Research Article

The Tensor Padé-Type Approximant with Application in Computing Tensor Exponential Function

Chuanqing Gu ¹ and Yong Liu ^{1,2}

¹Department of Mathematics, Shanghai University, Shanghai 200444, China

²Changzhou College of Information Technology, Changzhou 213164, China

Correspondence should be addressed to Chuanqing Gu; cqgu@staff.shu.edu.cn

Received 21 April 2018; Accepted 22 May 2018; Published 19 June 2018

Academic Editor: Liguang Wang

Copyright © 2018 Chuanqing Gu and Yong Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Tensor exponential function is an important function that is widely used. In this paper, tensor Padé-type approximant (TPTA) is defined by introducing a generalized linear functional for the first time. The expression of TPTA is provided with the generating function form. Moreover, by means of formal orthogonal polynomials, we propose an efficient algorithm for computing TPTA. As an application, the TPTA for computing the tensor exponential function is presented. Numerical examples are given to demonstrate the efficiency of the proposed algorithm.

1. Introduction

Tensor exponential function is an important function that is widely used, owing to its key role in the solution of tensor differential equations [1–4]. For instance, Markovian master equation can be written as tensor differential equations $(\partial/\partial t)\mathbf{P}(t) = \mathbf{A} \cdot \mathbf{P}(t)$, where the probabilities tensor $\mathbf{P}(t) \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ [5]. Consider the initial value problem defined by the tensor ordinary differential equation [6, 7]

$$\begin{aligned}\dot{\mathcal{Y}}(t) &= \mathcal{A}\mathcal{Y}(t), \\ \mathcal{Y}(t_0) &= \mathcal{Y}_0,\end{aligned}\tag{1}$$

where the superimposed dot denotes differentiation with respect to t and $\mathcal{A}$ and $\mathcal{Y}_0$ are given constant tensors. The solution to system (1) is $\mathcal{Y}(t) = \exp[(t - t_0)\mathcal{A}]\mathcal{Y}_0$, and $\exp(\cdot)$ is the tensor exponential function.

To solve (1), we need to calculate an exponential function about tensor $\mathcal{A}$. Recently, tensor computation, especially eigenvalues of tensors, has attracted attention of many scholars; some important results have been found in the current literature [8–15]. For instance, computation of $\exp(\mathcal{A}t)$ is required in applications such as finite strain hyperelastic-based multiplicative plasticity models [7, 16–19]. Explicitly,

for a generic tensor $\mathcal{A}$, the tensor exponential can be expressed by means of its series representation [7] (see p.749).

$$\exp(\mathcal{A}) = \sum_{n=0}^{\infty} \frac{1}{n!} \mathcal{A}^n, \tag{2}$$

and the preceding series is absolutely convergent for any argument $\mathcal{A}$ and, as its scalar counterpart, can be used to evaluate the tensor exponential function to any prescribed degree of accuracy [16]. The computation of (2) is carried out by simply truncating the infinite series with n_{max} terms:

$$\exp(\mathcal{A}) = \sum_{n=0}^{n_{max}} \frac{1}{n!} \mathcal{A}^n, \tag{3}$$

with n_{max} being such that $(1/n_{max})\|\mathcal{A}^{n_{max}}\| \leq \epsilon_{tol}$.

However, the accuracy and effectiveness of the preceding algorithm is limited by round-off and choice of termination criterion [16]. Padé approximant has become by far the most widely used one in calculation of exponential function or formal power series due to the following reasons: first, the series may converge too slowly to be of any use and the approximation can accelerate its convergence; second, only few coefficients of the series may be known and a good approximation to the series is needed to obtain properties of

the function that it represents [20]. For instance, matrix Padé-type approximant (MPTA) [21] can be used to simplify the high degree multivariable system by approximating transfer function matrix $G(s)$ that can be expanded into a power series with matrix coefficients, i.e., $G(s) = \sum_{i=0}^{\infty} c_i s^i$, where $c_i \in \mathbb{C}^{s \times t}$. The key to construct TPTA is to maintain the same order of tensor $\mathcal{A}$ with different powers. For this issue, we introduce t-product [22, 23] of two tensors to solve it. In addition, in order to give the definition of TPTA, we introduce a generalized linear functional in the tensor space for the first time.

