BAREKENG: Journal of Mathematics and Its Applications
December 2023 Volume 17 Issue 4 Page 2293–2304
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol17iss4pp2293-2304

AN ORDER-P TENSOR MULTIPLICATION WITH CIRCULANT STRUCTURE

Itsar Mangngiri[1], Qonita Qurrota A'yun[2*], Wasono[3]

[1,2,3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Mulawarman University*
Barong Tongkok Street No 4 Kampus Gn. Kelua, Samarinda, Kalimantan Timur, 75123, Indonesia

*Corresponding author's e-mail: * **qonitaqurrota@fmipa.unmul.ac.id***

Article History:

Received: 26th July 2023
Revised: 12th October 2023
Accepted: 11th November 2023

Keywords:

Circulant Matrix;
Discrete Fourier Transform Matrix;
MATLAB;
t-Product;
Tensors

ABSTRACT

Research on mathematical operations involving multidimensional arrays or tensors has increased along with the growing applications involving multidimensional data analysis. The t-product of order-p tensor is one of tensor multiplications. The t-product is defined using two operations that transform the multiplication of two tensors into the multiplication of two block matrices, then the result is a block matrix which is further transformed back into a tensor. The composition of both operations used in the definition of t-product can transform a tensor into a block circulant matrix. This research discusses the t-product of tensors based on their circulant structure. First, we present a theorem of the t-product of tensors involving circulant matrices. Second, we use the definition of identity, transpose, and inverse tensors under t-product operation and investigate their relationship with circulant matrices. Third, we manifest the computation of the t-product involving circulant matrices. The results of the discussion show that the t-product of tensors fundamentally involves circulant matrix multiplication, which means that the operation at its core relies on multiplying circulant matrices. This implies the t-product operation of tensors having properties analogous to standard matrix multiplication. Furthermore, since the t-product of tensors fundamentally involves circulant matrix multiplication, its computation can be simplified by diagonalizing the circulant matrix first using the discrete Fourier transform matrix. Finally, based on the obtained results, an algorithm is constructed in MATLAB to calculate the t-product.

How to cite this article:

I. Mangngiri, Q. Q. A'yun and Wasono., "AN ORDER-P TENSOR MULTIPLICATION WITH CIRCULANT STRUCTURE," *BAREKENG: J. Math. & App.*, vol. 17, iss. 4, pp. 2293-2304, December, 2023.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

In the era of big data, multidimensional data analysis becomes more important, since much of real-world data is inherently multidimensional. Research on mathematical operations involving multidimensional arrays or tensors has increased along with the growing applications involving multidimensional data analysis.

The t-product of order-3 tensor is one of the tensor multiplication operations defined by [1]. The t-product is defined using two operations that transform the multiplication of two tensors into the multiplication of block matrices, then the result is a block matrix which is further transformed back into a tensor. Order-3 t-product definition motivates further researchers to generalize its definition and define concepts on tensor over t-product which analogous to concepts on matrix over standard matrix multiplication. Then, [1] defined identity, inverse, and transpose of order-3 tensor. Hereafter, [2] generalized the definition of t-product, identity, inverse, and transpose of order-3 tensor into order-p tensor (recursively) and constructed an algorithm to compute it using Fast Fourier Transform function in MATLAB. Later, [3] and [4] defined and discuss about Moore-Penrose inverse of tensor over t-product and algorithm to compute it in MATLAB. Afterwards, some applications involving t-product are discussed including image deblurring [2], [5], facial recognition [2], [6], and data compression [7], [8]. Further and related discussion about t-product discussed in articles [5], [6], [9]–[17].

A magnificent theorem formulated by [4] states that the order-3 t-product between two tensors is fundamentally the multiplication of two block circulant matrices obtained by transforming the two tensors. The result of multiplication is then transformed back into a tensor using the inverse transformation of the same transformation. With this theorem, the concepts of tensors related to the t-product can be viewed based on their circulant matrix forms. However, this theorem is still limited to order-3 tensors.

In this research, a more general theorem is proposed compared to the [4]'s theorem, so that it applies to tensors of arbitrary order or order-p tensors. The proof is essentially a recursive version of proving [4]'s theorem, as the order-p t-product is a recursive version of the order-3 t-product. The theorem provides a new and simpler perspective in understanding the definition of the order-p t-product and its related concepts and applications, which have been discussed by previously researchers.

In this research, several concepts of tensors over the t-product will also be discussed based on their block circulant matrix forms, namely identity tensors, inverse, transpose, and Moore-Penrose inverse of tensor. Additionally, algorithms for calculating the t-product involving the diagonalization of circulant matrices using the Discrete Fourier Transform (DFT) matrix will also be explored.

2. RESEARCH METHODS

In this research, several definitions and theorems that have been made by previous studies are used. In this section, these definitions and theorems will be given, which will be presented in several subsections. The first subsection will mention the basic definition and theorem of t-product. The second subsection will present some definitions of concepts on the tensor over the t-product operation. The last subsection will give a theorem that can be used to construct a MATLAB algorithm to calculate the t-product.

2.1 Preliminary Definitions and Theorems

This section provides the definition of t-product and the definition of some operations used on it, then followed by the theorem about the order-3 t-product as the product of two circulant matrices. The theorem will be the basic material in this research.

Definition 1. [18] An $n \times n$ cyclic forward shift matrix, denoted by S_n is a matrix of the form

$$S_n = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Definition 2. [1] Let A be a block matrix of the form

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_\rho \end{bmatrix},$$

with A_i of size $n \times m$, then block circulant matrix of A denoted by $circ(A)$ and defined as follows

$$circ(A) = \left[(S_\rho^\rho \otimes I_n)A \middle| (S_\rho^{\rho-1} \otimes I_n)A \middle| \cdots \middle| (S_\rho^1 \otimes I_n)A\right] = \begin{bmatrix} A_1 & A_\rho & \cdots & A_2 \\ A_2 & A_1 & \cdots & A_3 \\ \vdots & \vdots & \ddots & \vdots \\ A_\rho & A_{\rho-1} & \cdots & A_1 \end{bmatrix}.$$

where $\otimes$ is Kronecker product and I_n is an identity matrix of size n.

