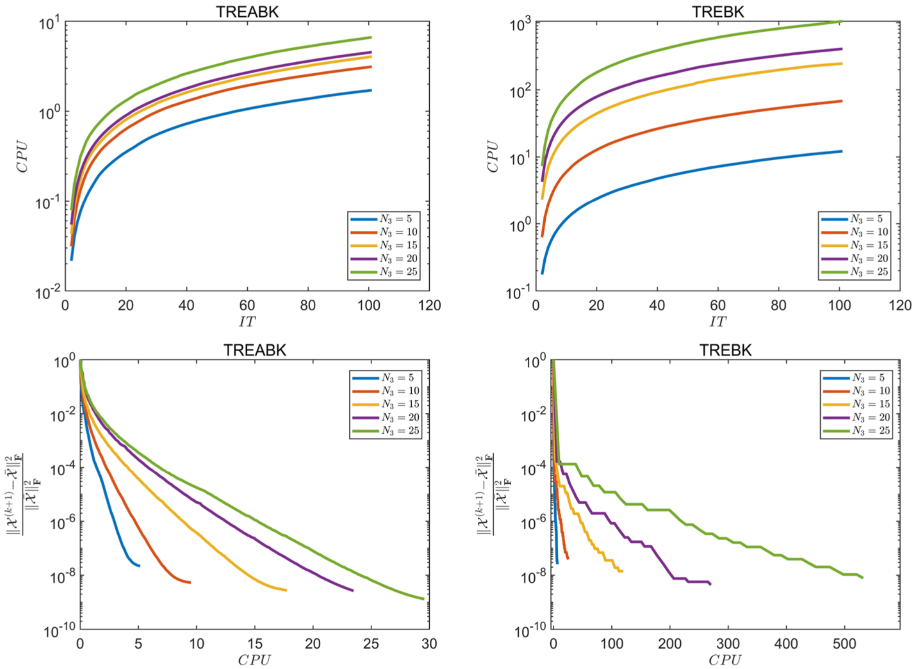


Fig. 1 Figure 1 illustrates the computational time and convergence accuracy of the TREABK and TREBK methods for solving (2) as the dimension of the equations changes. In (2), we set $N_1 = 200$, $N_2 = 200$, $K = 50$, and N_3 to be 5, 10, 15, 20, and 25, respectively, thereby generating five equations. The upper two figures show the time required for the TREABK and TREBK methods to complete 100 iterations. The lower two graphs display the time taken by both methods to achieve the same accuracy. From the upper two figures, we can observe that for solving equation systems with the same dimensions, the TREABK method requires significantly less time than the TREBK method. Furthermore, as N_3 increases, the computational time of the TREBK method grows much faster than that of the TREABK method

2 Notations and preliminaries

In this section, we mainly talk about notations and the properties of tensor that may be used in the sequel.

2.1 Notations

For an integer $m \geq 1$, let $[m] := \{1, 2, 3, \ldots, m\}$. Calligraphic letters are used to represent tensors. For a three-order tensor $\mathcal{A}$, $\mathcal{A}_{ijk}$ denotes the (i, j, k)th element of $\mathcal{A}$. $\mathcal{A}_{i,:,:}$, $\mathcal{A}_{:,i,:}$, and $\mathcal{A}_{:,:,i}$ used to represent the horizontal slices, lateral slices, and frontal slices of $\mathcal{A}$, respectively. Let the pseudoinverse of $\mathcal{A}$ be denoted as $\mathcal{A}^\dagger$. The minimum nonzero singular value of a matrix is denoted as $\sigma_{min}(\cdot)$. For a given random variable $\mathbf{x}$ and $\mathbf{y}$, denote $\mathbb{E}[\mathbf{x}]$ as the expected value of variable $\mathbf{x}$, and use $\mathbb{E}[\mathbf{x}|\mathbf{y}]$ to represent the expected value of the random variable $\mathbf{x}$ under the condition of random variable $\mathbf{y}$.

2.1.1 Preliminaries

In this part, we mainly introduce the basic definition and operation of tensor that will be used in the sequel. We follow the notations of tensor that use in [1, 3].

For a third-order tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, and letting $\mathbf{A}_i = \mathcal{A}_{:,:,i}$, then the block circulant matrix bcirc($\mathcal{A}$) of $\mathcal{A}$ is defined as follows

$$\mathrm{bcirc}(\mathcal{A}) := \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_{N_3} & \cdots & \mathbf{A}_2 \\ \mathbf{A}_2 & \mathbf{A}_1 & \cdots & \mathbf{A}_3 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{N_3} & \mathbf{A}_{N_3-1} & \cdots & \mathbf{A}_1 \end{bmatrix} \in \mathbb{R}^{N_1 N_3 \times N_2 N_3}.$$

In addition, we define the operator unfold($\cdot$) and its inversion fold($\cdot$) as follows

$$\mathrm{unfold}(\mathcal{A}) := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_{N_3} \end{bmatrix} \in \mathbb{R}^{N_1 N_3 \times N_2}, \ \mathrm{fold}\left(\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_{N_3} \end{bmatrix}\right) := \mathcal{A}.$$

Definition 1 [1] For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times K \times N_3}$, the t-product between $\mathcal{A}$ and $\mathcal{B}$ is a tensor with dimension $N_1 \times K \times N_3$,

$$\mathcal{A} * \mathcal{B} := \mathrm{fold}(\mathrm{bcirc}(\mathcal{A}) \cdot \mathrm{unfold}(\mathcal{B})).$$

Lemma 1 [32] *Assuming that $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times K \times N_3}$, then the t-product between $\mathcal{A}$ and $\mathcal{B}$ has the following properties,*

1. The separability in the first dimension

$$(\mathcal{A} * \mathcal{B})_{i,:,:} = \mathcal{A}_{i,:,:} * \mathcal{B}. \tag{3}$$

2. The component form of the product on the second dimension

$$\mathcal{A} * \mathcal{B} = \sum_{j=1}^{N_2} \mathcal{A}_{:,j,:} * \mathcal{B}_{j,:,:}. \tag{4}$$

Definition 2 [1] Let $\mathcal{A}^\top \in \mathbb{R}^{N_2 \times N_1 \times N_3}$ be the transpose of $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, which is obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through N_3.

Lemma 2 [32] *For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, it holds that*

$$\mathrm{bcirc}(\mathcal{A}^\top) = (\mathrm{bcirc}(\mathcal{A}))^\top. \tag{5}$$

$$(\mathcal{A}_{i,:,:})^\top = (\mathcal{A})^\top_{:,i,:}. \tag{6}$$

Remark 1 The establishment of (6) is due to the fact that $\mathcal{A}_{i,:,:} \in \mathbb{R}^{1\times N_2\times N_3}$ is a tensor that follows the tensor transposition rules described in Definition 2.

Definition 3 [1] For $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{N_1\times N_2\times N_3}$, the inner product between $\mathcal{A}$ and $\mathcal{B}$ defined as

$$\langle \mathcal{A}, \mathcal{B}\rangle := \sum_{i,j,k} \mathcal{A}_{ijk} \cdot \mathcal{B}_{ijk}.$$

Lemma 3 [32] *For* $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}, \mathcal{B} \in \mathbb{R}^{N_2\times K\times N_3}, \mathcal{C} \in \mathbb{R}^{N_1\times K\times N_3}$*, it holds that*

$$\langle \mathcal{A} * \mathcal{B}, \mathcal{C}\rangle = \left\langle \mathcal{B}, \mathcal{A}^\top * \mathcal{C} \right\rangle. \tag{7}$$

In this study, we will employ the Frobenius norm to assess the disparity between two tensors. To this end, we provide spectral norm, and Frobenius norm in the t-product sense.