This paper is organized as follows. In Section 2, we provide some preliminaries. First, we introduce the t-product of two tensors; then, we show the definitions of tensor exponential function and the Frobenius norm of a tensor. In Section 3.1, we define the tensor Padé-type approximant by using generalized linear functional; the expression of TPTA is of the form of tensor numerator and scalar denominator; and then we introduce the definition of orthogonal polynomial with respect to generalized linear functional and sketch an algorithm to compute the TPTA. Numerical examples are given and analyzed in Section 4. Finally, we finish the paper with concluding remarks in Section 5.

2. Preliminaries

There arise mainly a problem for approximating tensor exponential function. That is how to expand $e^{\mathcal{A}t}$ into the power series for order- p ($p \geq 3$) tensors. For a Symmetric and second-order tensor A , higher powers of A can be computed by the Cayley-Hamilton theorem [24], but it fails for the order- p ($p \geq 3$) tensors. Therefore, we shall utilize the t-product to obtain higher powers of order- p ($p \geq 3$) tensors in this section. Firstly, we introduce some notations and basic definitions which will be used in the sequel. Throughout this paper tensors are denoted by calligraphic letters (e.g., $\mathcal{A}$, $\mathcal{B}$), while capital letters represent matrices, and lowercase letters refer to scalars.

An order- p tensor, $\mathcal{A}$, can be written as

$$\mathcal{A} = (a_{i_1 i_2 \dots i_p}) \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_p}. \quad (4)$$

Thus, a matrix is considered a second-order tensor, and a vector is a first-order tensor [22], for $i = 1, \dots, n_p$, denoted by $\mathcal{A}_i \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_{p-1}}$, the tensor whose order is $p-1$ and is created by holding the p th index of $\mathcal{A}$ fixed at i . For example, consider a third-order tensor, $\mathcal{A} = (a_{i_1 i_2 i_3}) \in \mathbb{R}^{3 \times 3 \times 3}$. Fixing the 3rd index of $\mathcal{A}$, we can get three 3×3 matrices, namely, 2-order tensor, which are A_1 , A_2 , and A_3 and with elements

$$\begin{aligned} A_1 &: a_{111} \ a_{121} \ a_{131} \ a_{211} \ a_{221} \ a_{231} \ a_{311} \ a_{321} \ a_{331}, \\ A_2 &: a_{112} \ a_{122} \ a_{132} \ a_{212} \ a_{222} \ a_{232} \ a_{312} \ a_{322} \ a_{332}, \\ A_3 &: a_{113} \ a_{123} \ a_{133} \ a_{213} \ a_{223} \ a_{233} \ a_{313} \ a_{323} \ a_{333}, \end{aligned} \quad (5)$$

respectively.

Now, we will define the t-product of two tensors.

Definition 1 (see [22, 23]). Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_p}$. Then the block circulant pattern tensor of $\mathcal{A}$ is denoted by

$$\text{circ}(\mathcal{A}) = \begin{pmatrix} \mathcal{A}_1 & \mathcal{A}_{n_p} & \mathcal{A}_{n_p-1} & \dots & \mathcal{A}_2 \\ \mathcal{A}_2 & \mathcal{A}_1 & \mathcal{A}_{n_p} & \dots & \mathcal{A}_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathcal{A}_{n_p} & \mathcal{A}_{n_p-1} & \dots & \mathcal{A}_2 & \mathcal{A}_1 \end{pmatrix}, \quad (6)$$

where $\mathcal{A}_i \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_{p-1}}$, $i = 1, 2, \dots, n_p$.

Define $\text{unfold}(\cdot)$ to an $n_1 \times n_2 \times \dots \times n_p$ tensor by an $n_1 n_p \times n_2 \times \dots \times n_{p-1}$ block tensor in the following way:

$$\text{unfold}(\mathcal{A}) = \begin{pmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \\ \vdots \\ \mathcal{A}_{n_p} \end{pmatrix}. \quad (7)$$

If $\mathcal{A}$ is order-3 tensor, then $\text{unfold}(\mathcal{A})$ is a block vector. Similarly, define $\text{fold}(\cdot)$ as the inverse operation, which takes an $n_1 n_p \times n_2 \times \dots \times n_{p-1}$ block tensor and returns an $n_1 \times n_2 \times \dots \times n_p$ tensor; then

$$\text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A}. \quad (8)$$

Definition 2 (see [23]). Let $\mathcal{A}$ be $\mathbb{R}^{n_1 \times n_2 \times \dots \times n_p}$ and $\mathcal{B}$ be $\mathbb{R}^{n_2 \times l \times n_3 \times \dots \times n_p}$. Then the t-product $\mathcal{A}^* \mathcal{B}$ is the $n_1 \times l \times n_3 \times \dots \times n_p$ tensor defined recursively as

$$\mathcal{A}^* \mathcal{B} = \text{fold}(\text{circ}(\mathcal{A})^* \text{unfold}(\mathcal{B})). \quad (9)$$

Remark 3. If $\mathcal{A}$ and $\mathcal{B}$ are order-2 tensors, then the product “*” can be replaced by standard matrix multiplication.