Definition 3. [2] The $unfold$ operation works by taking a tensor $\mathcal{A}$ of size $n_1 \times n_2 \times \ldots \times n_p$ and converting it into a block tensor of size $n_p \times 1$ with each block being a tensor of size $n_1 \times n_2 \times \ldots \times n_{p-1}$, as follows

$$unfold(\mathcal{A}) = \begin{bmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \\ \vdots \\ \mathcal{A}_{n_\rho} \end{bmatrix},$$

and $fold$ is the inverse operation of $unfold$, which takes a block tensor of size $n_p \times 1$ with each block being a tensor of size $n_1 \times n_2 \times \ldots \times n_{p-1}$, and then converts it into an $n_1 \times n_2 \times \ldots \times n_p$ tensor. Thus, we get

$$fold\big(unfold(\mathcal{A})\big) = \mathcal{A}.$$

Definition 4. [1] Suppose the tensors $\mathcal{A}$ is $n_1 \times n_2 \times n_3$ and $\mathcal{B}$ is $n_2 \times l \times n_3$. Then the t-product $\mathcal{A} * \mathcal{B}$ is a tensor of size $n_1 \times l \times n_3$, that is

$$\mathcal{A} * \mathcal{B} = fold(circ(unfold(\mathcal{A}))unfold(\mathcal{B})).$$

Definition 5. [3] Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \ldots \times n_p}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3 \times \ldots \times n_p}$. The product of tensors $\mathcal{A} * \mathcal{B}$ is an order-p tensor ($p \geq 3$) of size $n_1 \times l \times n_3 \times \ldots \times n_p$, which is defined recursively as

$$\mathcal{A} * \mathcal{B} = fold(circ\big(unfold(\mathcal{A})\big) * unfold(\mathcal{B})).$$

Finally, a theorem from [4] is given which will be generalized in this research.

Theorem 1. [4] *Suppose $\mathcal{A}$ is a tensor of size $n_1 \times n_2 \times n_3$ and $\mathcal{B}$ is a tensor of size $n_2 \times l \times n_3$, the following holds*

$$\mathcal{A} * \mathcal{B} = \mathcal{C} \Leftrightarrow circ\big(unfold(\mathcal{A})\big)circ\big(unfold(\mathcal{B})\big) = circ(unfold(\mathcal{C})).$$

By **Theorem 1**, the conceptualization of an order-3 t-product can be construed as a multiplication operation involving block circulant. Using this theorem, the computation of the order-3 t-product can be done through the utilization of a simpler structure, namely block circulant matrix. The generalization of **Theorem 1** to encompass order-p tensors represent a significant advancement since it will open up a new and simpler view of understanding the order-p t-product and its concepts and applications that have been discussed in previous researches.

2.2 Some Concepts of Tensor

Some concepts of tensor over t-product operation have been defined and discussed in previous researches [1]–[4], [9], [11]–[15], [17]. Here the definition of some concepts of tensor over t-product that will be discussed more in this research.

Definition 6. [2] The identity tensor of an order-p tensor ($p \geq 3$), is the $n \times n \times n_3 \times \ldots \times n_p$ sized tensor $\mathcal{J}$, defined recursively as a tensor such that $\mathcal{J}_1$ is the identity tensor of an order-$(p-1)$ tensor of size $n \times n \times n_3 \times \ldots \times n_{p-1}$ and $\mathcal{J}_j$, $j = 2, 3, \ldots, n_p$ is an order-$(p-1)$ zero tensor of size $n \times n \times n_3 \times \ldots \times n_{p-1}$.

Definition 7. [2] Suppose $\mathcal{A} \in \mathbb{R}^{n\times n\times n_3\times\ldots\times n_p}$. If there exists order-$p$ tensor $(p \geq 3)$ that is $\mathcal{B} \in R^{n\times n\times n_3\times\ldots\times n_p}$, such that

$$\mathcal{A} * \mathcal{I} = \mathcal{B} = \mathcal{I} * \mathcal{A},$$

then $\mathcal{B}$ is the inverse tensor of $\mathcal{A}$ and is denoted by $\mathcal{A}^{-1}$.

Definition 8. [3] If $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3\times\ldots\times n_p}$, then tensor transpose of $\mathcal{A}$ denoted by $\mathcal{A}^T$, is a tensor of size $n_2 \times n_1 \times n_3 \times\ldots\times n_p$ which is obtained by transposing every $\mathcal{A}_i$, for $i = 1, 2, \ldots, n_p$ and then reversing the order of A_i *for* $i = 2, \ldots, n_p$, namely as follows

$$\mathcal{A}^T = fold\left(\begin{bmatrix} (\mathcal{A}_1)^T \\ \left(\mathcal{A}_{n_p}\right)^T \\ \left(\mathcal{A}_{n_p-1}\right)^T \\ \vdots \\ (\mathcal{A}_2)^T \end{bmatrix}\right).$$

2.3 *t*-Product Algorithm

Some researches on t-product algorithm have been done by [1]–[4], [15]. In this research, we will also construct the t-product algorithm to compute t-product between two tensors and inverse Moore-Penrose of a tensor based on our proposed theorem. Here will be given some definitions and theorems of t-product algorithm by previous researches.

Definition 9. [4] A Discrete Fourier Transform (DFT) Matrix of size $n \times n$ i.e., F_n is defined as

$$F_n = \frac{1}{\sqrt{n}}\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(n-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \omega^{3(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix}.$$

Theorem 2. [2] *Suppose* $circ(A)$ *is a block circulant matrix of* $\rho \times \rho$ *with each block of size* $n \times m$*, then* $(F_\rho \otimes I_n)circ(A)(F_\rho^* \otimes I_m)$ *is a block diagonal matrix.*