Definition 4 [1] For $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$, its spectral norm, and Frobenius norm defined as

$$\begin{aligned} \|\mathcal{A}\|_2 &= \|\mathrm{bcirc}(\mathcal{A})\|_2, \\ \|\mathcal{A}\|_\mathrm{F} &= \sqrt{\langle \mathcal{A}, \mathcal{A}\rangle} = \sqrt{\textstyle\sum_{i,j,k} (\mathcal{A}_{ijk})^2}. \end{aligned} \tag{8}$$

Lemma 4 [32] *For* $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}, \mathcal{B} \in \mathbb{R}^{N_2\times K\times N_3}$*, it holds that*

$$\|\mathcal{A} * \mathcal{B}\|_\mathrm{F} \le \|\mathcal{A}\|_2 \|\mathcal{B}\|_\mathrm{F}. \tag{9}$$

Lemma 5 [39] *For* $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$ *and* $\mathcal{B} \in \mathbb{R}^{N_1\times K\times N_3}$*, we have*

$$\mathcal{A}^\top * \mathcal{A} * \mathcal{A}^\dagger * \mathcal{B} = \mathcal{A}^\top * \mathcal{B}.$$

Definition 5 [1] For $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$, the range of $\mathcal{A}$ definitions as

$$\mathrm{range}(\mathcal{A}) = \{\mathcal{A} * \mathcal{Y} : \mathcal{Y} \in \mathbb{R}^{N_2\times K\times N_3}\}.$$

Definition 6 [1] For $\mathcal{A} \in \mathbb{R}^{N_1\times N_2\times N_3}$, the null of $\mathcal{A}$ definitions as

$$\mathrm{null}(\mathcal{A}) = \{\mathcal{X} \in \mathbb{R}^{N_2\times K\times N_3} : \mathcal{A} * \mathcal{X} = 0\}.$$

3 The tensor randomized extended average block Kaczmarz algorithm

From Definitions 5 and 6, it is easy to verify that $\mathrm{range}(\mathcal{A})$ and $\mathrm{null}(\mathcal{A}^\top)$ are orthogonal complement spaces of each other. For (2), since it is inconsistent, we have $\mathcal{B} \notin \mathrm{range}(\mathcal{A})$. There must exist a tensor $\mathcal{Z}$ belonging to $\mathrm{null}(\mathcal{A}^\top)$ such that $\mathcal{B} = (\mathcal{B} - \mathcal{Z}) + \mathcal{Z}$, where $\mathcal{B} - \mathcal{Z}$ belongs to $\mathrm{range}(\mathcal{A})$. In this case, the system of equations $\mathcal{A}*\mathcal{X}=\mathcal{B}-\mathcal{Z}$ is a consistent system, and therefore it must have a solution. To solve (2) is equivalent to finding $\mathcal{Z}$ and $\mathcal{X}$ such that they satisfy $\mathcal{A}^\top * \mathcal{Z}=0$ and $\mathcal{A}*\mathcal{X}=\mathcal{B}-\mathcal{Z}$, respectively. The resulting $\mathcal{X}$ is the approximate solution to the original (2).

According to (3), $\mathcal{A}^\top * \mathcal{Z}=0$ can be reviewed as a system of tensor equations

$$\begin{cases} (\mathcal{A}^\top)_{1,:,:} * \mathcal{Z} = 0 \\ (\mathcal{A}^\top)_{2,:,:} * \mathcal{Z} = 0 \\ \qquad \cdots\cdots \\ (\mathcal{A}^\top)_{N_2,:,:} * \mathcal{Z} = 0 \end{cases}$$

consisting of N_2 equations. Building upon the work of [26], we propose a novel approach to solving $\mathcal{A}^\top * \mathcal{Z}=0$ that employs several convex combinations of RK as the update direction to yield a fast algorithm. Specifically,

$$\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} - \alpha \left(\sum_{\ell\in\mathcal{J}_j} \omega_\ell^k \frac{\mathcal{A}_{:,\ell,:} * (\mathcal{A}_{:,\ell,:})^\top * \mathcal{Z}^{(k)}}{||(\mathcal{A}_{:,\ell,:})^\top||_F^2} \right), \tag{10}$$

where $\omega_\ell^k = \frac{||(\mathcal{A}_{:,\ell,:})^\top||_F^2}{\|(\mathcal{A}_{:,\mathcal{J}_j,:})^\top\|_F^2}$.

Similarly, for solving $\mathcal{A}*\mathcal{X}=\mathcal{B}-\mathcal{Z}$ the specific expression is as follows:

$$\mathcal{X}^{(k+1)} = \mathcal{X}^{(k)} - \alpha \left(\sum_{\ell\in\mathcal{I}_i} \omega_\ell^k \frac{(\mathcal{A}_{\ell,:,:})^\top * \left(\mathcal{A}_{\ell,:,:} * \mathcal{X}^{(k)} - \mathcal{B}_{\ell,:,:} + \mathcal{Z}_{\ell,:,:}^{(k+1)}\right)}{||\mathcal{A}_{\ell,:,:}||_F^2} \right), \tag{11}$$

where $\omega_\ell^k = \frac{||\mathcal{A}_{\ell,:,:}||_F^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}$.

Remark 2 Initialize $\mathcal{Z}^{(0)}=\mathcal{B}$ and $\mathcal{X}^{(0)} \in \mathrm{range}(\mathcal{A}^\top)$. Using (10), we obtain $\mathcal{Z}^{(1)}$, and using (11), we obtain $\mathcal{X}^{(1)}$. This process continues until the termination condition is met, allowing us to obtain an approximate solution to the original (2).

Moreover, by harnessing (4) of Lemma 1, we are able to streamline (10) into the subsequent form.

$$\begin{aligned} \mathcal{Z}^{(k+1)} &= \mathcal{Z}^{(k)} - \alpha \left(\sum_{\ell\in\mathcal{J}_j} \frac{||(\mathcal{A}_{:,\ell,:})^\top||_F^2}{\|\ (\mathcal{A}_{:,\mathcal{J}_j,:})^\top\ \|_F^2} \frac{\mathcal{A}_{:,\ell,:} * (\mathcal{A}_{:,\ell,:})^\top * \mathcal{Z}^{(k)}}{||(\mathcal{A}_{:,\ell,:})^\top||_F^2} \right) \\ &= \mathcal{Z}^{(k)} - \alpha \left(\sum_{\ell\in\mathcal{J}_j} \frac{\mathcal{A}_{:,\ell,:} * (\mathcal{A}_{:,\ell,:})^\top * \mathcal{Z}^{(k)}}{\|\ (\mathcal{A}_{:,\mathcal{J}_j,:})^\top\ \|_F^2} \right) \\ &= \mathcal{Z}^{(k)} - \alpha \frac{\mathcal{A}_{:,\mathcal{J}_j,:} * (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \mathcal{Z}^{(k)}}{\|\ (\mathcal{A}_{:,\mathcal{J}_j,:})^\top\ \|_F^2}, \end{aligned}$$

this provides us with the updated formula for $\mathcal{Z}^{(k+1)}$ in Algorithm 1. Similarly, we simplify (11) to obtain the updated formula for $\mathcal{X}^{(k+1)}$ in Algorithm 1.

Algorithm 1 Tensor extend average block Kaczmarz (TREABK) method.

Let $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_s\}$ and $\{\mathcal{J}_1, \mathcal{J}_2, \cdots, \mathcal{J}_t\}$ be partitions of $[N_1]$ and $[N_2]$.
Let $\alpha > 0$. Initialize $\mathcal{Z}^{(0)} = \mathcal{B}$, $\mathcal{X}^{(0)} \in \text{range}(\mathcal{A}^\top)$.
for $k = 0, 1, 2, \ldots,$ maxiter **do**

Pick $j \in [t]$ with probability $\frac{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_F^2}{\|\mathcal{A}\|_F^2}$.

Set $\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} - \alpha \frac{\mathcal{A}_{:,\mathcal{J}_j,:} * (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \mathcal{Z}^{(k)}}{\|(\mathcal{A}_{:,\mathcal{J}_j,:})^\top\|_F^2}$.

Pick $i \in [s]$ with probability $\frac{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}{\|\mathcal{A}\|_F^2}$.