Remark 4. The k times power of $\mathcal{A}$ is defined as $\mathcal{A}^k = \mathcal{A}^* \mathcal{A}^* \dots^* \mathcal{A}$ (k times); “*” denotes the t-product.

Example 5. Letting

$$\mathcal{A} = \left(\begin{array}{ccc|ccc} 1 & 0 & -1 & 3 & 1 & 0 \\ 2 & 1 & 0 & 2 & 0 & 0 \\ 0 & -1 & 1 & 1 & -1 & 2 \end{array} \right) \in \mathbb{R}^{3 \times 3 \times 2}, \quad (10)$$

then, from Definition 2, we have

$$\begin{aligned} \mathcal{A}^4 &= \mathcal{A}^* \mathcal{A}^* \mathcal{A}^* \mathcal{A} \\ &= \left(\begin{array}{ccc|ccc} 271 & 113 & -98 & 237 & 104 & -119 \\ 242 & 97 & -96 & 242 & 96 & -96 \\ -73 & -46 & 56 & -94 & -52 & 43 \end{array} \right). \end{aligned} \quad (11)$$

Remark 6. One of the characteristic features of t-product is that it ensures that the order of multiplication result of two tensors does not change, whereas other tensor multiplications do not have the feature; that is why we chose the t-product as the multiplication of tensors.

The tensor exponential function is a tensor function on tensors analogous to the ordinary exponential function, which can be defined as follows.

Definition 7. Let $\mathcal{A}$ be an $n_1 \times n_2 \times \cdots \times n_p$ real or complex tensor. The tensor exponential function of t , denoted by $e^{\mathcal{A}t}$ or $\exp(\mathcal{A}t)$, is the $n_1 \times n_2 \times \cdots \times n_p$ tensor given by the power series

$$e^{\mathcal{A}t} = \sum_{k=0}^{\infty} \frac{1}{k!} (\mathcal{A}t)^k, \quad (12)$$

where $\mathcal{A}^0$ is defined to be the identity tensor $\mathcal{I}$ (see Definition 8) with the same orders as $\mathcal{A}$.

Definition 8 (see [23]). The $n \times n \times l_1 \times \cdots \times l_{p-2}$ order- p identity tensor ($p > 3$) $\mathcal{I}$ is the tensor such that $\mathcal{I}_1$ is the $n \times n \times l_1 \times \cdots \times l_{p-3}$ order- $(p-1)$ identity tensor, and $\mathcal{I}_j$ is the order- $(p-1)$ zero tensor, for $j = 2, 3, \dots, l_{p-2}$.

By Definition 8, we can define tensor inverse, transpose, and orthogonality. However, we do not discuss these works here, as it is beyond the scope of the present work. For the details of these definitions of tensor, we refer to reader to [22, 23, 25] and the references therein.

Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$; then the norm of a tensor is the square root of the sum of the squares of all its entries [25]; i.e.,

$$\|\mathcal{A}\| = \sqrt{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_p=1}^{n_p} a_{i_1 i_2 \cdots i_p}^2}. \quad (13)$$

This is analogous to the matrix Frobenius norm. The inner product of two same-sized tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$ is the sum of the products of their elements [25]; i.e.,

$$(\mathcal{A}, \mathcal{B}) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_p=1}^{n_p} a_{i_1 i_2 \cdots i_p} b_{i_1 i_2 \cdots i_p}. \quad (14)$$

It follows immediately that $(\mathcal{A}, \mathcal{A}) = \|\mathcal{A}\|^2$.

3. Tensor Padé-Type Approximant

Let $f(x)$ be a given power series with tensor coefficients; i.e.,

$$f(x) = \mathcal{A}_0 + \mathcal{A}_1 x + \mathcal{A}_2 x^2 + \cdots + \mathcal{A}_n x^n + \cdots, \quad (15)$$

$$\mathcal{A}_i \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}, \quad x \in \mathbb{R}.$$

Let $\mathbf{P}$ denote the set of scalar polynomials in one real variable whose coefficients belong to the real field $\mathbb{R}$ and $\mathbf{P}_k$ denote the set of elements of $\mathbf{P}$ of degree less than or equal to k .