Theorem 3. [2] *For arbitrary tensor* $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_p}$ *and* $\mathcal{B} \in \mathbb{R}^{n_2\times l\times\cdots\times n_p}$*, if* F_{n_i} *is DFT matrix of size* $n_i \times n_i$*, because*

$$(F_{n_p} \otimes F_{n_{p-1}} \otimes \cdots \otimes F_{n_3} \otimes I_{n_1})\tilde{A}(F_{n_p}^* \otimes F_{n_{p-1}}^* \otimes \cdots \otimes F_{n_3}^* \otimes I_{n_1})$$

is a block diagonal matrix with $\rho = n_3 n_4 \ldots n_p$ *blocks, with all blocks have size* $n_1 \times n_2$*, then the t-product of* $\mathcal{A} * \mathcal{B}$ *can be computed by folding back to appropriate size tensor of*

$$\left(\tilde{F}^* \otimes I_{n_1}\right)\left(\left(\tilde{F} \otimes I_{n_1}\right)\tilde{A}\left(\tilde{F}^* \otimes I_{n_2}\right)\right)\left(\tilde{F} \otimes I_{n_2}\right)\hat{B}$$

with $\tilde{A}$ *is a block circulant matrix obtained by performing the* $unfold$ *operation then* $circ$ *operation on* $\mathcal{A}$ *repeatedly (recursively),* $\hat{B}$ *is a block matrix obtained by applying the* $unfold$ *operation on* $\mathcal{A}$ *repeatedly (recursively),* $\tilde{F} = F_{n_p} \otimes F_{n_{p-1}} \otimes \cdots \otimes F_{n_3}$*, and the block diagonal matrix can be obtained by repeating FFTs along each mode of* $\mathcal{A}$*.*

3. RESULTS AND DISCUSSION

This research will be conducted in three subsections. The first subsection will create and prove a theorem that extends the theorem on the order-3 ***t***-product as a multiplication of the block circulant matrices form of the tensors to be multiplied by [4] (see **Theorem 1**), so that it applies to any order of tensor. The second subsection will discuss the concepts of tensor over ***t***-product based on its block circulant matrix form. The third subsection will discuss the algorithm for calculating the ***t***-product utilizing the diagonalization of the circulant matrix using the discrete Fourier transform matrix.

3.1 The Main Theorem

In this research, our main theorem will be recursive version of [4]'s with the main objective to calculate the order-p t-product as a standard matrix multiplication of block circulant matrices. As its recursive version, first, we will introduce some recursive operation notations that will help in proving the main theorem.

- $Unfold(\mathcal{A}) = \hat{A}$.
  The $Unfold$ operation on $\mathcal{A}$, denoted $Unfold(\mathcal{A})$ (using uppercase 'U'), works by applying the $unfold$ operation on $\mathcal{A}$ repeatedly (recursively) until the tensor $\mathcal{A}$ turns into a matrix of $n_3 n_4 \dots n_p \times 1$ blocks with each block of size $n_1 \times n_2$, and the result is denoted as $\hat{A}$.
- $Fold(\hat{A}) = \mathcal{A}$.
  The inverse operation of the $Unfold$ operation is $Fold$ (using uppercase 'F'), which is an operation that works by performing the $fold$ operation repeatedly (recursively) on the block matrix the result of the tensor subjected to the $Unfold$ operation such that $Fold(Unfold(\mathcal{A})) = Fold(\hat{A}) = \mathcal{A}$.
- $CU(\mathcal{A}) = \tilde{A}$

  The CU operation on $\mathcal{A}$ or $CU(A)$ works by performing the $unfold$ operation then $circ$ operation on $\mathcal{A}$ repeatedly (recursively) until the tensor $\mathcal{A}$ turns into a block matrix of size $n_3 n_4 \dots n_p \times n_3 n_4 \dots n_p$ with each block of size $n_1 \times n_2$, and the result is denoted by $\tilde{A}$.
- $FU(\tilde{A}) = \mathcal{A}$

  The inverse operation of CU is FU, which is an operation that works by performing the $uncirc$ and then $fold$ operation repeatedly (recursively) on the block matrix the result of the tensor subjected to the CU operation such that $FU(CU(\mathcal{A})) = FU(\tilde{A}) = \mathcal{A}$.

The following lemma is given before proving the main theorem.

Lemma 1. *Suppose tensor* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_p}$ *and tensor* $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3 \times \cdots \times n_p}$ *such that* $\mathcal{A} * \mathcal{B} = \mathcal{C}$*, it follows that*

$$\mathcal{A} * \mathcal{B} = \mathcal{C} \iff \tilde{A}\hat{B} = \tilde{C}.$$

Theorem 4. *Suppose tensor* $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \cdots \times n_p}$ *and tensor* $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3 \times \cdots \times n_p}$ *such that* $\mathcal{A} * \mathcal{B} = \mathcal{C}$*, it follows that*

$$\mathcal{A} * \mathcal{B} = \mathcal{C} \iff \tilde{A}\tilde{B} = \tilde{C}.$$

Proof.

$(\Longleftarrow)$ $\tilde{A}\tilde{B} = \tilde{C} \Rightarrow \mathcal{A} * \mathcal{B} = \mathcal{C}$.

Suppose $\vec{v}$ is a vector, then based on the definition of circulant matrix, we obtain that the first column of $circ(\vec{v})$ is $\vec{v}$. The same is true for tensors, i.e., suppose $\mathcal{A}$ is a tensor, then the first column of $circ(unfold(A))$ is $unfold(A)$, so by a recursive process, it can be obtained that the first column of $\tilde{A}$ is $\hat{A}$. Since $\tilde{A}\tilde{B} = \tilde{C}$ then, based on the definition of matrix multiplication, we have

$$\tilde{A}(first\ column\ of\ \tilde{B}) = (first\ colomn\ of\ \tilde{C})$$
$$\tilde{A}\hat{B} = \tilde{C},$$

or $\tilde{A}\tilde{B} = \tilde{C} \Rightarrow \tilde{A}\hat{B} = \tilde{C}$. Using **Lemma 1** and the rule of syllogism, it is proved that $\tilde{A}\tilde{B} = \tilde{C} \Rightarrow \mathcal{A} * \mathcal{B} = \mathcal{C}$.

$(\Rightarrow)$ $\mathcal{A} * \mathcal{B} = \mathcal{C} \Rightarrow \tilde{A}\tilde{B} = \tilde{C}$.