Set $\mathcal{X}^{(k+1)} = \mathcal{X}^{(k)} - \alpha \frac{(\mathcal{A}_{\mathcal{I}_i,:,:})^\top * (\mathcal{A}_{\mathcal{I}_i,:,:} * \mathcal{X}^{(k)} - \mathcal{B}_{\mathcal{I}_i,:,:} + \mathcal{Z}^{(k+1)}_{\mathcal{I}_i,:,:})}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}$.

end for

Next, we will theoretically prove that the solution obtained by TREABK converges to the least-squares norm solution of the (2). First, we show that the iterative sequence $\{\mathcal{Z}^{(k+1)}\}$ computed by Algorithm 1 is convergent. See Theorem 1 for the specific proof procedure.

Theorem 1 *For $\alpha \in (0, 2)$, denote $\mathcal{B}_\perp = \mathcal{B} - \mathcal{A} * \mathcal{A}^\dagger * \mathcal{B}$, $\beta_{\mathcal{J}} = \max_j \frac{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_2^2}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_F^2}$, then the iterative sequence $\{\mathcal{Z}^{(k+1)}\}$ generated by Algorithm 1 satisfies*

$$\mathbb{E}\left[\left\|\mathcal{Z}^{(k+1)} - \mathcal{B}_\perp\right\|_F^2\right] \le \rho^{(k+1)} \left\|\mathcal{Z}^{(0)} - \mathcal{B}_\perp\right\|_F^2, \tag{12}$$

where $\rho = 1 - \frac{(2\alpha - \alpha^2 \beta_{\mathcal{J}})\sigma_{\min}^2(\text{bcirc}(\mathcal{A}))}{\|\mathcal{A}\|_F^2}$.

Proof First of all, from the definition of $\mathcal{B}_\perp$ and Lemma 5, we can get

$$\mathcal{A}^\top * \mathcal{B}_\perp = \mathcal{A}^\top * \mathcal{B} - \mathcal{A}^\top * \mathcal{A} * \mathcal{A}^\dagger * \mathcal{B} \quad = \mathcal{A}^\top * \mathcal{B} - \mathcal{A}^\top * \mathcal{B} = 0.$$

Using the properties of Frobenius norm and inner product of tensor, we have the following estimates.

$$\begin{aligned} &\left\|\mathcal{Z}^{(k+1)} - \mathcal{B}_\perp\right\|_F^2 \\ &= \left\|\mathcal{Z}^{(k)} - \mathcal{B}_\perp - \frac{\alpha}{\left\|\mathcal{A}_{:,\mathcal{J}_j,:}\right\|_F^2} \mathcal{A}_{:,\mathcal{J}_j,:} * (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_\perp\right)\right\|_F^2 \end{aligned}$$

$$
\begin{aligned}
&= \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \frac{2\alpha}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2} \left\| (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2 \\
&\quad + \frac{\alpha^2}{\left\| \mathcal{A}_{:,\mathcal{J}_j,:} \right\|_{\mathrm{F}}^4} \left\| \mathcal{A}_{:,\mathcal{J}_j,:} * (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2 \\
&\le \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \frac{2\alpha}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2} \left\| (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2 \\
&\quad + \frac{\alpha^2 \beta_{\mathcal{J}}}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2} \left\| (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2 \\
&= \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \frac{\left\| (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2},
\end{aligned}
\tag{13}
$$

where $\beta_{\mathcal{J}} = \max_j \frac{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_2^2}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2}$.

Let $\mathbb{E}_{(k)}[\cdot]$ denote the conditional expectation conditioned on $\mathcal{Z}^{(k)}$, taking the expectation for (13), we obtain

$$
\begin{aligned}
&\mathbb{E}_{(k)} \left[\left\| \mathcal{Z}^{(k+1)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 \right] \\
&\le \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \mathbb{E} \left[\frac{\left\| (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2} \right] \\
&= \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \sum_{j=1}^{t} \frac{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2}{\|\mathcal{A}\|_{\mathrm{F}}^2} \frac{\left\| (\mathcal{A}_{:,\mathcal{J}_j,:})^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2}{\|\mathcal{A}_{:,\mathcal{J}_j,:}\|_{\mathrm{F}}^2} \\
&= \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \frac{\left\| \mathcal{A}^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2}{\|\mathcal{A}\|_{\mathrm{F}}^2} \\
&\le \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \frac{\left\| \mathcal{A}^\top \right\|_2^2 \left\| \left(\mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right) \right\|_{\mathrm{F}}^2}{\|\mathcal{A}\|_{\mathrm{F}}^2} \\
&= \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 - \frac{\left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \left\| \mathrm{bcirc} \left(\mathcal{A}^\top \right) \right\|_2^2}{\|\mathcal{A}\|_{\mathrm{F}}^2} \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 \\
&= \left[1 - \frac{\left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \|\mathrm{bcirc}(\mathcal{A})\|_2^2}{\|\mathcal{A}\|_{\mathrm{F}}^2} \right] \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 \\
&\le \left[1 - \frac{\left(2\alpha - \alpha^2 \beta_{\mathcal{J}} \right) \sigma_{\min}^2 \left(\mathrm{bcirc}(\mathcal{A}) \right)}{\|\mathcal{A}\|_{\mathrm{F}}^2} \right] \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 \\
&\le \rho \left\| \mathcal{Z}^{(k)} - \mathcal{B}_{\perp} \right\|_{\mathrm{F}}^2 .
\end{aligned}
$$

Specifically, the validity of the second equation is attributed to the application of (4) in Lemma 1 and (8) in Definition 4, whereas the fourth equation holds true due to (6) in Lemma 2.

By utilizing the properties of expectations, it can be obtained that

$$\begin{aligned}\mathbb{E}\left[\left\|\mathcal{Z}^{(k+1)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}\right]&=\mathbb{E}\left[\mathbb{E}_{(k)}\left[\left\|\mathcal{Z}^{(k+1)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}\right]\right]\\&\leq\rho\mathbb{E}\left[\left\|\mathcal{Z}^{(k)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}\right]\\&\leq\rho^{(k+1)}\left\|\mathcal{Z}^{(0)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}.\end{aligned}$$

The proof is complete. □

Theorem 2 *Let $\bar{\mathcal{X}}=\mathcal{A}^{\dagger}*\mathcal{B}$ be the least-squares solution of the tensor equations (2), $\mathcal{X}^{(k+1)}$ is the sequence of solutions generated by Algorithm 1. Let $\beta_{\mathcal{I}}=\max_i\frac{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_2^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_{\mathrm{F}}^2}$, for any $\delta>0$, it holds*

$$\begin{aligned}\mathbb{E}\left[\left\|\mathcal{X}^{(k+1)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}\right]\leq&\left(1+\frac{1}{\delta}\right)\frac{\alpha^{2}\beta_{\mathcal{I}}\left\|\mathcal{Z}^{(0)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}}{\|\mathcal{A}\|_{\mathrm{F}}^{2}}\sum_{l=0}^{k+1}\left[\rho^{k+1-l}+(1+\delta)^{l}\eta^{l}\right]\\&+(1+\delta)^{k+1}\eta^{k+1}\left\|\mathcal{X}^{(0)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2},\end{aligned}$$

where $\rho=1-\frac{(2\alpha-\alpha^2\beta_{\mathcal{J}})\sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}))}{\|\mathcal{A}\|_{\mathrm{F}}^2}$, $\eta=1-\frac{(2\alpha-\alpha^2\beta_{\mathcal{I}})\sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}))}{\|\mathcal{A}\|_{\mathrm{F}}^2}$.