Let $\phi : \mathbf{P} \rightarrow \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$ be a linear functional on $\mathbf{P}$. Let it act on t by

$$\phi(t^i) = \mathcal{A}_i, \quad i = 0, 1, 2, \dots \quad (16)$$

Then, by the linearity of ϕ , we have

$$\begin{aligned} \phi((1 - tx)^{-1}) &= \phi(1 + tx + (tx)^2 + \cdots + (tx)^n + \cdots) \\ &= \mathcal{A}_0 + \mathcal{A}_1 x + \mathcal{A}_2 x^2 + \cdots + \mathcal{A}_n x^n \\ &\quad + \cdots = f(x). \end{aligned} \quad (17)$$

3.1. Definition of Tensor Padé-Type Approximant. Let $v_n(x) = b_0 + b_1 x + \cdots + b_n x^n$ ($b_n \neq 0$) be a scalar polynomial of $\mathbf{P}_n$ of exact degree n . In this case, $v_n(x)$ is said to be quasi-monic. Define the tensor polynomial $\mathcal{W}_{n-1}(x)$ associated with $v_n(x)$ with tensor-valued coefficients, by

$$\mathcal{W}_{n-1}(x) = \phi\left(\frac{v_n(x) - v_n(t)}{x - t}\right). \quad (18)$$

It is easily seen that $\mathcal{W}_{n-1}(x)$ is a tensor polynomial of exact degree $n-1$ in x . Set

$$\tilde{v}_n(x) = x^n v_n(x^{-1}), \quad (19)$$

$$\widetilde{\mathcal{W}}_{n-1}(x) = x^{n-1} \mathcal{W}_{n-1}(x^{-1}). \quad (20)$$

Then, the polynomials $\tilde{v}_n(x)$ and $\widetilde{\mathcal{W}}_{n-1}(x)$ are obtained from $v_n(x)$ and $\mathcal{W}_{n-1}(x)$, respectively, by reversing the numbering of the coefficients. By the procedure given above, the following conclusion is obtained.

Theorem 9. Let $\tilde{v}_n(0) \neq 0$; then $\widetilde{\mathcal{W}}_{n-1}(x)/\tilde{v}_n(x) - f(x) = O(x^n)$.

Proof. Expanding $(v_n(x) - v_n(t))/(x - t)$ in (18) and applying ϕ yields that

$$\begin{aligned} \widetilde{\mathcal{W}}_{n-1}(x) &= \sum_{l=0}^{n-1} \left(\sum_{i=0}^{n-l-1} b_{l+i+1} \mathcal{A}_i \right) x^l \\ &= \sum_{l=0}^{n-1} \left(\sum_{i=0}^l b_{n-l+i} \mathcal{A}_i \right) x^l. \end{aligned} \quad (21)$$

Computing $\tilde{v}_n(x)f(x)$, we get

$$\begin{aligned} \tilde{v}_n(x)f(x) &= \left(\sum_{j=0}^n b_{n-j} x^j \right) \left(\sum_{i=0}^{\infty} \mathcal{A}_i x^i \right) \\ &= \sum_{l=0}^{\infty} \left(\sum_{i=0}^l b_{n-l+i} \mathcal{A}_i \right) x^l. \end{aligned} \quad (22)$$

Thus,

$$\begin{aligned} \tilde{v}_n(x)f(x) - \widetilde{\mathcal{W}}_{n-1}(x) &= \sum_{l=n}^{\infty} \left(\sum_{i=0}^l b_{n-l+i} \mathcal{A}_i \right) x^l \\ &= O(x^n). \end{aligned} \quad (23)$$

□

Definition 10. $R_{n-1,n}(x) = \widetilde{\mathcal{W}}_{n-1}(x)/\tilde{v}_n(x)$ is called a tensor Padé-type approximant with order n for the given power series (15) and is denoted by $(n-1/n)_f(x)$.

Remark 11. The polynomial $v_n(x)$, called the generating polynomial of $(n-1/n)_f$ with respect to power series $f(x)$, can be arbitrarily chosen.

Remark 12. The tensor Padé-type approximant $(n-1/n)_f$ possesses the degree constraint, which is caused by its construction process. The constraint implies that the method does not construct tensor Padé-type approximant of type (m/n) when m is different from $n-1$.

To fill this gap, we define a new tensor Padé-type approximant by introducing a generalized linear functional.

Let $\phi^{(q)} : \mathbf{P} \rightarrow \mathbb{R}^{n_1 \times n_2 \times \dots \times n_p}$ be a generalized linear functional on $\mathbf{P}$. