Numer. Math. Theor. Meth. Appl. Vol. **18**, No. 2, pp. 353-394
doi: 10.4208/nmtma.OA-2024-0096 May 2025

Frontal Slice Approaches for Tensor Linear Systems

Hengrui Luo[1,3] and Anna Ma[2,*]

[1] *Department of Statistics, Rice University, Houston 77005, USA*
[2] *Department of Mathematics, University of California, Irvine 92617, USA*
[3] *Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley 94701, USA*

Received 26 August 2024; Accepted (in revised version) 22 January 2025

Abstract. Inspired by the row and column action methods for solving large-scale linear systems, in this work, we explore the use of frontal slices for solving tensor linear systems. In particular, this paper presents a novel approach for using frontal slices of a tensor $\mathcal{A}$ to solve tensor linear systems $\mathcal{A} * \mathcal{X} = \mathcal{B}$ where $*$ denotes the t-product. In addition, we consider variations of this method, including cyclic, block, and randomized approaches, each designed to optimize performance in different operational contexts. Our primary contribution lies in the development and convergence analysis of these methods. Experimental results on synthetically generated and real-world data, including applications such as image and video deblurring, demonstrate the efficacy of our proposed approaches and validate our theoretical findings.

AMS subject classifications: 15A69, 15A72, 65F10
Key words: Tensor linear system, t-product, iterative method, tensor sketching.

1. Introduction

In the realm of contemporary data science, the availability of multi-dimensional data, commonly referred to as tensors, has catalyzed transformative advances across diverse domains such as machine learning [8, 46], neuroimaging [38], recommendation systems [6] and signal processing [7, 43, 47]. Tensors, which extend beyond the simpler constructs of matrices, encapsulate higher-order interactions within data that matrices alone cannot. While potentially offering a more comprehensive framework for analysis and predictive modeling [7, 27], tensors come with complexity and high dimensionality, which introduce escalated computational costs and demanding storage requirements, particularly in large-scale and high-fidelity compressible datasets [1, 25].

*Corresponding author. *Email address:* anna.ma@uci.edu (A. Ma)

In this work, we are interested in solving large-scale consistent tensor multi-linear systems of the form

$$\mathcal{A} * \mathcal{X} = \mathcal{B}, \tag{1.1}$$

where $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n}$, $\mathcal{X} \in \mathbb{R}^{n_2 \times n_3 \times n}$, $\mathcal{B} \in \mathbb{R}^{n_1 \times n_3 \times n}$, and $*$ denotes the tensor product, known as the t-product [26]. The t-product is a modern tool for working with tensors proposed and developed by Kilmer and Martin [26] a little over a decade ago. This product was motivated by the search for a tensor operation closed under multiplication, which classical tensor products such as the n-mode product violate, and for applications that require tensor factorization. Since then, more and more analyses under the t-product [42, 52] and applications of the t-product have been proposed and studied, including dictionary learning [23, 48], image processing [20, 21], and in neural networks [44]. The deblurring problem specifically can be reformulated into a t-product linear system of the form $\mathcal{A} * \mathcal{X} = \mathcal{B}$ where $\mathcal{X}$ is the underlying images/video, $\mathcal{A}$ is the deblurring operator, and $\mathcal{B}$ is the resulting blurred image/video [12, 46]. Typically, applications in which one needs to solve a t-product linear system can be determined by choice of model for prediction or in settings in where t-product linear systems naturally occur. An example of a naturally occurring setting is the image deblurring problem. In model selection, one assumes that a good model for the response variable $\mathcal{B}$ given $\mathcal{A}$ is given by the t-product operator.

The t-product can be viewed as a generalization of the matrix-vector product. In particular, when $n = 1$ and $n_3 = 1$, the t-product simplifies to the matrix-vector product. In the matrix-data setting, row and column iterative methods have been proposed to solve large-scale linear systems of equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}, \boldsymbol{b} \in \mathbb{R}^{n_1 \times 1}$ are given and $\boldsymbol{x} \in \mathbb{R}^{n_2 \times 1}$ is unknown. When n_1 and n_2 are very large, solving the linear system directly (e.g., by computing the pseudo-inverse) quickly becomes impractical. In other large-scale settings, one may not even be able to load all entries but only a few rows or columns of the matrix $\boldsymbol{A}$ at a time. In such settings, stochastic iterative methods with low memory footprints, such as the Randomized Kaczmarz or Randomized Gauss-Seidel (RGS) algorithms, can be used to solve (1.2). The relationship between these two methods, one using rows of the matrix and the other using columns of the matrix, has been studied in previous works [40]. Such row and column action methods solving linear systems have been further generalized to a framework known as sketch-and-project [8, 17, 18, 43, 45].

Sketching simplifies computations by solving a sub-system as a proxy to the original system while preserving the data's intrinsic characteristics, thereby addressing the practical limitations of direct manipulation due to size or complexity [43,45,51], which proves essential in scenarios where handling full datasets is impractical. A primary advantage of sketching is the enhancement of computational efficiency. In tensor operations such as multiplications or factorizations, multiplication complexity (and so is the solving complexity) can increase exponentially with the sizes of the tensor mode