For simplicity let $\rho = n_3 n_4 \dots n_p$. Based on the definition of $circ$ operation, the second and onwards column of $circ(A)$ is the result of multiplying the cyclic forward shift matrix with its the first column. This also applies to the block matrix $\tilde{A}$, with the second and onwards columns of $\tilde{A}$ being the result of multiplying the cyclic forward shift matrix with its first column which is $\hat{A}$. Therefore, there must be a matrix called M_i, with $i = 1, 2, \dots, \rho$, and $M_i \in \{S_{n_p}^x \otimes S_{n_{p-1}}^y \otimes \cdots \otimes S_{n_3}^z | 1 \le x \le n_p, 1 \le x \le n_{p-1}, \cdots, 1 \le z \le n_3\}$, such that $(M_i \otimes I_{n_1})\hat{A}$, $(M_i \otimes I_{n_2})\hat{B}$, and $(M_i \otimes I_{n_1})\hat{C}$ are respectively the i-th column of $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$, respectively. **Lemma 1** has shown that $\mathcal{A} * \mathcal{B} = \mathcal{C} \Rightarrow \tilde{A}\hat{B} = \hat{C}$, by multiplying both sides by $(M_i \otimes I_{n_1})$ we get $(M_i \otimes I_{n_1})\tilde{A}\hat{B} = (M_i \otimes I_{n_1})\hat{C}$. Then, since the forward sliding cyclic matrix commutes with the circulant

matrix, we get $\tilde{A}(M_i \otimes I_{n_2})\hat{B} = (M_i \otimes I_{n_1})\hat{C}$. Then, if combined into the form of matrix multiplication, using the definition of matrix multiplication is obtained

$$\begin{gathered}\left[\tilde{A}(M_1 \otimes I_{n_2})\hat{B}\middle|\tilde{A}(M_2 \otimes I_{n_2})\hat{B}\middle|\cdots\middle|\tilde{A}(M_p \otimes I_{n_2})\hat{B}\right] = \left[(M_1 \otimes I_{n_1})\hat{C}\middle|(M_2 \otimes I_{n_1})\hat{C}\middle|\cdots\middle|(M_p \otimes I_{n_1})\hat{C}\right] \\ \tilde{A}\left[(M_1 \otimes I_{n_2})\hat{B}\middle|(M_2 \otimes I_{n_2})\hat{B}\middle|\cdots\middle|(M_p \otimes I_{n_2})\hat{B}\right] = \left[(M_1 \otimes I_{n_1})\hat{C}\middle|(M_2 \otimes I_{n_1})\hat{C}\middle|\cdots\middle|(M_p \otimes I_{n_1})\hat{C}\right] \\ \tilde{A}\tilde{B} = \tilde{C},\end{gathered}$$

Therefore, $\tilde{A}\hat{B} = \hat{C} \Rightarrow \tilde{A}\tilde{B} = \tilde{C}$. Then by **Lemma 1**, $\mathcal{A} * \mathcal{B} = \mathcal{C} \Rightarrow \tilde{A}\hat{B} = \tilde{C}$. It follows that $\mathcal{A} * \mathcal{B} = \mathcal{C} \Rightarrow \tilde{A}\tilde{B} = \tilde{C}$. Then it is proved that $\mathcal{A} * \mathcal{B} = \mathcal{C} \Leftrightarrow \tilde{A}\tilde{B} = \tilde{C}$.

Based on **Theorem 4**, the order-p t-product $\mathcal{A} * \mathcal{B}$ can be calculated as

$$\mathcal{A} * \mathcal{B} = FU\big(CU(\mathcal{A})CU(\mathcal{B})\big). \tag{1}$$

By **Equation (1)**, the t-product is fundamentally a matrix multiplication between two block circulant matrices. The t-product multiplication structure based on **Theorem 4** can be more easily understood through the illustration below.

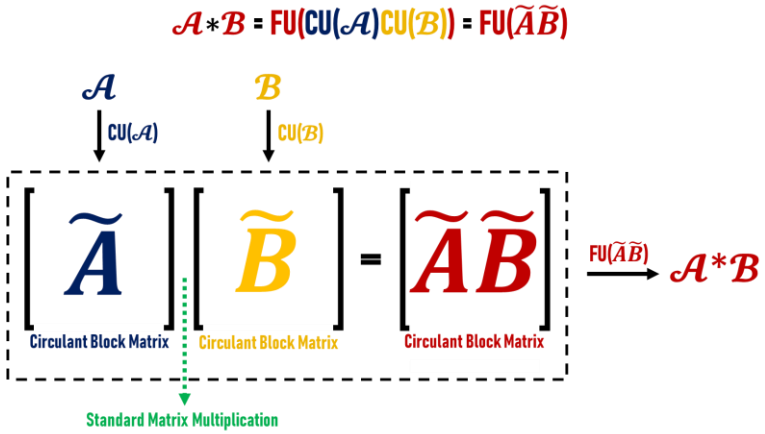

Figure 1. Illustration to Compute t-product Based On Proposed Theorem

This way to compute is simpler than using the definition, because it only uses one reversible transformation, that is the transformation from tensor domain into block circulant matrix domain. **Theorem 1** also states that when we want to multiply two tensors, we must transform the tensor first into its block circulant matrix form.

3.2 Some Concepts of Tensor

As by **Theorem 4** the t-product of two tensor is essentially multiplication of block circulant matrix form of both tensors. In this subsection we will discuss more about some concepts of tensor namely identity, inverse, and transpose of tensor in the view of its block circulant matrix form.

- Identity tensor

  By the **Definition 6**, if $\mathcal{I}$ is an order-p identity tensor of size $n \times n \times n_3 \times \cdots \times n_p$, then the block circulant form of identity tensor is an identity matrix of size $nn_3n_4 \ldots n_p$.

  $$CU(\mathcal{I}) = \tilde{I} = I_{nn_3n_4\ldots n_p}.$$

  By this and by **Theorem 4**, the t-product between any tensor and the identity tensor is essentially the multiplication between any block circulant matrix and the identity matrix. This strengthens the results of previous research which proved that the identity tensor over the t-product has properties analogous to the identity matrix over standard matrix multiplication.

