Greedy double block extended Kaczmarz method for solving inconsistent tensor linear systems under t-product.

Jeremie Mabiala[*]

Abstract

The randomized Kaczmarz method is a widely adopted iterative method to solving linear systems of equations. To solve large-scale inconsistent linear systems of tensor equations under the t-product, we propose a tensor greedy double block extended Kaczmarz (TGDBEK) method. We prove theoretically the convergence guarantees and showed that the proposed method converges linearly to the minimum-norm least-square solution of the tensor system. Moreover, we assess the performance of the proposed method through several numerical experiments. Compared to existing methods, the TGDBEK does not require predefined partitions of the tensor system and reduces the running time for solving large inconsistent system and requires less iteration.

Keywords. Tensor equations, randomized extended Kaczmarz, tensor greedy randomized Kaczmarz, t-product.

1 Introduction

In this work, we focus on solving the tensor inconsistent linear system of the form

$$\mathcal{A} * \mathcal{X} = \mathcal{B},\ \mathcal{B} = \overline{\mathcal{B}} + \epsilon \tag{1}$$

where $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{N_1 \times K \times N_3}$, $\mathcal{X} \in \mathbb{R}^{N_2 \times K \times N_3}$, are third-order tensors, ϵ the contamination, and $*$ the t-product proposed by Kilmer and collaborators [1, 2]. The t-product provides a matrix-like algebra for third-order tensors, including analogues of inverse, transpose, orthogonal projections, and pseudo-inverse, etc. [1, 2]. One can see [3, 5, 4] for applications of the t-product in various topics like Singular Value Decomposition, Image recognition, neural network [?]. The problem eq.(1) arises in many applications such color image recovery, MRI images deblurring with multiple channels or multiple slices, where the tensor $\mathcal{X}$ is the unknown clean tensor image to be found, $\mathcal{A}$, is a blur operator tensor, and $\mathcal{B}$ is the observed blurred and noisy data. When the noise is relatively higher, it becomes difficult to solve this problem through direct methods. Moreover, this setting makes typically an inconsistent system where the problem is a least-square problem and a natural targeted solution is the minimum-norm solution $\mathcal{A}^\dagger * \mathcal{X}$.

In the matrix framework where $\mathcal{A}$ is a matrix and $\mathcal{X}$ and $\mathcal{B}$ are vectors, iterative methods like the randomized Kaczmarz (RK) method can be used to solve the linear system eq. (1). The RK method selects rows at each iteration with probability proportional to their Euclidean norm square. It was proved that this row selection strategy makes the RK to converge linearly in expectation for consistent and over determined linear systems [28]. To solve inconsistent linear systems, Zouzias and Freris [29] enhanced the randomized Kaczmarz by introducing an additional iterative dual variable and proposed the randomized extended Kaczmarz (REK) which converges linearly in expectation to the minimum-norm least square solution $A^\dagger x$, where $A^\dagger$ is the Moore-Penrose pseudoinverse of the system matrix A. It is worth mentioning that the RK

[*]African Institute of Mathematical Sciences and African Master's of Machine Intelligence,Senegal, (jeremy@aimsammi.org).

method can be slow or can reduce to the cyclic Kaczmarz method when the matrix coefficients are scaled in an efficiently way. Two practical ideas emerged and were studied in order to speed up the randomized Kaczmarz method. The first modifies the row sampling by introducing block sampling and results to block variants randomized Kaczmarz (and other related averaging variants) [24, 10, ?]. In other words, in order to sample a single row, block-based randomized Kaczmarz methods and their variants sample a bock of rows with probability proportional to their Frobenius norm square. The second observes that the rows or blocks can be selected in each iteration according to some greedy criterion where only rows or blocks with largest residuals are selected. This results to greedy randomized Kaczmarz variants studied in [6, 7, 8] and references therein . The greedy selection shrinks the rows or blocks candidates list to project on and typically can results in a fewer iterations and convergence decay.

These ideas have been recently brought to the t-product tensor setting. In their seminal work, Ma and Molitor [12] proposed the tensor randomized Kaczmarz (TRK) method , which extends the RK method in matrix form to solve tensor linear systems of equations under the t-product. It was shown that using the t-product linear algebra that the TRK method is equivalent to the block randomized Kaczmarz method in the Fourier domain and hence it is superior compared to the RK method applied to vectorized tensor system. Subsequently, Chen and Qin[13] 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 [13, 14, 15, 16, 17]. The authors of [18] 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. Huang et al. [21] 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. The TREK method requires computing a pseudoinverse at each iteration. Following that observation, recently, Liyuan An and collaborators [20] proposed a pseudoinverse-free randomized extended block Kaczmarz that leverage the averaging idea we have seen in [24, 10]. Numerical experiments have shown that the TREABK method is relatively similar to the TREBK.

Many of these tensors block methods we have seen above [20, 18] rely inherently on predefined partitions of rows or columns of the tensor system, which can be problem-dependent and many interact unfavorably with heterogeneous sparsity patterns commons in practical problems. In this paper, inspired by the [9], we apply the concept of the greedy randomized block extended Kaczmarz method from matrices to tensor equations. Consequently, we propose a tensor greedy double block extended Kaczmarz (TGDBEK) method, which dynamically builds active set for row and column slices at each iteration, avoiding fixed partitions and focusing projection computation where the the current residual is the largest. In addition, we establish the convergence guarantes of the TGDBEK. Finally, we also demonstrate practical effectiveness of the proposed method against the TREBK and TREGBK [?] and TREABK [9]. We also highlight its application on image deblurring problem and on sparse tensor system constructed from SuiteSparse matrices [?].

The remainder of the paper is organized as follows. Section 2 introduces some definitions and notations that will be used in the sequel. In Section 3, we describe the TGDBEK algorithm and give the convergence guarantee. Numerical examples and application are provided to discuss the effectiveness of the proposed method in Section 4. Finally, some conclusions are drawn in section 5.

2 Notations and preliminaries

In this section , we discuss about the notations and the properties of tensor using the t-product that are used in the sequel.