Proof Firstly, construct two auxiliary formulas based on the iterative update formula.

$$\begin{aligned}\mathcal{X}^{(k+1)}&=\mathcal{X}^{(k)}-\frac{\alpha}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_{\mathrm{F}}^2}\left(\mathcal{A}_{\mathcal{I}_i,:,:}\right)^{\top}*\left(\mathcal{A}_{\mathcal{I}_i,:,:}*\mathcal{X}^{(k)}-\mathcal{B}_{\mathcal{I}_i,:,:}+\mathcal{Z}_{\mathcal{I}_i,:,:}^{(k+1)}\right),\\\hat{\mathcal{X}}^{(k+1)}&=\mathcal{X}^{(k)}-\frac{\alpha}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_{\mathrm{F}}^2}(\mathcal{A}_{\mathcal{I}_i,:,:})^{\top}*\mathcal{A}_{\mathcal{I}_i,:,:}*\left(\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right),\end{aligned}$$

then we have

$$\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}=\frac{\alpha}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_{\mathrm{F}}^2}(\mathcal{A}_{\mathcal{I}_i,:,:})^{\top}*\left(\mathcal{B}_{\mathcal{I}_i,:,:}-\mathcal{A}_{\mathcal{I}_i,:,:}*\bar{\mathcal{X}}-\mathcal{Z}_{\mathcal{I}_i,:,:}^{(k+1)}\right),\quad(14)$$

$$\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}=\mathcal{X}^{(k)}-\bar{\mathcal{X}}-\frac{\alpha}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_{\mathrm{F}}^2}(\mathcal{A}_{\mathcal{I}_i,:,:})^{\top}*\mathcal{A}_{\mathcal{I}_i,:,:}*\left(\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right).\quad(15)$$

Taking Frobenius norm both sides of (14), we obtain

$$\left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_{\mathrm{F}}^{2}$$

$$
\begin{aligned}
&= \frac{\alpha^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^4} \left\| (\mathcal{A}_{\mathcal{I}_i,:,:})^\top * \left(\mathcal{B}_{\mathcal{I}_i,:,:} - \mathcal{A}_{\mathcal{I}_i,:,:} * \bar{\mathcal{X}} - \mathcal{Z}^{(k+1)}_{\mathcal{I}_i,:,:} \right) \right\|_F^2 \\
&\leq \frac{\alpha^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2} \frac{\|(\mathcal{A}_{\mathcal{I}_i,:,:})^\top\|_2^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2} \left\| \mathcal{B}_{\mathcal{I}_i,:,:} - \mathcal{A}_{\mathcal{I}_i,:,:} * \bar{\mathcal{X}} - \mathcal{Z}^{(k+1)}_{\mathcal{I}_i,:,:} \right\|_F^2 \\
&\leq \frac{\alpha^2 \beta_{\mathcal{I}}}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2} \left\| \mathcal{B}_{\mathcal{I}_i,:,:} - \mathcal{A}_{\mathcal{I}_i,:,:} * \bar{\mathcal{X}} - \mathcal{Z}^{(k+1)}_{\mathcal{I}_i,:,:} \right\|_F^2 .
\end{aligned}
\tag{16}
$$

Taking expectation to (16), we obtain

$$
\begin{aligned}
&\mathbb{E}_{(k)} \left[\left\| \mathcal{X}^{(k+1)} - \hat{\mathcal{X}}^{(k+1)} \right\|_F^2 \right] \\
&\leq \sum_i^s \frac{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}{\|\mathcal{A}\|_F^2} \frac{\alpha^2 \beta_{\mathcal{I}}}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2} \left\| \mathcal{B}_{\mathcal{I}_i,:,:} - \mathcal{A}_{\mathcal{I}_i,:,:} * \bar{\mathcal{X}} - \mathcal{Z}^{(k+1)}_{\mathcal{I}_i,:,:} \right\|_F^2 \\
&\leq \frac{\alpha^2 \beta_{\mathcal{I}}}{\|\mathcal{A}\|_F^2} \left\| \mathcal{B} - \mathcal{A} * \bar{\mathcal{X}} - \mathcal{Z}^{(k+1)} \right\|_F^2 \\
&= \frac{\alpha^2 \beta_{\mathcal{I}}}{\|\mathcal{A}\|_F^2} \left\| \mathcal{B}_\perp - \mathcal{Z}^{(k+1)} \right\|_F^2 \\
&\leq \frac{\alpha^2 \beta_{\mathcal{I}}}{\|\mathcal{A}\|_F^2} \rho \left\| \mathcal{Z}^{(k)} - \mathcal{B}_\perp \right\|_F^2 ,
\end{aligned}
$$

where $\beta_{\mathcal{I}} = \max_i \frac{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_2^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}$. Then

$$
\begin{aligned}
&\mathbb{E} \left[\left\| \mathcal{X}^{(k+1)} - \hat{\mathcal{X}}^{(k+1)} \right\|_F^2 \right] \\
&= \mathbb{E} \left[\mathbb{E}_{(k)} \left[\left\| \mathcal{X}^{(k+1)} - \hat{\mathcal{X}}^{(k+1)} \right\|_F^2 \right] \right] \\
&\leq \frac{\alpha^2 \beta_{\mathcal{I}}}{\|\mathcal{A}\|_F^2} \mathbb{E} \left[\left\| \mathcal{B}_\perp - \mathcal{Z}^{(k+1)} \right\|_F^2 \right] \\
&\leq \frac{\alpha^2 \beta_{\mathcal{I}}}{\|\mathcal{A}\|_F^2} \rho^{(k+1)} \left\| \mathcal{Z}^{(0)} - \mathcal{B}_\perp \right\|_F^2 .
\end{aligned}
$$

Taking Frobenius norm both sides of (15), we obtain

$$
\begin{aligned}
&\left\| \hat{\mathcal{X}}^{(k+1)} - \bar{\mathcal{X}} \right\|_F^2 \\
&= \left\| \mathcal{X}^{(k)} - \bar{\mathcal{X}} \right\|_F^2 - \frac{2\alpha}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2} \left\| \mathcal{A}_{\mathcal{I}_i,:,:} * \left(\mathcal{X}^{(k)} - \bar{\mathcal{X}} \right) \right\|_F^2
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\alpha^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^4}\left\|(\mathcal{A}_{\mathcal{I}_i,:,:})^\top * \mathcal{A}_{\mathcal{I}_i,:,:} * \left(\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right)\right\|_F^2\\
\leq& \left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_F^2-\frac{2\alpha}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}\left\|\mathcal{A}_{\mathcal{I}_i,:,:} * (\mathcal{X}^{(k)}-\bar{\mathcal{X}})\right\|_F^2\\
&+\frac{\alpha^2\left\|(\mathcal{A}_{\mathcal{I}_i,:,:})^\top\right\|_2^2}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^4}\left\|\mathcal{A}_{\mathcal{I}_i,:,:} * (\mathcal{X}^{(k)}-\bar{\mathcal{X}})\right\|_F^2\\
\leq& \left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_F^2-\frac{2\alpha-\alpha^2\beta_{\mathcal{I}}}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}\left\|\mathcal{A}_{\mathcal{I}_i,:,:} * \left(\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right)\right\|_F^2.
\end{aligned}
\tag{17}
$$

Taking the condition expectation of (17), then it holds that

$$
\begin{aligned}
&\mathbb{E}_{(k)}\left[\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_F^2\right]\\
\leq& \left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_F^2-\frac{2\alpha-\alpha^2\beta_{\mathcal{I}}}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}\left\|\mathcal{A}_{\mathcal{I}_i,:,:} * \left(\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right)\right\|_F^2\\
=& \left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_F^2-\sum_i^s\frac{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}{\|\mathcal{A}\|_F^2}\frac{\left(2\alpha-\alpha^2\beta_{\mathcal{I}}\right)}{\|\mathcal{A}_{\mathcal{I}_i,:,:}\|_F^2}\left\|\mathcal{A}_{\mathcal{I}_i,:,:} * \left(\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right)\right\|_F^2\\
=& \left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_F^2-\frac{\left(2\alpha-\alpha^2\beta_{\mathcal{I}}\right)}{\|\mathcal{A}\|_F^2}\left\|\mathcal{A} * \left(\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right)\right\|_F^2\\
\leq& \left(1-\frac{\left(2\alpha-\alpha^2\beta_{\mathcal{I}}\right)\sigma_{\min}^2(\mathcal{A})}{\|\mathcal{A}\|_F^2}\right)\left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_F^2.
\end{aligned}
\tag{18}
$$

Taking the expectation of (18) yields

$$
\mathbb{E}\left[\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_F^2\right] \leq \eta\mathbb{E}\left[\left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_F^2\right].
$$

Next, we calculate the error between $\mathcal{X}^{(k+1)}$ and $\bar{\mathcal{X}}$ under Frobenius norm

$$
\begin{aligned}
&\left\|\mathcal{X}^{(k+1)}-\bar{\mathcal{X}}\right\|_F^2\\
=& \left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}+\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_F^2\\
\leq& \left(\left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_F+\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_F\right)^2\\
=& \left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_F^2+\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_F^2\\
&+2\left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_F\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_F\\
=& \left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_F^2+\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_F^2
\end{aligned}
$$

$$
\begin{aligned}
&+2\frac{1}{\delta}\left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_{\mathrm{F}}\delta\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}\\
&\leq\left(1+\frac{1}{\delta}\right)\left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_{\mathrm{F}}^{2}+(1+\delta)\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}.
\end{aligned}
\tag{19}
$$

Finally, taking the expected value of (19) yields an upper bound on the expected error.