- Inverse of tensor

  By the **Definition 7**, if $\mathcal{A}^{-1}$ is inverse of tensor $\mathcal{A}$ then the block circulant matrix form of $\mathcal{A}^{-1}$ is inverse matrix of block circulant matrix form of $\mathcal{A}$.

$$CU(\mathcal{A}^{-1}) = \widetilde{\mathcal{A}^{-1}} = \tilde{\mathcal{A}}^{-1} = (CU(\mathcal{A}))^{-1}$$

By this and by **Theorem 4**, the t-product between any tensor and its invers is essentially the multiplication between any block circulant matrix and its inverse. This strengthens the results of previous research which proved that the inverse tensor over the t-product has properties analogous to the invers matrix over standard matrix multiplication.

- Transpose of tensor

  By the **Definition 8**, if $\mathcal{A}^T$ is transpose of tensor $\mathcal{A}$ then the block circulant matrix form of $\mathcal{A}^T$ is transpose matrix of block circulant matrix form of $\mathcal{A}$.

$$CU(\mathcal{A}^T) = \widetilde{\mathcal{A}^T} = \tilde{\mathcal{A}}^T = (CU(\mathcal{A}))^T$$

  By this and by **Theorem 4**, the t-product between any tensor and its transpose is essentially the multiplication between any block circulant matrix and its invers. This strengthens the results of previous research which proved that the transpose tensor over the t-product has properties analogous to the transpose matrix over standard matrix multiplication.

- Moore-Penrose inverse of tensor

  By the **Definition 9**, if $\mathcal{A}^+$ is inverse Moore-Penrose of tensor $\mathcal{A}$ then the block circulant matrix form of $\mathcal{A}^+$ is inverse Moore-Penrose of matrix of block circulant matrix form of $\mathcal{A}$.

$$CU(\mathcal{A}^+) = \widetilde{\mathcal{A}^+} = \tilde{\mathcal{A}}^+ = (CU(\mathcal{A}))^+$$

  By this and by **Theorem 4**, the t-product between any tensor and its Moore-Penrose inverse is essentially the multiplication between any block circulant matrix and its Moore-Penrose inverse. This strengthens the results of previous research which proved that the Moore-Penrose inverse tensor over the t-product has properties analogous to the Moore-Penrose inverse matrix over standard matrix multiplication.

3.3 t-Product Algorithm

As **Theorem 4** simplifies the equation for calculating the t-product, **Theorem 4** can also be used to simplify the equation for calculating t-product in **Theorem 3**. The simpler form of the equation for calculating the t-product in **Theorem 3** will make it easier to understand the t-product algorithm. This section will discuss the MATLAB algorithm for calculating the t-product and the inverse Moore-Penrose of a tensor.

By the **Theorem 3** and **Theorem 4,** the t-product $\mathcal{A} * \mathcal{B}$ can be compute as

$$\mathcal{A} * \mathcal{B} = FU\left(\left(\tilde{F}^* \otimes I_{n_1}\right)\left(\left(\tilde{F} \otimes I_{n_1}\right)\tilde{A}\left(\tilde{F}^* \otimes I_{n_2}\right)\right)\left(\left(\tilde{F} \otimes I_{n_2}\right)\tilde{B}\left(\tilde{F}^* \otimes I_l\right)\right)\left(\tilde{F} \otimes I_l\right)\right).$$

The illustration to compute it can be seen in **Figure 2**.

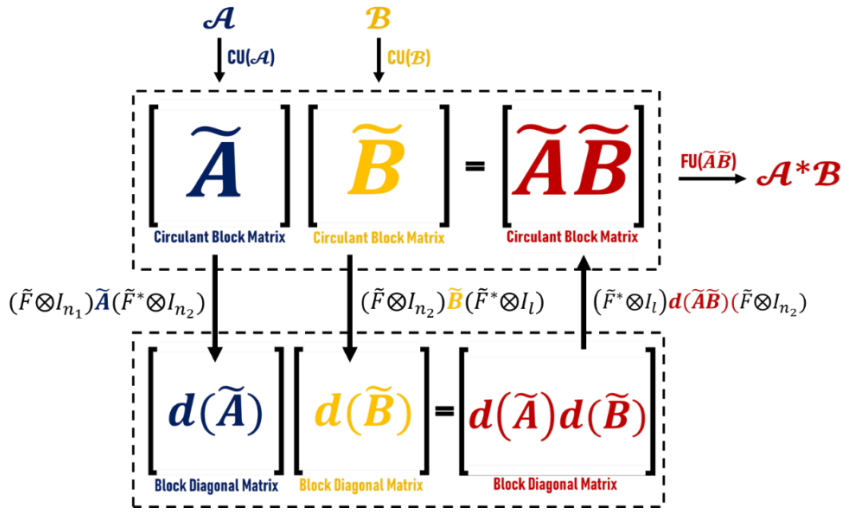

Figure 2. Illustration to Compute t-product using DFT Matrix

As on MATLAB, the block diagonal matrix results of diagonalization of a block circulant matrix that is $(\tilde{F} \otimes I_{n_1})\tilde{A}(\tilde{F}^* \otimes I_{n_2})$ can be obtained by applying repeating FFTs along each mode of tensor $\mathcal{A}$. Using MATLAB one can compute t-product by the following algorithm.

MATLAB algorithm to compute the t-product

Input: $n_1 \times n_2 \times n_3 \times \cdots \times n_p$ tensor $\mathcal{A}$ and $n_2 \times l \times n_3 \times \cdots \times n_p$ tensor $\mathcal{B}$

```
% Defining tensor A is done by writing each piece in the form of the matrix as A(:,:,i3,i4,...,ip ), as
well with B

SA = size(A);   % stores the size of the tensor A as a vector
SB = size(B);   % stores the size of the tensor A as a vector

for i = 3 : ndims(A)  % ndims(A) = order of A = p
   A = fft(A,[],i);      % repeating FFTs along each mode of A, until a block diagonal matrix is obtained
   B = fft(B,[],i);      % repeating FFTs along each mode of B, until a block diagonal matrix is obtained
end

C = repmat(0,[SA(1) SB(2) SA(3:ndims(A))]);   % create an empty tensor of size n1 × l × n3 × ... × np

for i = 1 : prod(SA(3 : ndims(A)))    % calculate the size of the block diagonal matrix
   C(:,:,i) = A(:,:,i)*B(:,:,i);              % multiplying the two matrix blocks, then store it on C
end

for i = ndims(A):-1:3
   C = ifft(C,[],i);          % repeating inverse FFTs along each mode of tensor A, until a tensor obtained
end

C      % displays the result
```