2.1 Basic notation

Throughout the paper, the calligraphic letters such $\mathcal{A}, \mathcal{B}$ are used to denote tensors. For an integer $m \geq 1$, we denote $[m] := \{1, \ldots, m\}$. For a three-order tensor $\mathcal{A}$ $\mathcal{A}_{ijk}$ denotes the $(i, j, k)^t h$ element of $\mathcal{A}$. $\mathcal{A}(i, :, :)$, $\mathcal{A}(:, j, :)$ and $\mathcal{A}(:, :, k)$ are used to denote the horizontal slices, the lateral slices, and the frontal slices of the tensor $\mathcal{A}$, respectively. The pseudo-inverse of $\mathcal{A}$ is denoted as $\mathcal{A}^\dagger$ and the minimum and the maximum non zero singular values of a matrix are denoted as $\sigma_{\min}(\cdot)$ and $\sigma_{\max}(\cdot)$, respectively.

2.2 Tensor preliminaries

In this part, we provide a brief review of main definitions and results related to tensor using the t-product that are used in the sequel. We use the notations from [][].
For a third-order tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, let $\boldsymbol{A}_i = \mathcal{A}(:, :, i)$. We define the block circulant matrix bcirc$(\mathcal{A})$ of $\mathcal{A}$ as ,

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

Moreover, we define the operations fold$(\cdot)$ and its inverse unfold$(\cdot)$ as follows

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

Definition 2.1 (t-product, [**?**])**.** *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 $N_1 \times K \times N_3$ tensor defined and denoted $\mathcal{A} * \mathcal{B}$*

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

Definition 2.2 ([**?**])**.** *Let $\mathcal{A} \in \mathbb{R}^{N_1 \times \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times K \times N_3}$. Then the t-product between $\mathcal{A}$ and $\mathcal{B}$ satisfy the following properties*

1. *The separability in the first dimension, that is*

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

2. *The component form decomposition on the second dimension*

$$\mathcal{A} * \mathcal{B} = \sum_{j=1}^{N_2} \mathcal{A}(:, j, :) * \mathcal{B}(j, :, :). \tag{2}$$

Definition 2.3 (identity tensor)**.** *The identity tensor $\mathcal{I} \in \mathbb{R}^{N \times N \times N_3}$ is the tensor whose first frontal slice is the $N \times N$ identity matrix, and whose other frontal slices are all zeros.*

For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{I} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$, it holds that

$$\mathcal{A}_{i,:,:} = \mathcal{I}_{i,:,:} * \mathcal{A}. \tag{3}$$

For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{I} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$, it holds that

$$\mathcal{A}_{:,j,:} = \mathcal{A} * \mathcal{I}_{:,j,:}. \tag{4}$$

Definition 2.4 (transpose). *The transpose of $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, denoted by $\mathcal{A}^\top$, is the $N_2 \times N_1 \times N_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through N_3.*

For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, we have

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

Definition 2.5 (identity tensor). *The identity tensor $\mathcal{I} \in \mathbb{R}^{N \times N \times N_3}$ is the tensor whose first frontal slice is the $N \times N$ identity matrix, and whose other frontal slices are all zeros.*

Definition 2.6 (inner product). *The inner product between $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^{N_1 \times N_2 \times N_3}$ is defined as*

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

For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{N_2 \times K \times N_3}$, and $\mathcal{C} \in \mathbb{R}^{N_1 \times K \times N_3}$, it holds that

$$\langle \mathcal{A} * \mathcal{B}, \mathcal{C} \rangle = \langle \mathcal{B}, \mathcal{A}^\top * \mathcal{C} \rangle. \tag{6}$$

We will intensively employ the Frobenius norm of tensors to measure the distance between tensors. We provide the spectral norm, Frobenius norms under the t-product.

Definition 2.7. *For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, its spectral norm and the Frobenius norm are defined as follows*

$$\|\mathcal{A}\|_2 := \|\text{bcirc}(\mathcal{A})\|_2 \tag{7}$$

$$\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = \sqrt{\sum_{i,j,k} \mathcal{A}(i,j,k)^2} \tag{8}$$

Lemma 2.8. *For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, it holds that*

$$\|\mathcal{A} * \mathcal{B}\|_F \leq \|\mathcal{A}\|_2 \|\mathcal{B}\|_F \tag{9}$$

Definition 2.9. *For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, its pseudoinverse, denoted by $\mathcal{A}^\dagger$, is the $N_2 \times N_1 \times N_3$ tensor satisfying*

$$\text{bcirc}(\mathcal{A}^\dagger) = \text{bcirc}(\mathcal{A})^\dagger \tag{10}$$

Lemma 2.10. *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 following holds*

$$\mathcal{A}^\top * \mathcal{A} * \mathcal{A}^\dagger = \mathcal{A}^\top * \mathcal{B} \tag{11}$$

Definition 2.11 ([?]). *For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, the range and the null of of $\mathcal{A}$ are defined as*

$$\begin{aligned} \textit{range}(\mathcal{A}) &:= \left\{ \mathcal{A} * \mathcal{Y} \ : \ \mathcal{Y} \in \mathbb{R}^{N_2 \times K \times N_3} \right\} \\ \textit{null}(\mathcal{A}) &:= \left\{ \mathcal{X} \in \mathbb{R}^{N_2 \times K \times N_3} \ : \ \mathcal{A} * \mathcal{X} = 0 \right\} \end{aligned}$$

Lemma 2.12. *For $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{X} \in$ range($\mathcal{A}$), the following holds*

$$\|\mathcal{A}^\top * \mathcal{X}\|_F^2 \geq \sigma_{\min}^2(\text{bcirc}(\mathcal{A})) \|\mathcal{X}\|_F^2 \tag{12}$$

3 The tensor greedy double block extended Kaczmarz algorithm

Note that the range($\mathcal{A}$) and null($\mathcal{A}$) are orthogonal spaces to each other. For the the system (??), $\mathcal{B} \notin$ range($\mathcal{A}$). Therefore, there must exist $\mathcal{Z} \in$null($\mathcal{A}$) such that $\mathcal{B} = (\mathcal{B} - \mathcal{Z}) + \mathcal{Z}$, where $\mathcal{B} - \mathcal{Z} \in$ range($\mathcal{A}$). Solving the system (??) amounts to solving separately $\mathcal{A}^\top * \mathcal{Z} = \mathcal{O}$ and $\mathcal{A} * \mathcal{X} = \mathcal{B} - \mathcal{Z}$.