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\mathcal{X}^{(k+1)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}\right]\\
&\leq\left(1+\frac{1}{\delta}\right)\mathbb{E}\left[\left\|\mathcal{X}^{(k+1)}-\hat{\mathcal{X}}^{(k+1)}\right\|_{\mathrm{F}}^{2}\right]+(1+\delta)\,\mathbb{E}\left[\left\|\hat{\mathcal{X}}^{(k+1)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}\right]\\
&\leq\left(1+\frac{1}{\delta}\right)\frac{\alpha^{2}\beta_{\mathcal{I}}}{\|\mathcal{A}\|_{\mathrm{F}}^{2}}\rho^{k+1}\left\|\mathcal{Z}^{(0)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}+(1+\delta)\,\eta\mathbb{E}\left[\left\|\mathcal{X}^{(k)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}\right]\\
&\leq\left(1+\frac{1}{\delta}\right)\frac{\alpha^{2}\beta_{\mathcal{I}}}{\|\mathcal{A}\|_{\mathrm{F}}^{2}}\rho^{k+1}\left\|\mathcal{Z}^{(0)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}+(1+\delta)\,\eta\\
&\quad\cdot\left[\left(1+\frac{1}{\delta}\right)\frac{\alpha^{2}\beta_{\mathcal{I}}}{\|\mathcal{A}\|_{\mathrm{F}}^{2}}\rho^{k}\left\|\mathcal{Z}^{(0)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}+(1+\delta)\eta\mathbb{E}\left[\left\|\mathcal{X}^{(k-1)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}\right]\right]\\
&=\left(1+\frac{1}{\delta}\right)\frac{\alpha^{2}\beta_{\mathcal{I}}\left\|\mathcal{Z}^{(0)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}}{\|\mathcal{A}\|_{\mathrm{F}}^{2}}\left[\rho^{k+1}+(1+\delta)\eta\rho^{k}\right]\\
&\quad+(1+\delta)^{2}\,\eta^{2}\mathbb{E}\left[\left\|\mathcal{X}^{(k-1)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}\right]\\
&\leq\cdots\\
&\leq\left(1+\frac{1}{\delta}\right)\frac{\alpha^{2}\beta_{\mathcal{I}}\left\|\mathcal{Z}^{(0)}-\mathcal{B}_{\perp}\right\|_{\mathrm{F}}^{2}}{\|\mathcal{A}\|_{\mathrm{F}}^{2}}\sum_{l=0}^{k+1}\left[\rho^{k+1-l}\,(1+\delta)^{l}\,\eta^{l}\right]\\
&\quad+(1+\delta)^{k+1}\,\eta^{k+1}\left\|\mathcal{X}^{(0)}-\bar{\mathcal{X}}\right\|_{\mathrm{F}}^{2}.
\end{aligned}
$$

The proof is complete. □

4 Numerical experiments

In this section, numerical experiments are used to demonstrate the effectiveness of the TREABK method by comparing it with the TRBK [37], TRK [31], and TREBK [38] methods. We use tensor t-product toolbox [40] in our compution. All experiments were simulated in MATLAB R2021b with Intel(R) Core(TM) i5-8265U CPU with 8G memory.

For equation

$$\mathcal{A}*\mathcal{X}=\mathcal{B},\mathcal{B}=\bar{\mathcal{B}}+\varepsilon.$$

The noise part ε, it is randomly generated by the "randn" function in MATLAB. The relative error of the solution is represented by the following equation

$$Error = \frac{\|\mathcal{X}^{(k+1)} - \bar{\mathcal{X}}\|_{\mathrm{F}}^2}{\|\bar{\mathcal{X}}\|_{\mathrm{F}}^2},$$

where $\bar{\mathcal{X}} = \mathcal{A}^{\dagger} * \mathcal{B}$.

We use IT to represent the number of iterations of the algorithm and CPU to represent the running time of the algorithm. For the block methods, we assume that the subsets $\{\mathcal{I}_i\}_{i=1}^{s-1}$ of $[N_1]$ and $\{\mathcal{J}_j\}_{j=1}^{t-1}$ of $[N_2]$ have size τ_1 and τ_2, respectively. We consider the partition $\{\mathcal{I}_i\}_{i=1}^{s}$:

$$\mathcal{I}_i = \{(i-1)\tau_1 + 1, (i-1)\tau_1 + 2, \ldots, i\tau_1\}, i = 1, 2, \ldots, s-1$$
$$\mathcal{I}_s = \{(s-1)\tau_1 + 1, (s-1)\tau_1 + 2, \ldots, m\}, |\mathcal{I}_s| \leq \tau_1,$$

and the partition $\{\mathcal{J}_j\}_{j=1}^{t}$:

$$\mathcal{J}_j = \{(j-1)\tau_2 + 1, (j-1)\tau_2 + 2, \ldots, j\tau_2\}, j = 1, 2, \ldots, t-1$$
$$\mathcal{J}_t = \{(t-1)\tau_2 + 1, (t-1)\tau_2 + 2, \ldots, n\}, |\mathcal{J}_t| \leq \tau_2.$$

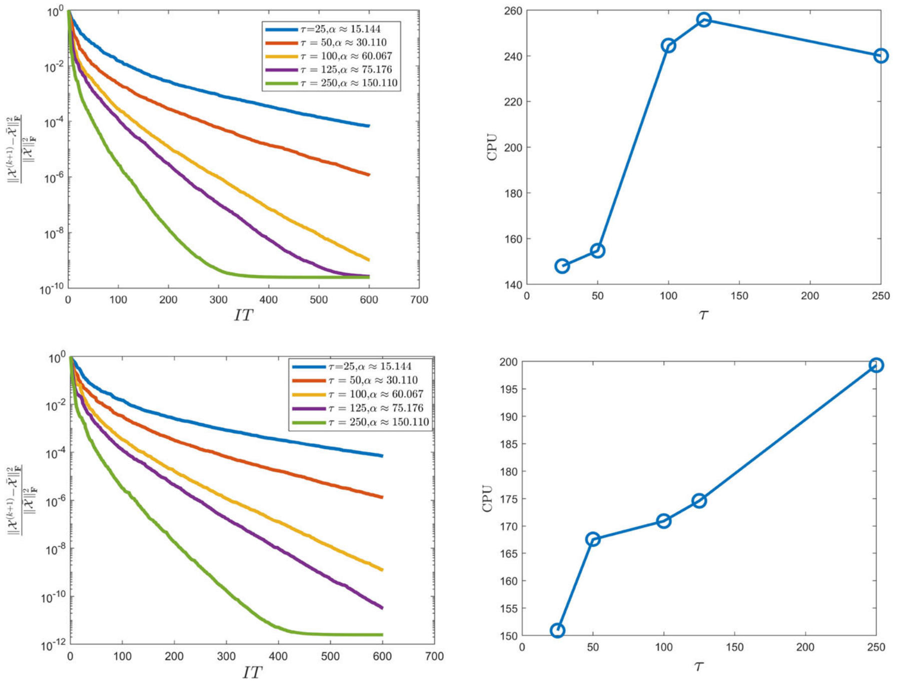


Fig. 2 The error and CPU of TREABK with different block sizes for a inconsistent tensor equation with different noise level 0.1 and 0.01. Upper: noise level is 0.1. Lower: noise level with 0.01

4.1 Block parameter analysis

In this experiment, we primarily deliberate on the impact of block size upon the TREABK method. We used MATLAB to randomly generate tensor $\mathcal{A} \in \mathbb{R}^{500\times300\times20}$ and tensor $\mathcal{X} \in \mathbb{R}^{300\times100\times20}$, let $\mathcal{B} \in \mathbb{R}^{500\times100\times20}$ with noise levels set to 0.1 and 0.01 respectively, and the maximum number of iterations prescribed as 600. Concerning the block sizes of $\mathcal{A}$, let $\tau_2 = 100$, τ_1 takes the values of 10, 20, 30, 40, and 50 respectively. In Fig. 2, we plotted the graphs of time and error required for solving equations of different blocks.