For instance, let $\mathcal{A}$ be a tensor of size $2 \times 3 \times 2 \times 2$ as follows

$$\mathcal{A}_{11} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad \mathcal{A}_{12} = \begin{bmatrix} 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix}, \quad \mathcal{A}_{21} = \begin{bmatrix} 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}, \quad \mathcal{A}_{22} = \begin{bmatrix} 19 & 20 & 21 \\ 22 & 23 & 24 \end{bmatrix},$$

and $\mathcal{B}$ be a tensor of size $3 \times 2 \times 2 \times 2$ below

$$\mathcal{B}_{11} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \quad \mathcal{B}_{12} = \begin{bmatrix} 13 & 14 \\ 15 & 16 \\ 17 & 18 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 19 & 20 \\ 21 & 22 \\ 23 & 24 \end{bmatrix},$$

Then we can compute tensor $\mathcal{C} = \mathcal{A} * \mathcal{B}$ using MATLAB code as follows

Example MATLAB code to compute the t-product

```
A(:,:,1,1) = [1 2 3 ; 4 5 6];
A(:,:,1,2) = [7 8 9 ; 10 11 12];
A(:,:,2,1) = [13 14 15 ; 16 17 18];
A(:,:,2,2) = [19 20 21 ; 22 23 24];

B(:,:,1,1) = [1 2 ; 3 4 ; 5 6];
B(:,:,1,2) = [7 8 ; 9 10 ; 11 12];
B(:,:,2,1) = [13 14 ; 15 16 ; 17 18];
B(:,:,2,2) = [19 20 ; 21 22 ; 23 24];

SA = size(A);   % stores the size of the tensor A as a vector
SB = size(B);   % stores the size of the tensor A as a vector

for i = 3 : ndims(A)  % ndims(A) = order of A = p
   A = fft(A,[],i);      % repeating FFTs along each mode of A, until a block diagonal matrix is obtained
   B = fft(B,[],i);      % repeating FFTs along each mode of B, until a block diagonal matrix is obtained
end
```

C = repmat(0,[SA(1) SB(2) SA(3:ndims(A))]); % create an empty tensor of size $n_1 \times l \times n_3 \times \ldots \times n_p$

for i = 1 : prod(SA(3 : ndims(A))) % calculate the size of the block diagonal matrix
C(:,:,i) = A(:,:,i)*B(:,:,i); % multiplying the two matrix blocks, then store it on C
end

for i = ndims(A):-1:3
C = ifft(C,[],i); % repeating inverse FFTs along each mode of tensor $\mathcal{A}$, until a tensor obtained
end

C % displays the result

We obtain the result as follows

$$\mathcal{C}_{11} = \begin{bmatrix} 2140 & 2272 \\ 2572 & 2740 \end{bmatrix}, \ \mathcal{C}_{12} = \begin{bmatrix} 1276 & 1408 \\ 1708 & 1876 \end{bmatrix}, \ \mathcal{C}_{21} = \begin{bmatrix} 1924 & 2056 \\ 2356 & 2524 \end{bmatrix}, \ \mathcal{C}_{22} = \begin{bmatrix} 1060 & 1192 \\ 1492 & 1660 \end{bmatrix}.$$

By the **Theorem 3** and **Theorem 4** we can compute invers Moore-Penrose of tensor $\mathcal{A}$ as

$$\mathcal{A}^{+} = FU\left(\left(\tilde{F}^{*} \otimes I_{n_1}\right)\left(\left(\tilde{F} \otimes I_{n_1}\right)\tilde{A}\left(\tilde{F}^{*} \otimes I_{n_2}\right)\right)^{+}\left(\tilde{F} \otimes I_{n_2}\right)\right)$$

The illustration to compute it can be seen in **Figure 3.**

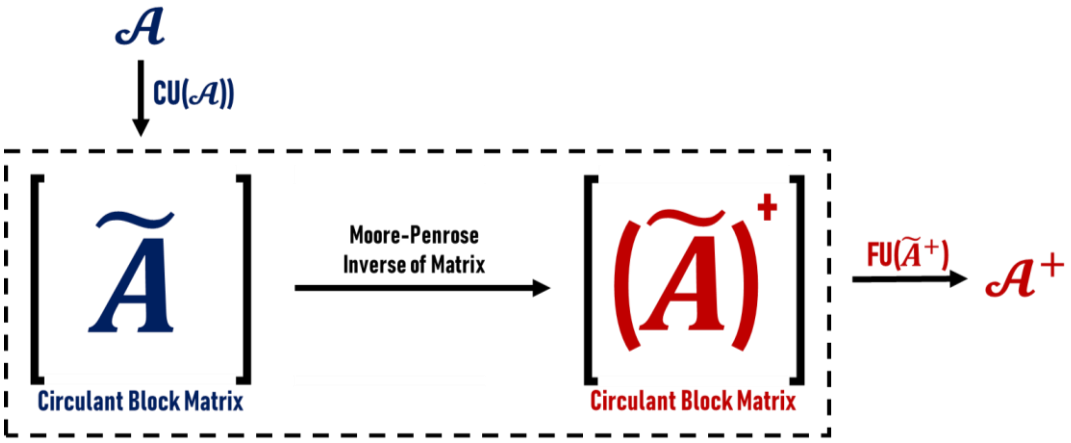

Figure 3. Illustration to Compute Moore-Penrose Inverse Tensor Using DFT matrix

Same as the t-product, using FFT function in MATLAB, one can compute Moore-Penrose inverse of a tensor easier. The algorithm to compute it can be seen in the following algorithm.