Algorithm 1: Tensor Greedy Double Block Extended Kaczmarz (TGDBEK)

```
**Input**: $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, $\mathcal{B} \in \mathbb{R}^{N_1 \times K \times N_3}$, iterations $\ell$, parameter $\eta \in (0,1]$.
**Initialize**: $\mathcal{X}^{(0)} = \mathcal{O}$ and $\mathcal{Z}^{(0)} = \mathcal{B}$.
**for** $k = 0, 1, \dots, \ell - 1$ **do**
    $\varepsilon_k^z = \eta \max\limits_{1 \le j \le N_2} \left\{ \frac{\|(\mathcal{A}_{:,j,:})^\top * \mathcal{Z}^{(k)}\|_{\mathrm{F}}^2}{\|\mathcal{A}_{:,j,:}\|_{\mathrm{F}}^2} \right\}$
    $U_k = \left\{ j \in [N_2] : \|(\mathcal{A}_{:,j,:})^\top * \mathcal{Z}^{(k)}\|_{\mathrm{F}}^2 \ge \varepsilon_k^z \|\mathcal{A}_{:,j,:}\|_{\mathrm{F}}^2 \right\}$
    Update $\mathcal{Z}^{(k+1)} = \mathcal{Z}^{(k)} - \mathcal{A}_{:,U_k,:} * (\mathcal{A}_{:,U_k,:})^\dagger * \mathcal{Z}^{(k)}$
    $\varepsilon_k^x = \eta \max\limits_{1 \le i \le N_1} \left\{ \frac{\|\mathcal{B}_{i,:,:} - \mathcal{Z}_{i,:,:}^{(k+1)} - \mathcal{A}_{i,:,:} * \mathcal{X}^{(k)}\|_{\mathrm{F}}^2}{\|\mathcal{A}_{i,:,:}\|_{\mathrm{F}}^2} \right\}$
    $J_k = \{ i \in [N_1] : \|\mathcal{B}_{i,:,:} - \mathcal{Z}_{i,:,:}^{(k+1)} - \mathcal{A}_{i,:,:} * \mathcal{X}^{(k)}\|_{\mathrm{F}}^2 \ge \varepsilon_k^x \|\mathcal{A}_{i,:,:}\|_F^2 \}$
    Update $\mathcal{X}^{(k+1)} = \mathcal{X}^{(k)} + (\mathcal{A}(J_k,:,:))^\dagger * \left( \mathcal{B}_{J_k,:,:} - \mathcal{Z}_{J_k,:,:}^{(k+1)} - \mathcal{A}(J_k,:,:) * \mathcal{X}^{(k)} \right)$
**end**
```

Theorem 3.1. *Assume the system (??) is inconsistent and let $\mathcal{B}^\perp := \mathcal{B} - \mathcal{A}\mathcal{A}^\dagger\mathcal{B}$. The tensor sequence $\mathcal{X}^{(k)}$ generated by TGDBEK algorithm converges to the tensor least-square solution $\mathcal{X}_* = \mathcal{A}^\dagger * \mathcal{B}$. Moreover, for $k \ge 1$, the following holds*

$$\|\mathcal{X}^{(k+1)} - \mathcal{X}_*\|_F^2 \le \alpha^{k+1} \|\mathcal{X}_0 - \mathcal{X}_*\|_F^2 + \frac{\alpha^{k+1} - \beta^{k+1}}{\alpha - \beta} \|\mathcal{B}^\perp\|_F^2 \tag{13}$$

where $\alpha = 1 - \theta \frac{\sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}^\top))}{\sigma_{\max}^2(\mathrm{bcirc}(\mathcal{A}^\top))}$, $\beta = 1 - \frac{\sigma_{\max}^2(\mathcal{A})}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}(\overline{U})\|_F^2}$, $\gamma = \sigma_{\min}^2(\mathcal{A}^{\mathcal{J}})$, $\theta \in (0,1]$ *and* $\eta \in (0,1]$.

Proof. From Lemma 5, we obtain

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

In particular, for any $U_k \subset [N_2]$ we get $\mathcal{A}(:,U_k,:)^\top \mathcal{B}^\perp = 0$. Now, using the fact $\mathcal{A} * \mathcal{A}^\dagger = (\mathcal{A} * \mathcal{A}^\dagger)^\top = (\mathcal{A}^\dagger)^\top * \mathcal{A}^\top = (\mathcal{A}^\top)^\dagger * \mathcal{A}^\top$ and the z-update in Alg 1, we can get

$$\begin{aligned} \mathcal{Z}^{(k+1)} - \mathcal{B}^\perp &= \left(\mathcal{Z}^{(k)} - \mathcal{B}^\perp\right) - \left(\mathcal{A}_{:,U_k,:}^\top\right)^\dagger * \mathcal{A}_{:,U_k,:}^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}^\perp\right) \\ &= \left(\mathcal{I}_{:,U_k,:} - \left(\mathcal{A}_{:,U_k,:}^\top\right)^\dagger \mathcal{A}_{:,U_k,:}^\top\right) * \left(\mathcal{Z}^{(k)} - \mathcal{B}^\perp\right). \end{aligned}$$

Since $\left(\mathcal{A}_{:,U_k,:}^\top\right)^\dagger \mathcal{A}_{:,U_k,:}^\top$ is a $t-$ tensor projector, using the Pythagorean theorem, we can obtain

$$\|\mathcal{Z}^{(k+1)} - \mathcal{B}^\perp\|_F^2 = \|\mathcal{Z}^{(k)} - \mathcal{B}^\perp\|_F^2 - \| \left(\mathcal{A}_{:,U_k,:}^\top\right)^\dagger * \mathcal{A}_{:,U_k,:}^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}^\perp\right) \|_F^2. \tag{14}$$

Using Lemma (), we note that

$$\begin{aligned}\| \left(\mathcal{A}_{:,U_k,:}^\top\right)^\dagger * \mathcal{A}_{:,U_k,:}^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}^\perp\right) \|_F^2 &= \| \left(\mathcal{A}_{:,U_k,:}^\top\right)^\dagger * \mathcal{A}_{:,U_k,:}^\top * \mathcal{Z}^{(k)} \|_F^2 \\ &\geq \sigma_{\min}^2((\mathcal{A}_{:,U_k,:}^\top)^\dagger) \|\mathcal{A}_{:,U_k,:}^\top * \mathcal{Z}^{(k)}\|_F^2 \\ &= \sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}_{:,U_k,:}^\top)^\dagger) \sum_{j\in U_k} \|\mathcal{A}_{:,j,:}^\top * \mathcal{Z}^{(k)}\|_F^2 \\ &\geq \sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}_{:,U_k,:}^\top)^\dagger)\epsilon_k^z \|\mathcal{A}_{:,U_k,:}\|_F^2 \\ &= \frac{\|\mathcal{A}_{:,U_k,:}\|_F^2}{\sigma_{\max}^2(\mathrm{bcirc}(\mathcal{A}_{:,U_k,:}^\top))}\epsilon_k^z. \end{aligned} \tag{15}$$

Now, using the lemma (), we observe that

$$\begin{aligned}\mathcal{A}_{:,U_{k-1},:}^\top \mathcal{Z}^{(k)} &= \mathcal{A}_{:,U_{k-1},:}^\top \left(\mathcal{Z}^{(k-1)} - \mathcal{A}_{:,U_{k-1},:} * \mathcal{A}_{:,U_{k-1},:}^\dagger \mathcal{Z}^{k-1}\right) \\ &= \mathcal{A}_{:,U_{k-1},:}^\top * \mathcal{Z}^{(k-1)} - \mathcal{A}_{:,U_{k-1},:}^\top * \mathcal{A}_{:,U_{k-1},:} \mathcal{A}_{:,U_{k-1},:}^\dagger * \mathcal{Z}^{k-1} = \mathcal{O},\end{aligned}$$

so that for all $k = 0, 1, \ldots,$, we can get

$$\begin{aligned}\|\mathcal{A}^\top \mathcal{Z}^{(k)}\|_F^2 &= \sum_{j\in U_k} \|\mathcal{A}_{:,j,:}^\top * \mathcal{Z}^{(k)}\|_F^2 + \sum_{j\in U_{k-1}^c} \|\mathcal{A}_{:,j,:}^\top * \mathcal{Z}^{(k)}\|_F^2 \\ &= \sum_{j\in U_{k-1}^c} \|\mathcal{A}_{:,j,:}^\top * \mathcal{Z}^{(k)}\|_F^2 = \sum_{j\in U_{k-1}^c} \frac{\|\mathcal{A}_{:,j,:}^\top * \mathcal{Z}^{(k)}\|_F^2}{\|\mathcal{A}_{:,j,:}\|_F^2} \|\mathcal{A}_{:,j,:}\|_F^2 \\ &\leq \max_{j\in[N_2]} \left\{ \frac{\|\mathcal{A}_{:,j,:}^\top * \mathcal{Z}^{(k)}\|_F^2}{\|\mathcal{A}_{:,j,:}\|_F^2} \right\} \left(\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,U_k,:}\|_F^2\right).\end{aligned}$$

Hence

$$\begin{aligned}\epsilon_k^z = \eta \max_{j\in[N_2]} \left\{ \frac{\|\mathcal{A}_{:,j,:}^\top * \mathcal{Z}^{(k)}\|_F^2}{\|\mathcal{A}_{:,j,:}\|_F^2} \right\} &\geq \eta \frac{\|\mathcal{A}^\top \mathcal{Z}^{(k)}\|_F^2}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,U_k,:}\|_F^2} \\ &\geq \eta \frac{\sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}^\top))\|\mathcal{Z}^{(k)} - \mathcal{B}^\perp\|_F^2}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,U_k,:}\|_F^2},\end{aligned} \tag{16}$$

where in the last inequality we used $\|\mathcal{A}^\top * \mathcal{Z}^{(k)}\|_F^2 = \|\mathcal{A}^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}^\perp\right)\|_F^2 \geq \sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}^\top))\|\mathcal{Z}^{(k)} - \mathcal{B}^\perp\|_F^2$.

Substitute (16) in (15) we can obtain

$$\| \left(\mathcal{A}_{:,U_k,:}^\top\right)^\dagger * \mathcal{A}_{:,U_k,:}^\top * \left(\mathcal{Z}^{(k)} - \mathcal{B}^\perp\right) \|_F^2 \geq \frac{\eta\|\mathcal{A}_{:,U_k,:}\|_F^2}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,U_k,:}\|_F^2} \frac{\sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}^\top))\|\mathcal{Z}^{(k)} - \mathcal{B}^\perp\|_F^2}{\sigma_{\max}^2(\mathrm{bcirc}(\mathcal{A}_{:,U_k,:}^\top))}. \tag{17}$$