From Fig. 2, it is evident that an elevated value of τ_1 corresponds to an enhanced degree of solution accuracy. However, this is accompanied by an increased computational time.

4.2 Dense

We use TREABK, TREBK, TRBK, and TRK to solve the equation with $\mathcal{A} \in \mathbb{R}^{200\times50\times50}$ and $\mathcal{B} \in \mathbb{R}^{200\times50\times50}$, let $IT = 300$ and show the error and the time required to achieve the same accuracy in Fig. 3.

It can be observed from Fig. 3 that the convergence accuracy of TREABK, TREBK, and TRBK methods are higher than that of TRK method. The two error diagrams on the left side of Fig. 3 also show that the number of iterations required for TREBK

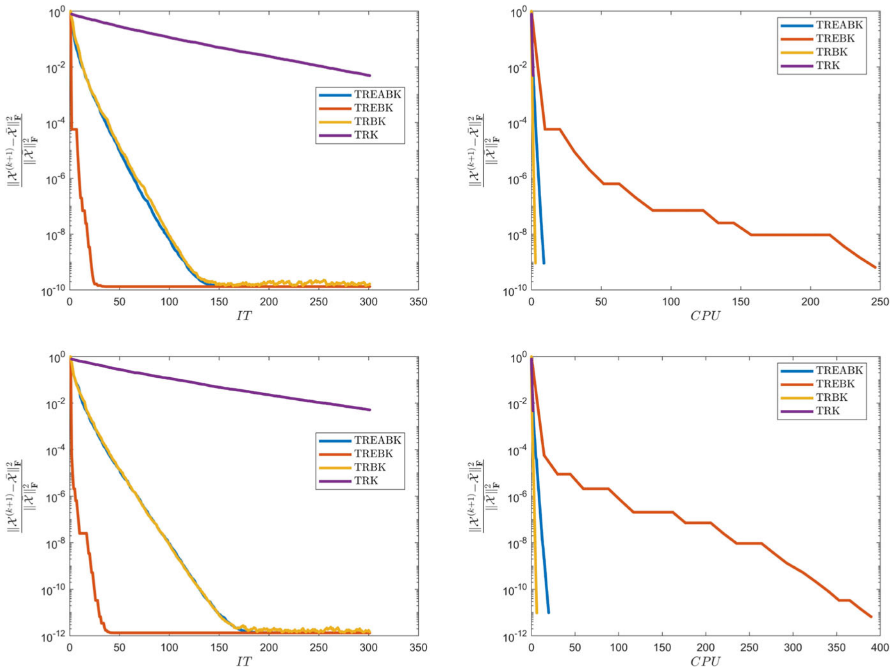


Fig. 3 The error and CPU of TREABK, TREBK, TRBK and TRK to solve a inconsistent tensor equation which size is $200 \times 50 \times 50$ and the noise level is 0.1 and 0.01, respectively. Upper: noise level is 0.1. Lower: noise level is 0.01

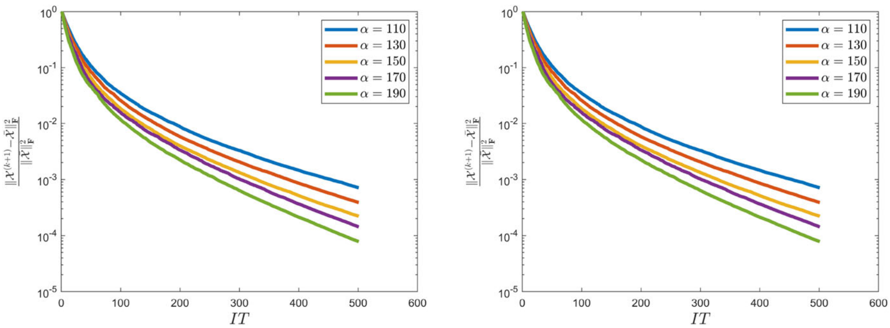


Fig. 4 The impact of parameters on the convergence of the TREABK method. Left side: noise level is 0.1. Right side: noise level is 0.01

method to converge is less than TREABK and TRBK methods, but when the three methods reach the specified accuracy, as shown in the two diagrams on the right, TREBK obviously takes more time. Finally, we can also see that the convergence curve of TRBK method will fluctuate during the whole convergence process, so the convergence of TREABK method is more stable than that of TRBK method.

Remark 3 In fact, the TREABK method is equivalent to the TRBK method twice in each iteration, so the computation time is slightly slower than the TRBK method.

4.3 The influence of the parameter α

In this section, we primarily investigate the influence of the parameter α on the convergence of the TREABK method through experiments. In (2), we set $N_1 = 400$, $N_2 = 200$, $N_3 = 20$, and $K = 50$. The number of iterations is 500, and the size of each block is fixed at $\tau_1 = 80$, $\tau_2 = 50$. The noise levels are set to 0.1 and 0.01, respectively. The experimental results are illustrated in Fig. 4, where the left image represents the case with a noise level of 0.1, and the right image corresponds to a noise level of 0.01. By examining these figures, we can observe that the convergence accuracy of the TREABK method varies with changes in the parameter.

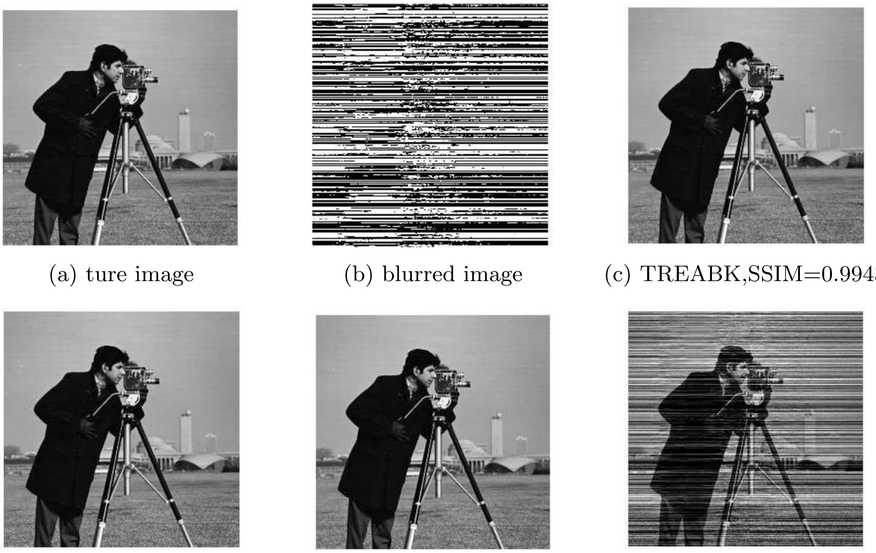
4.4 Sparse

In this part, we use three methods, TREABK, TREBK and TRBK, to solve the sparse tensor equation, i.e. the tensor $\mathcal{A}$ in (2) is sparse. We use the sparse matrix in the Sparse Matrix Set[1] and the reshape function in to construct the sparse tensor. The exact solution $\bar{\mathcal{X}}$ is randomly generated by MATLAB, let $\mathcal{B} = \mathcal{A} * \bar{\mathcal{X}} + \varepsilon$. Table 1 lists the sparsity of $\mathcal{A}$, the noise level of the equation, the number of iterations to solve the equation, the time CPU required, and the error of the solution. From Table 1 it can be observed that as the sparsity of $\mathcal{A}$ increases, TREABK and TREBK achieve higher accuracy compared to TRBK, although TRBK requires less time and computes faster.

[1] http://www.cise.ufl.edu/research/sparse/matrices

Table 1 TREABK, TREBK and TRBK methods for spare tensor equation

Name (Sparsity)	IT	Method	Noise level	CPU	err	Noise level	CPU	err	Noise level	CPU	err
		TREABK		0.5307	0.0021		0.0096	0.3442		0.4650	0.0106
ash85(7.23%)	100	TREBK	1	1.1547	0.3621	0.1	1.1094	0.1886	0.01	0.9789	0.1754
		TRBK		0.4250	0.1965		0.3947	0.3461		0.3923	0.3551
		TREABK		6.2079	1.0085e-4		6.0671	8.5963e-4		6.1129	1.4927e-4
ash292(2.59%)	300	TREBK	1	22.0683	8.9463e-2	0.1	21.2133	7.1882e-2	0.01	22.1090	7.2100e-2
		TRBK		4.0821	1.0392e-3		4.1164	9.3676e-4		4.0665	9.9985e-4
		TREABK		13.6451	1.2534e-34		15.1010	9.4510e-35		18.6017	5.4670e-35
ash331(1.92%)	500	TREBK	1	62.6890	5.3477e-34	0.1	59.9896	3.8697e-34	0.01	62.2058	3.5702e-34
		TRBK		8.3442	5.0995e-34		7.0231	4.9285e-34		8.3145	4.1903e-34
		TREABK		29.5238	4.1132e-16		29.6699	2.9285e-14		45.6399	2.9554e-14
ash608(1.06%)	700	TREBK	1	86.5506	4.8593e-15	0.1	98.9277	5.4886e-14	0.01	123.5880	6.2947e-9
		TRBK		7.4514	2.0922e-7		9.6294	2.3624e-9		10.7635	1.0717e-13
		TREABK		133.1909	8.3523e-29		123.5416	1.1936e-28		127.5105	3.2619e-29
ash958(0.68%)	1100	TREBK	1	691.5615	8.9852e-29	0.1	533.5952	3.7856e-28	0.01	563.2502	9.4674e-29
		TRBK		51.9423	5.8497e-09		40.5250	4.8489e-11		40.8493	6.1485e-13



(a) ture image (b) blurred image (c) TREABK,SSIM=0.9943

(d) TREBK,SSIM=1 (e) TRBK,SSIM=0.9995 (f) TRK,SSIM=0.1937

Fig. 5 (a) ture image, (b) blurred image, (c) TREABK, (d) TREBK, (e) TRBK, (f) TRK

4.5 Gray image

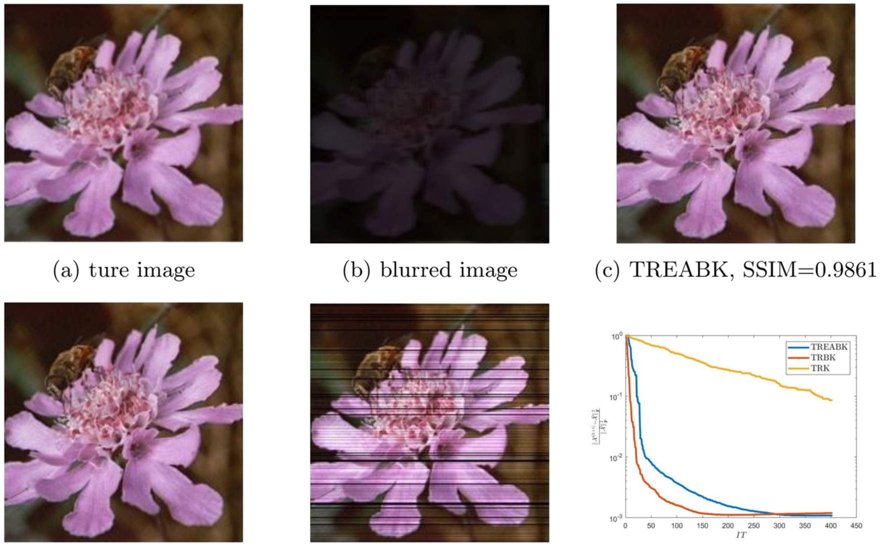
In this part, we utilize the TREABK, TREBK, TRBK, and TRK methods to address image deblurring. Let $\bar{\mathcal{X}} \in \mathbb{R}^{256\times 256\times 3}$ denote the original image, as shown in Fig. 5a. We denote by $\mathcal{A} \in \mathbb{R}^{300\times 256\times 3}$ the Gaussian blur applied to $\bar{\mathcal{X}}$, resulting in the blurred image $\mathcal{B} \in \mathbb{R}^{300\times 256\times 3}$, defined as ($\mathcal{B} = \mathcal{A} * \bar{\mathcal{X}} + \varepsilon$), where ε represents added noise, as illustrated in Fig. 5b. By solving the equation system, we obtain the deblurred image $\mathcal{X}$ and assess the accuracy of the solution by comparing it with the original image's similarity. During computation, we set a maximum of 1000 iterations and a noise level of 0.1, using the "SSIM" function in MATLAB to measure image similarity. The final experimental results are presented in Fig. 5.

Remark 4 In our experiments, the initial iteration tensor of all methods takes the same tensor $\mathcal{X}^{(0)} \in \mathrm{range}(\mathcal{A}^\top)$, which is slightly different from the selection method of TRBK method in the reference [38]. This way of selecting the initial iteration value is conducive to improving the accuracy of the algorithm.

4.6 Color image

This example presents a solution to the colour image[2] deblurring problem using the TREBK, TREABK, and TRK methods. A color image "flower" with the size of $480 \times 512 \times 3$, we adjusted its size to $200 \times 3 \times 200$ and recorded it as $\bar{\mathcal{X}}$, which was shown in

[2] http://www.hlevkin.com/TestImages



(a) ture image (b) blurred image (c) TREABK, SSIM=0.9861

(d) TREBK, SSIM=0.9836 (e) TRBK, SSIM=0.6727 (f)

Fig. 6 (a) ture image, (b) blurred image, (c) TREABK, (d) TREBK, (e) TRBK, (f) error convergence diagram

Fig. 6a and named as the "ture image". Similar to [14], the blurr tensor $\mathcal{A}$ is generated by the MATLAB command

$$z = [\exp(-([0 : \text{band} - 1].^2)/(2\sigma^2)), \text{zeros}(1, N - \text{band})],$$
$$A = \frac{1}{\sigma\sqrt{2\pi}}\text{toeplitz}(z), \mathcal{A}_{:,:,i} = A(i, 1)A, i = 1, \ldots, 200, \sigma = 1, \text{band} = 12.$$

Take the noise level as 1, the blurred image is represented by tensor $\mathcal{B} = \mathcal{A} * \bar{\mathcal{X}} + \varepsilon \in \mathbb{R}^{200\times3\times200}$, as shown in Fig. 6b. By solving the equation system, we obtain the deblurred image $\mathcal{X}$ and assess the accuracy of the solution by comparing it with the original image's similarity. During computation, we set $IT = 400$, using the "SSIM" function in MATLAB to measure image similarity. The final experimental results are presented in Fig. 6.

5 Conclusion

In this paper, we propose the TREABK algorithm to improve the speed of solving large-scale tensor equations in the sense of t-product, while achieving high accuracy and stable convergence in the whole iterative process. We have focused on solving equations involving third-order tensors. As the order of a tensor increases, it becomes capable of representing richer data and information. For instance, a fourth-order tensor can represent a segment of color video or face inpainting, tasks beyond the capability of third-order tensors. However, with the increase in tensor order, the properties of tensor operations also become more complex. Therefore, in our upcoming work, we aim to apply the Kaczmarz method to higher-order tensors to explore its potential performance in these contexts.

Acknowledgements This research was supported by Guizhou Provincial Basic Research Program (Natural Science) under Grant QKHJC-ZK[2023]YB245 and the National Natural Science Foundation of China under Grant 12461072, 12061025.

Author Contributions Liu. made substantial contributions to the conception of the work. An. and Liu. wrote the main manuscript text. Liang. completed the numerical experiment. Jiao. made revisions to the manuscript. All authors reviewed the manuscript.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algebra Appl. **435**(3), 641–658 (2011)
2. Lu, C.Y., Feng, J.S., Chen, Y.D., Liu, W., Lin, Z.C., Yan, S.C.: Tensor robust principal component analysis with a new tensor nuclear norm. IEEE T. Pattern Anal. **42**(4), 925–938 (2019)
3. Miao, Y., Qi, L.Q., Wei, Y.M.: Generalized tensor function via the tensor singular value decomposition based on the t-product. Linear Algebra Appl. **590**, 258–303 (2020)
4. Chen, J.F., Wei, Y.M., Xu, Y.W.: Tensor CUR decomposition under t-product and its perturbation. Numer. Func. Anal. Opt. **43**(6), 698–722 (2022)
5. Lu, C.Y., Peng, X., Wei, Y.C.: Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5996–6004 (2019)
6. Zhou, P., Lu, C.Y., Lin, Z.C., Zhang, C.: Tensor factorization for low-rank tensor completion. IEEE T. Image Process. **27**(3), 1152–1163 (2017)
7. Yu, Q., Zhang, X.Z.: T-product factorization based method for matrix and tensor completion problems. Comput. Optim. Appl. **84**(3), 761–788 (2023)
8. Wang, Z.L., Dong, J.Y., Liu, X.G., Zeng, X.Y.: Low-rank tensor completion by approximating the tensor average rank. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4612–4620 (2021)
9. Song, G.J., Ng, M.K., Zhang, X.J.: Robust tensor completion using transformed tensor singular value decomposition. Numer. Linear Algebra Appl. **27**(3), 2299 (2020)
10. Tarzanagh, D.A., Michailidis, G.: Fast randomized algorithms for t-product based tensor operations and decompositions with applications to imaging data. SIAM J. Imaging Sci. **11**(4), 2629–2664 (2018)
11. Guide, M.EI., Ichi, A.EI., Jbilou, K., Sadaka, R.: On tensor Gmres and Golub-Kahan methods via the t-product for color image processing. Electron. J. Linear. Al. **37**, 524–543 (2021)
12. Pena-Pena, K., Lau, D.L., Arce, G.R.: T-HGSP: Hypergraph signal processing using t-product tensor decompositions. IEEE Trans. Signal Inf. Process Netw. (2023)
13. Long, Z., Liu, Y.P., Chen, L.X., Zhu, C.: Low rank tensor completion for multiway visual data. Signal Process. **155**, 301–316 (2019)

14. Reichel, L., Ugwu, U.O.: Weighted tensor Golu-Kahan-Tikhonov-Type methods applied to image processing using a t-product. J. Comput. Appl. Math. **415**, 114488 (2022)
15. Karczmarz, S.: Angenaherte auflosung von systemen linearer glei-chungen. Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat. 355–357 (1937)
16. Strohmer, T., Vershynin, R.: A randomized Kaczmarz algorithm with exponential convergence. J. Fourier Anal. Appl. **15**(2), 262–278 (2009)
17. Needell, D., Tropp, J.A.: Paved with good intentions: Analysis of a randomized block Kaczmarz method. Linear Algebra Appl. **441**(1), 199–221 (2014)
18. Necoara, I.: Faster randomized block Kaczmarz algorithms. SIAM J. Matrix Anal. Appl. **40**(4), 1425–1452 (2019)
19. Zhang, J.J.: A new greedy Kaczmarz algorithm for the solution of very large linear systems. Appl. Math. Lett. **91**, 207–212 (2019)
20. Bai, Z.-Z., Wu, W.-T.: On greedy randomized Kaczmarz method for solving large sparse linear systems. SIAM J. Sci. Comput. **40**(1), 592–606 (2018)
21. Nutini, J., Sepehry, B., Laradji, I., Schmidt, M., Koepke, H., Virani, A.: Convergence rates for greedy Kaczmarz algorithms, and faster randomized Kaczmarz rules using the orthogonality graph. In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, pp. 547–556 (2016)
22. Niu, Y.Q., Zheng, B.: A greedy block Kaczmarz algorithm for solving large-scale linear systems. Appl. Math. Lett. **104**, 106294 (2020)
23. Miao, C.Q., Wu, W.T.: On greedy randomized average block Kaczmarz method for solving large linear systems. J. Comput. Appl. Math. **413**, 114372 (2022)
24. Needell, D.: Randomized Kaczmarz solver for noisy linear systems. BIT Numer. Math. **50**(2), 395–403 (2010)
25. Zouzias, A., Freris, N.M.: Randomized extended Kaczmarz for solving least-squares. SIAM J. Matrix Anal. Appl. **34**(2), 773–793 (2012)
26. Du, K., Si, W.T., Sun, X.H.: Randomized extended average block Kaczmarz for solving least squares. SIAM J. Sci. Comput. **42**(6), 3541–3559 (2020)
27. Liu, Y.: Accelerated greedy randomized augmented Kaczmarz algorithm for inconsistent linear systems. Appl. Numer. Math. **195**, 142–156 (2024)
28. Bai, Z.-Z., Wang, L.: On multi-step randomized extended Kaczmarz method for solving large sparse inconsistent linear systems. Appl. Numer. Math. **192**, 197–213 (2023)
29. Zeng, Y., Han, D., Su, Y., Xie, J.: Randomized Kaczmarz method with adaptive stepsizes for inconsistent linear systems. Numer. Algorithms **94**(3), 1403–1420 (2023)
30. Schöpfer, F., Lorenz, D.A., Tondji, L., Winkler, M.: Extended randomized Kaczmarz method for sparse least squares and impulsive noise problems. Linear Algebra Appl. **652**, 132–154 (2022)
31. Ma, A., Molitor, D.: Randomized Kaczmarz for tensor linear systems. BIT Numer. Math. **62**(1), 171–194 (2022)
32. Chen, X.M., Qin, J.: Regularized Kaczmarz algorithms for tensor recovery. SIAM J. Imaging Sci. **14**(4), 1439–1471 (2021)
33. Wang, X.Z., Che, M.L., Wei, Y.M.: Randomized Kaczmarz methods for tensor complementarity problems. Comput. Optim. Appl. **82**(3), 595–615 (2022)
34. Wang, X.Z., Che, M.L., Mo, C.X., Wei, Y.M.: Solving the system of nonsingular tensor equations via randomized Kaczmarz-like method. J. Comput. Appl. Math. **421** (2023)
35. Liao, Y.M., Li, W., Yang, D.: The accelerated tensor Kaczmarz algorithm with adaptive parameters for solving tensor systems. Appl. Numer. Math. **202**, 100–119 (2024)
36. Zhang, X.Q., Guo, X.F., Pan, J.Y.: A sampling greedy average regularized Kaczmarz method for tensor recovery. Numer. Linear Algebra, 2560 (2024)
37. Bao, W.D., Zhang, F.Y., Li, W.G., Wang, Q., Gao, Y.: Randomized average Kaczmarz algorithm for tensor linear systems. MATHEMATICS-BASEL **10**(23), 4594 (2022)
38. Huang, G.X., Zhong, S.Y.: Tensor randomized extended Kaczmarz methods for large inconsistent tensor linear equations with t-product. Numer. Algorithms, 1–24 (2023)
39. Du, K., Sun, X.H.: Randomized regularized extended Kaczmarz algorithms for tensor recovery. arXiv preprint arXiv:2112.08566 (2021)
40. Lu, C.Y.: Tensor-tensor product toolbox. arXiv preprint arXiv:1806.07247 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.