MATLAB algorithm to compute inverse Moore-Penrose of a tensor

Input: $n_1 \times n_2 \times n_3 \times \cdots \times n_p$ tensor $\mathcal{A}$
% defining tensor $\mathcal{A}$ is done by writing each piece in the form of the matrix as $A(:,:,i_3,i_4,\ldots,i_p)$

SA = size(A); % stores the size of the tensor $\mathcal{A}$ as a vector

for i = 3:ndims(A) % ndims(A) = order of $\mathcal{A} = p$
A = fft(A,[],i); % repeating FFTs along each mode of $\mathcal{A}$, until a block diagonal matrix is obtained
end

MP = repmat(0,[SA(1) SB(2) SA(3:ndims(A))]); % create an empty tensor of size $n_1 \times l \times n_3 \times \ldots \times n_p$

for i = 1:prod(SA(3:ndims(A))) % prod(SA(3:ndims(A))) to calculate the size of the block diagonal matrix
MP(:,:,i) = pinv(A(:,:,i)); % calculate inverse Moore-Penrose of every block, then store it on MP
end

for i = ndims(A):-1:3

```
    MP = ifft(MP,[],i);        % repeating inverse FFTs along each mode of tensor 𝒜, until a tensor obtained
end

MP      % displays the result
```

For instance, let $\mathcal{A}$ be a tensor of size $2 \times 2 \times 2 \times 2$, where

$$\mathcal{A}_{11} = \begin{bmatrix} 1.75 & 0 \\ 0 & 1.25 \end{bmatrix}, \quad \mathcal{A}_{12} = \begin{bmatrix} -0.25 & 0 \\ 0 & -0.25 \end{bmatrix}, \quad \mathcal{A}_{21} = \begin{bmatrix} 0.75 & 0 \\ 0 & -0.75 \end{bmatrix}, \quad \mathcal{A}_{22} = \begin{bmatrix} -1.25 & 0 \\ 0 & 0.25 \end{bmatrix}.$$

We implement the following MATLAB algorithm to compute inverse Moore-Penrose of tensor $\mathcal{A}$.

Example MATLAB code to compute inverse Moore-Penrose of a tensor

```
A(:,:,1,1)=[1.75 0; 0 1.25];
A(:,:,1,2)=[0.75 0 ; 0 -0.75];
A(:,:,2,1)=[-0.25 0 ; 0 0.25];
A(:,:,2,2)=[-1.25 0; 0 0.25];

SA = size(A);   % stores the size of the tensor 𝒜 as a vector

for i = 3:ndims(A)  % ndims(A) = order of 𝒜 = p
    A = fft(A,[],i);    % repeating FFTs along each mode of 𝒜, until a block diagonal matrix is obtained
end

MP = repmat(0,[SA(1) SB(2) SA(3:ndims(A))]);  % create an empty tensor of size n1 × l × n3 × ... × np

for i = 1:prod(SA(3:ndims(A)))  % prod(SA(3:ndims(A))) to calculate the size of the block diagonal matrix
    MP(:,:,i) = pinv(A(:,:,i));         % calculate inverse Moore-Penrose of every block, then store it on MP
end

for i = ndims(A):-1:3
    MP = ifft(MP,[],i);        % repeating inverse FFTs along each mode of tensor 𝒜, until a tensor obtained
end

MP      % displays the result
```

Then we obtain

$$\mathcal{A}^{+}_{11} = \begin{bmatrix} 0.4375 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad \mathcal{A}^{+}_{12} = \begin{bmatrix} 0.3125 & 0 \\ 0 & 0.25 \end{bmatrix}, \quad \mathcal{A}^{+}_{21} = \begin{bmatrix} 0.1875 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{A}^{+}_{22} = \begin{bmatrix} 0.0625 & 0 \\ 0 & 0.25 \end{bmatrix}.$$

The algorithm we provide in this article is a more detail and improved version of the MATLAB algorithm previously constructed by [2] and [3]. Our focus in the algorithm is to give a stronger understanding that the algorithms can work because the t-product is basically a multiplication between two block circulant matrices as stated in our theorem and can be clearly understood through **Figure 2** and **Figure 3**. Based on **Figure 2** and **Figure 3** and **Theorem 3**, a simpler illustration of the algorithms can be seen in **Figure 4** and **Figure 5**.

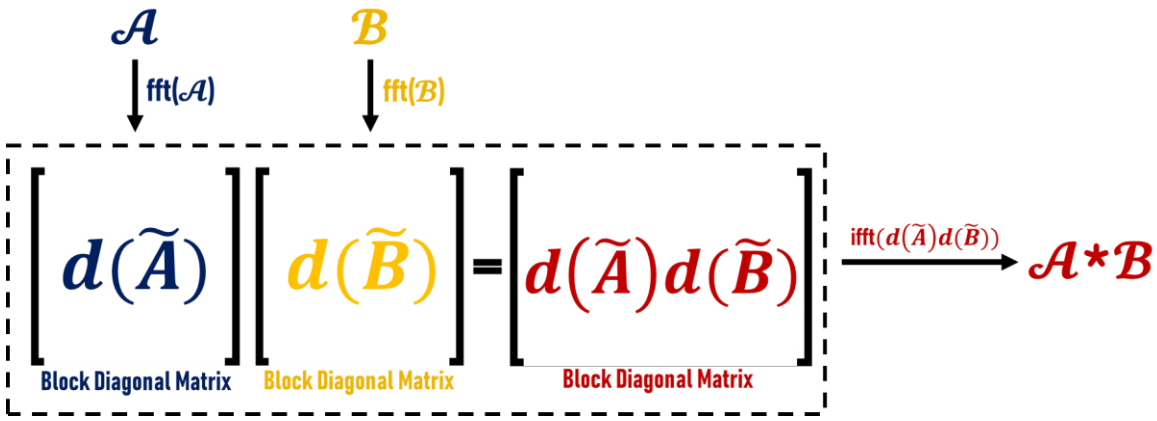

Figure 4. Illustration to compute t-product using FFT MATLAB

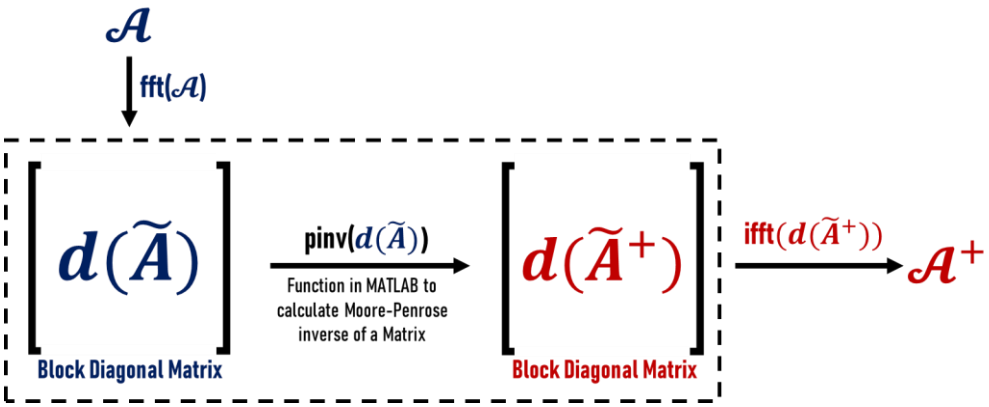

Figure 5. Illustration to compute Moore-Penrose inverse using FFT MATLAB

4. CONCLUSIONS

According the result and discussion above, there are three main points that can be concluded, which are as follows:

1. The t-product of tensors fundamentally involves circulant matrix multiplication, which means that the operation at its core relies on multiplying circulant matrices.
2. The proposed theorem makes the concepts of tensor over t-product more understandable to be analogues to the same concepts in matrix over standard multiplication.
3. The theorem simplifies the computation of t-product and inverse Moore-Penrose of a tensor using matrix DFT.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support for this research with the PNBP Research and Community Service Financing Scheme of the Faculty of Mathematics and Natural Sciences, Mulawarman University based on Letter No. 1410/UN17.7/PP/2023.

REFERENCES

[1] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011, doi: 10.1016/j.laa.2010.09.020.
[2] C. D. Martin, R. Shafer, and B. LaRue, "An Order-p Tensor Factorization with Applications in Imaging," *SIAM J. Sci. Comput.*, vol. 35, no. 1, pp. A474–A490, Jan. 2013, doi: 10.1137/110841229.
[3] H. Jin, M. Bai, J. Benítez, and X. Liu, "The generalized inverses of tensors and an application to linear models," *Comput. Math. with Appl.*, vol. 74, no. 3, pp. 385–397, Aug. 2017, doi: 10.1016/j.camwa.2017.04.017.
[4] H. Jin, P. Zhou, H. Jiang, and X. Liu, "The generalized inverses of the quaternion tensor via the T-product," Nov. 2022, Accessed: Aug. 10, 2023. [Online]. Available: https://arxiv.org/abs/2211.02836v1
[5] M. Liu, X. Zhang, and L. Tang, "Real Color Image Denoising Using t-Product- Based Weighted Tensor Nuclear Norm Minimization," *IEEE Access*, vol. 7, pp. 182017–182026, 2019, doi: 10.1109/ACCESS.2019.2960078.
[6] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover, "Facial Recognition Using Tensor-Tensor Decompositions," *SIAM J. Imaging Sci.*, vol. 6, no. 1, pp. 437–463, Jan. 2013, doi: 10.1137/110842570.
[7] M. Kilmer, L. Horesh, H. Avron, and E. Newman, "Tensor-Tensor Products for Optimal Representation and Compression," pp. 1–27, 2019, [Online]. Available: http://arxiv.org/abs/2001.00046
[8] M.-M. Zheng and G. Ni, "Approximation strategy based on the T-product for third-order quaternion tensors with application to color video compression," *Appl. Math. Lett.*, vol. 140, p. 108587, Jun. 2023, doi: 10.1016/j.aml.2023.108587.
[9] M. Rezghi and L. Eldén, "Diagonalization of tensors with circulant structure," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 422–447, Aug. 2011, doi: 10.1016/j.laa.2010.03.032.
[10] J. Chen, Y. Wei, and Y. Xu, "Tensor CUR Decomposition under T-Product and Its Perturbation," *Numer. Funct. Anal. Optim.*, vol. 43, no. 6, pp. 698–722, Apr. 2022, doi: 10.1080/01630563.2022.2056198.
[11] X.-Y. Liu and X. Wang, "Fourth-order Tensors with Multidimensional Discrete Transforms," May 2017, Accessed: Aug. 10, 2023. [Online]. Available: http://arxiv.org/abs/1705.01576
[12] J. Ji and Y. Wei, "The Drazin inverse of an even-order tensor and its application to singular tensor equations," *Comput. Math. with Appl.*, vol. 75, no. 9, pp. 3402–3413, May 2018, doi: 10.1016/j.camwa.2018.02.006.

[13] J. K. Sahoo, R. Behera, P. S. Stanimirović, V. N. Katsikis, and H. Ma, "Core and core-EP inverses of tensors," *Comput. Appl. Math.*, vol. 39, no. 1, p. 9, Mar. 2020, doi: 10.1007/s40314-019-0983-5.
[14] K. Panigrahy, R. Behera, and D. Mishra, "Reverse-order law for the Moore–Penrose inverses of tensors," *Linear Multilinear Algebr.*, vol. 68, no. 2, pp. 246–264, Feb. 2020, doi: 10.1080/03081087.2018.1502252.
[15] Y. Miao, L. Qi, and Y. Wei, "Generalized tensor function via the tensor singular value decomposition based on the T-product," *Linear Algebra Appl.*, vol. 590, pp. 258–303, Apr. 2020, doi: 10.1016/j.laa.2019.12.035.
[16] Y.-N. Cui and H.-F. Ma, "The perturbation bound for the T-Drazin inverse of tensor and its application," *Filomat*, vol. 35, no. 5, pp. 1565–1587, 2021, doi: 10.2298/FIL2105565C.
[17] Y. Miao, L. Qi, and Y. Wei, "T-Jordan Canonical Form and T-Drazin Inverse Based on the T-Product," *Commun. Appl. Math. Comput.*, vol. 3, no. 2, pp. 201–220, Jun. 2021, doi: 10.1007/s42967-019-00055-4.
[18] B. J. Olson, S. W. Shaw, C. Shi, C. Pierre, and R. G. Parker, "Circulant Matrices and Their Application to Vibration Analysis," *Appl. Mech. Rev.*, vol. 66, no. 4, Jul. 2014, doi: 10.1115/1.4027722.