Let $\|\mathcal{A}_{:,\overline{U},:}\|_F^2 := \max_k\{\|\mathcal{A}_{:,U_k,:}\|_F^2\}$ so that, combining (14) and (17) above, we can finally obtain the upper bound

$$\begin{aligned}\|\mathcal{Z}^{(k+1)} - \mathcal{B}^\perp\|_F^2 &\leq \left(1 - \frac{\eta\|\mathcal{A}_{:,U_k,:}\|_F^2}{\sigma_{\max}^2(\mathrm{bcirc}(\mathcal{A}_{:,U_k,:}^\top))} \frac{\sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}^\top))}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,\overline{U},:}\|_F^2}\right) \|\mathcal{Z}^{(k)} - \mathcal{B}^\perp\|_F^2 \\ &\leq \left(1 - \eta \frac{\sigma_{\min}^2(\mathrm{bcirc}(\mathcal{A}^\top))}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,\overline{U},:}\|_F^2}\right) \|\mathcal{Z}^{(k)} - \mathcal{B}^\perp\|_F^2.\end{aligned} \tag{18}$$

where in the last inequality we used the fact that $\frac{\|\mathcal{A}_{:,U_k,:}\|_F^2}{\sigma^2_{\max}(\text{bcirc}(\mathcal{A}_{:,U_k,:}))} \geq 1$. The fact that we can decompose $\mathcal{B}_{J_k,:,:} = \mathcal{A}_{J_k,:,:} * \mathcal{X}^* + \mathcal{B}^\perp_{J_k,:,:}$, we can rewrite the the $x-$ update as,

$$\mathcal{X}^{(k+1)} = \mathcal{X}^{(k)} - \mathcal{A}^\dagger_{J_k,:,:} * \left(\mathcal{A}_{:,J_k,:} * \left(\mathcal{X}^{(k)} - \mathcal{X}^*\right) + \left(\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\right)\right)$$

Subtracting $\mathcal{X}_*$ both sides in the equality above, we obtain

$$\begin{aligned}\mathcal{X}^{(k+1)} - \mathcal{X}_* &= \mathcal{X}^{(k)} - \mathcal{X}_* - \mathcal{A}^\dagger_{J_k,:,:} * \left(\mathcal{A}_{:,J_k,:} * \left(\mathcal{X}^{(k)} - \mathcal{X}^*\right) + \left(\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\right)\right) \\ &= \left(\mathcal{I} - \mathcal{A}^\dagger_{J_k,:,:} * \mathcal{A}_{:,J_k,:}\right) * \left(\mathcal{X}^{(k)} - \mathcal{X}^*\right) + \mathcal{A}^\dagger_{J_k,:,:} * \left(\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\right) \qquad (19)\end{aligned}$$

Again $\mathcal{A}^\dagger_{:,J_k,:}\mathcal{A}_{:,J_k,:}$ is a $t-$ orthogonal projector (slice-wise orthogonal in the Fourier domain), hence the two terms are orthogonal in the Frobenius norm. Therefore, from the Pythagorean theorem , we can obtain

$$\|\mathcal{X}^{(k+1)} - \mathcal{X}_*\|_F^2 = \|\left(\mathcal{I} - \mathcal{A}^\dagger_{J_k,:,:} * \mathcal{A}_{:,J_k,:}\right) * \left(\mathcal{X}^{(k)} - \mathcal{X}^*\right)\|_F^2 + \|\mathcal{A}^\dagger_{J_k,:,:} * \left(\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\right)\|_F^2 \qquad (20)$$

Let $\mathcal{X}^{(k)} - \mathcal{X}_* = \left(\mathcal{X}^{(k)} - \mathcal{X}_*\right)_{\mathrm{R}((\mathcal{A}_{:,J_k,:})^\top)} + \left(\mathcal{X}^{(k)} - \mathcal{X}_*\right)_{\mathrm{R}((\mathcal{A}_{:,J_k,:})^\top)^\perp}$ and $\|\left(\mathcal{X}^{(k)} - \mathcal{X}_*\right)_{\mathrm{R}((\mathcal{A}_{:,J_k,:})^\top)}\|_F^2 = \theta_k\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2$. Then, using the Pythagorean theorem and the Lemma (), we get :

$$\begin{aligned}\|\left(\mathcal{I} - \mathcal{A}^\dagger_{J_k,:,:} * \mathcal{A}_{:,J_k,:}\right) * \left(\mathcal{X}^{(k)} - \mathcal{X}^*\right)\|_F^2 &= \|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2 - \|\mathcal{A}^\dagger_{J_k,:,:} * \mathcal{A}_{:,J_k,:} * \left(\mathcal{X}^{(k)} - \mathcal{X}^*\right)\|_F^2 \\ &\leq \|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2 \\ &\quad - \sigma^2_{\min}(\mathcal{A}^\dagger_{:,J_k,:})\sigma^2_{\min}(\mathcal{A}_{:,J_k,:})\|\left(\mathcal{X}^{(k)} - \mathcal{X}_*\right)_{\mathrm{R}((\mathcal{A}_{:,J_k,:})^\top)}\|_F^2 \\ &= \left(1 - \theta_k\frac{\sigma^2_{\min}(\text{bcirc}(\mathcal{A}_{:,J_k,:}))}{\sigma^2_{\max}(\mathcal{A}_{:,J_k,:})}\right)\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2 \\ &\leq \left(1 - \theta\frac{\sigma^2_{\min}(\text{bcirc}(\mathcal{A}_{:,J,:}))}{\sigma^2_{\max}(\mathcal{A}_{:,J,:})}\right)\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2, \qquad (21)\end{aligned}$$

where we denote $J = \{J_0, 1, \ldots, J_k\}$ and $\theta := \min_k\{\theta_k\}$, and $\mathcal{A}_{:,J,:}$ the stacked or the union block of tensors $\mathcal{A}_{:,J_k,:}$. For the second term, we first note that $\|\mathcal{A}^\dagger\mathcal{X}\|_F \leq \|\mathcal{A}^\dagger\|_2^2\|\mathcal{X}\|_F^2$. Then, we obtain

$$\begin{aligned}\|\mathcal{A}^\dagger_{J_k,:,:} * \left(\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\right)\|_F^2 &\leq \sigma^2_{\max}((\mathcal{A}_{:,J_k,:})^\dagger)\|\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\|_F^2 \\ &= \frac{1}{\sigma^2_{\min}(\mathcal{A}_{:,J_k,:})}\|\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\|_F^2. \qquad (22)\end{aligned}$$

It results from (20), (21), and (22) that

$$\begin{aligned}&\|\mathcal{X}^{(k+1)} - \mathcal{X}_*\|_F^2 \\ &\leq \left(1 - \theta\frac{\sigma^2_{\min}(\text{bcirc}(\mathcal{A}_{:,J,:}))}{\sigma^2_{\max}(\mathcal{A}_{:,J,:})}\right)\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2 + \frac{1}{\sigma^2_{\min}(\mathcal{A}_{:,J_k,:})}\|\mathcal{Z}^{(k+1)}_{J_k,:,:} - \mathcal{B}^\perp_{J_k,:,:}\|_F^2 \\ &\leq \left(1 - \theta\frac{\sigma^2_{\min}(\text{bcirc}(\mathcal{A}_{:,J,:}))}{\sigma^2_{\max}(\mathcal{A}_{:,J,:})}\right)\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2 + \frac{1}{\sigma^2_{\min}(\mathcal{A}_{:,J_k,:})}\left(1 - \eta\frac{\sigma^2_{\min}(\text{bcirc}(\mathcal{A}^\top))}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,\overline{U},:}\|_F^2}\right)\|\mathcal{Z}^{(k)} - \mathcal{B}^\perp\|_F^2.\end{aligned}$$

To ease the notation, let $\alpha = 1 - \theta\frac{\sigma^2_{\min}(\text{bcirc}(\mathcal{A}_{:,J,:}))}{\sigma^2_{\max}(\mathcal{A}_{:,J,:})}$, $\beta = 1 - \eta\frac{\sigma^2_{\min}(\text{bcirc}(\mathcal{A}^\top))}{\|\mathcal{A}\|_F^2 - \|\mathcal{A}_{:,\overline{U},:}\|_F^2}$, $\gamma = \frac{1}{\sigma^2_{\min}(\mathcal{A}_{:,J_k,:})}$.

Then, induction over k yields

$$\begin{aligned}
\|\mathcal{X}^{(k+1)} - \mathcal{X}_*\|_F^2 &\leq \alpha\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2 + \frac{1}{\gamma}\beta\|\mathcal{Z}^{(k)} - \mathcal{B}^{\perp}\|_F^2 \\
&\leq \alpha\left(\alpha\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2 + \frac{1}{\gamma}\beta\|\mathcal{Z}^{(k)} - \mathcal{B}^{\perp}\|_F^2\right) + \beta^2\frac{1}{\gamma}\|\mathcal{Z}^{(k-1)} - \mathcal{B}^{\perp}\|_F^2 \\
&\vdots \\
&\leq \alpha^{k+1}\|\mathcal{X}^{(0)} - \mathcal{X}_*\|_F^2 + \frac{1}{\gamma}\sum_{l=0}^{k}\alpha^l\beta^{k-l+1}\|\mathcal{Z}^{(k-1)} - \mathcal{B}^{\perp}\|_F^2 \\
&\leq \alpha^{k+1}\|\mathcal{X}^{(0)} - \mathcal{X}_*\|_F^2 + \frac{\alpha^{k+1} - \beta^{k+1}}{\alpha - \beta}\frac{\beta}{\alpha}\|\mathcal{B}_{R(\mathcal{A})}\|_F^2
\end{aligned}$$

□

4 Numerical experiments

In this section, we assess the performance of the TGDBEK method through numerical experiments to demonstrate its effectiveness compared with TREABK [?], TREBK[?], TREBK[?], and TREGBK. We use PyTorch [?] to implement and run our esperiments. All the experiments were conducted in Python and PyTorch on Google Colab and a Macbook m3 Apple chip.
We measure the performance of the aforementioned methods using the number of iteration steps (”IT”) and the running time (denoted as ”CPU”). Here, IT and CPU represents the arithmetic mean of the number of iterations and running time required to each algorithms to reach the prescribed tolerance 10^{-4}, and these values are replicated 10 times.

For the equation (??), the tensor $\mathcal{A}$ is randomly generated using PyTorch, and the right-hand side $\mathcal{B}$ is set to be $\mathcal{B} = \mathcal{A} * \mathcal{X}_* + \epsilon$, where $\mathcal{X}_*$ is the least-square solution of (??) and the noise term. The initial guess is given $\mathcal{X}^{(0)} = \mathcal{O}$ and $\mathcal{Z}^{(0)} = \mathcal{B}$. All the computations stop when the relative error (err_k) at the $k-$th iteration satisfies $err_k < 10^{-5}$ or when IT exceeds 2000 steps. The relative error of the solution is represented by the following

$$err_k = \frac{\|\mathcal{X}^{(k)} - \mathcal{X}_*\|_F^2}{\|\mathcal{X}_*\|_F^2}$$

where $\mathcal{X}_* = \mathcal{A}^{\dagger} * \mathcal{B}$ is generated using PyTorch. For the noise ϵ, it is computing like in [?] as follows

$$\epsilon = a\zeta\frac{\|\beta\|_F}{\|\zeta\|_F} \tag{23}$$

Example 4.1 (**Dense**). *We test the the performance of TGDBEK, REK, TREBK, TREGBEK, and TREABK algorithms dense tensor systems of the form* (??)*. We solve the equation with* $\mathcal{A}\mathbb{R}^{200\times50\times50}$ *and* $\mathcal{B} \in \mathbb{R}^{200\times50\times50}$*, and the noise* ϵ *given as . For the algorithms which require block pre partition, we set* $\tau = 10$ *for both row and column slices, and the number of partition is set as the ceiling of the* $m//\tau$ *and* $n//\tau$*. Finally, the TGDBEK we let* $\eta = .5$*, and for the others methods we take* $\delta = .7$*.*

Example 4.2 (**Sparse**). *In this example, we test the performance TGDBEK, TREBK, TREGBK methods to solve tensor linear systems of equations, where* $\mathcal{A}$ *is a sparse tensor. To construct the sparse tebsor* $\mathcal{A}$*, we In this example, we test the performance of the TGDBEK, TREK, RDBK (q, k), and REBK (q, k) algorithms on sparse matrices of varying sizes and types. The detailed results are presented in Tables 1–3. It is important to note that for the GDBEK algorithm,* η

is set to 0.5, and the values of q and k are empirically chosen as suitable parameters. In Table 3, we set both the row block and column block sizes in the RDBK algorithm to 5. As shown in Tables 1–3, the GDBEK algorithm outperforms the REK, RDBK, and REBK algorithms in both iteration steps and computing time. This demonstrates the strong competitiveness of the GDBEK algorithm in solving large-scale sparse inconsistent linear systems.

Example 4.3 (**Color image recovery**). *In this example we present a color image restoration problem using the the TGDBEK, TREABK, TREBK, TREGBK, and TREK methods. We apply these method to restore a blurred and noisy colored image of "flower". The original image taken from the internet has the size* $480 \times 512 \times 3$*, and was adjusted to the size of* $200 \times 200 \times 3$ *in order to apply the methods. The deblurring tensor* $\mathcal{A}$ *is obtained as follow*

$$t \in [200],\ z \in \mathbb{R}^{200},\ z_{1:band} = \exp\left(-\frac{t^2_{1:band}}{2\sigma^2}\right),\ A = \frac{1}{\sigma\sqrt{2\pi}}T(z), \textit{ and } \mathcal{A}_{:,:,i} = A(i,:)\,A,\ i = 1,2,3$$

where $T(z)$ *denotes the Toeplitz matrix generated by the vector* z*,* $\mathcal{A} \in \mathbb{R}^{N\times N\times p}$ *is formed slice-by-slice,* σ *is the variance of the Gaussian filter, and band is the band of the Toeplitz, both set to* 3 *and* 25*, respectively. The full code can found in the GitHub repository.*

The noise level was chosen as 10^{-1}*, the blurred and noisy image was obtained as the tensor* $\mathcal{B} = \mathcal{A} * \mathcal{X}_* + \epsilon$*, as displayed in the fig 1b. For all the methods the maximum iteration steps is set* 800*. For the block-based methods TREBK and TREGBK, we use* 10 *partitions for each generated sequentially following [?]. The final results are given in Table 1 and the RSE vs IT convergence plot is given in fig. 1g. We also assess the accuracy of the restoration by comparing the original and the recovered image's similarity with the "SSM" function from scikit-image, which is provided below each image in fig. 1. We can see that the TREGBK and TGDBEK*

Method	Time (s)	RSE	IT
TREK	13.277	0.673	800
TREBK	11.906	0.019	800
TREGBK	18.066	0.000	800
TGDBEK	18.752	0.000	503

Table 1: Results for the color "flower" image restoration

achieve almost the same SSM and PSNR, while TGDBEK requires less iterations to achieve the same results.

Example 4.4 (Gray image recovery).

5 Conclusion

This work presents and extends the greedy double block extended Kaczmarz method from matrix to tensor systems for solving large-scale linear systems of tensor equations under the t=product. The convergence guarantee is given and several numerical examples illustrate the effectiveness of the proposed method.

no. 4, pp. 1678–1693, 2014.

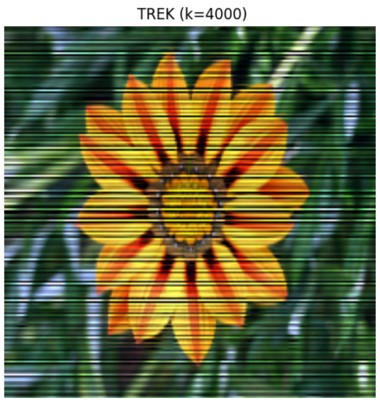
Acknowledgments

References

[1] Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. *Linear Algebra Appl.*, 435(3):641–658, 2011.

(a) True image

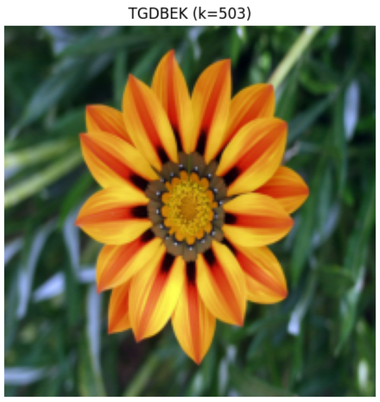
(b) Blurred + noisy



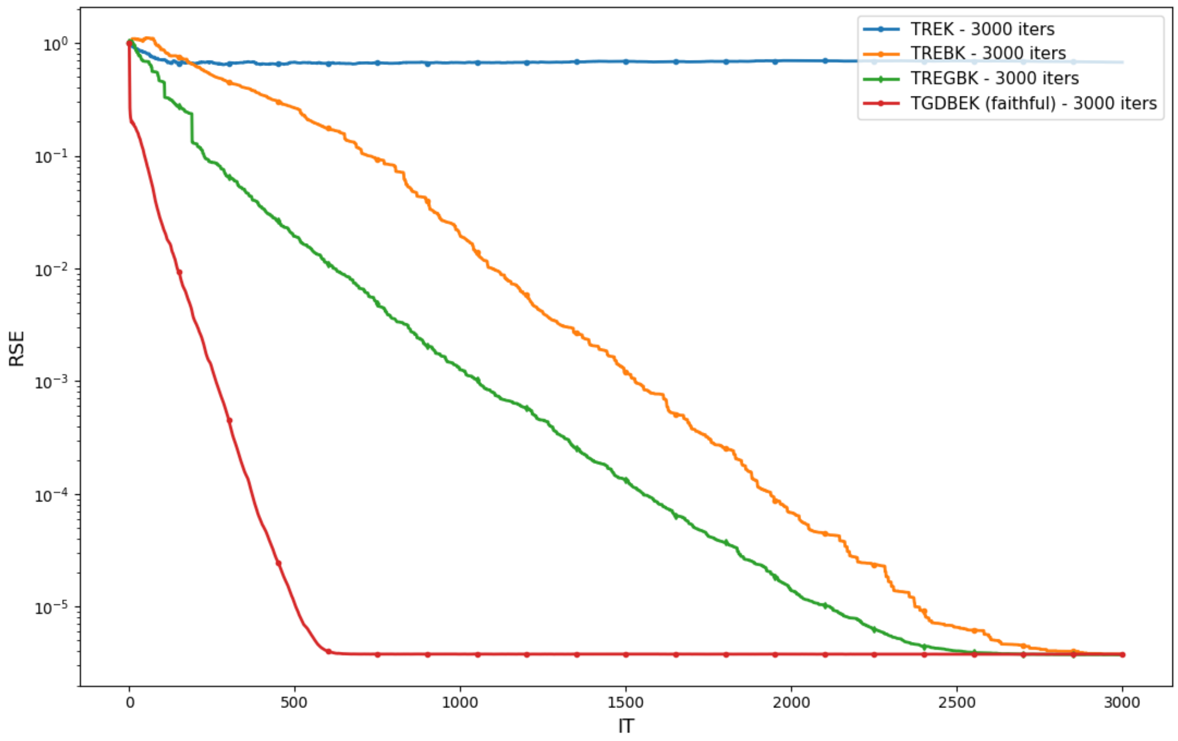
(c) TREK

(d) TREBK

(e) TREGBK



(f) TGDBEK



(g) Plot of the RSE versus IT for each method for the colored ”flower” image recovery by TREK, TREBK, TREGBK and TGDBEK, respectively.

Figure 1

[2] M. Kilmer, K. Braman, N. Hao and R. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging *SIAM Journal on Matrix Analysis and Applications*, 34 (2013), 148-172.

[3] Zhang, A.K. Saibaba, M.E. Kilmer and S. Aeron, "A randomized tensor singular value decomposition based on the t-product". *Numerical Linear Algebra with Applications* 25 (2018) e2179.

[4] Wang, X.Z., Che, M.L., Wei, Y.M.. Tensor neural network models for tensor singular value decompositions. *Comput. Optim. Appl.* 75, 753–777 (2020)

[5] Qi, Liqun and Yu, Gaohang. T-singular values and t-sketching for third order tensors. *arXiv preprint arXiv:2103.00976, 2021.*

[6] Zhong-Zhi Bai and Wen-Ting Wu. On greedy randomized Kaczmarz method for solving large sparse linear systems. *SIAM J. Sci. Comput.*, 40(1):A592–A606, 2018.

[7] Zhong-Zhi Bai and Wen-Ting Wu. On greedy randomized coordinate descent methods for solving large linear least-squares problems. *Numer. Linear Algebra Appl.*, 26(4):e2237, 15, 2019.

[8] Zhong-Zhi Bai and Wen-Ting Wu. On Greedy Randomized Augmented Kaczmarz Method for Solving Large Sparse Inconsistent Linear Systems. *SIAM J. Sci. Comput.*, 43(6):A3892–A3911, 2021.

[9] Wen-Ning Sun and Mei Qin On greedy double block extended Kaczmarz algorithm for solving large inconsistent linear systems. *J. Math. Anal. Appl.* 546 (2025).

[10] Kui Du, Wu-Tao Si, and Xiao-Hui Sun. Randomized extended average block Kaczmarz for solving least squares. *SIAM J. Sci. Comput.*, 42(6):A3541–A3559, 2020.

[11] Stefan Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bull. Intern. Acad. Polonaise Sci. Lett., Cl. Sci. Math. Nat. A*, 35:355–357, 1937.

[12] Ma, A., Molitor, D. Randomized Kaczmarz for tensor linear systems. *BIT* 62(1), 171–194 (2022)

[13] Chen, X.M., Qin, J. Regularized Kaczmarz algorithms for tensor recovery. *SIAM J. Imaging Sci.* 14(4), 1439–1471 (2021)

[14] Wang, X.Z., Che, M.L., Wei, Y.M. Randomized Kaczmarz methods for tensor complementarity problems. *Comput. Optim. Appl.* 82(3), 595–615 (2022)

[15] 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)

[16] 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)

[17] Zhang, X.Q., Guo, X.F., Pan, J.Y. A sampling greedy average regularized Kaczmarz method for tensor recovery. *Numer. Linear Algebra*, 2560 (2024)

[18] 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)

[19] Du, K., Sun, X.H. Randomized regularized extended Kaczmarz algorithms for tensor recovery. *arXiv preprint arXiv:2112.08566* (2021)

[20] Liyuan An, Kun Liang1 Han, and Jiao1 Qilong Liu. Randomized extended average block Kaczmarz method for inconsistent tensor equations under t-product. *Numerical Algorithms* (2025) 100:1123–1144

[21] 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)

[22] Anna Ma, Deanna Needell, and Aaditya Ramdas. Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods. *SIAM J. Matrix Anal. Appl.*, 36(4):1590–1604, 2015.

[23] Yun Miao, Liqun Qi, and Yimin Wei. Generalized tensor function via the tensor singular value decomposition based on the T-product. *Linear Algebra Appl.*, 590:258–303, 2020.

[24] Ion Necoara. Faster randomized block Kaczmarz algorithms. *SIAM J. Matrix Anal. Appl.*, 40(4):1425–1452, 2019.

[25] Deanna Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT*, 50(2):395–403, 2010.

[26] Deanna Needell and Joel A. Tropp. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra Appl.*, 441:199–221, 2014.

[27] Deanna Needell, Ran Zhao, and Anastasios Zouzias. Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra Appl.*, 484:322–343, 2015.

[28] Thomas Strohmer and Roman Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2):262–278, 2009.

[29] Anastasios Zouzias and Nikolaos M. Freris. Randomized extended Kaczmarz for solving least squares. *SIAM J. Matrix Anal. Appl.*, 34(2):773–793, 2013.

[30] Canyi Lu. *Tensor-Tensor Product Toolbox.* Carnegie Mellon University, June 2018. `https://github.com/canyilu/tproduct`.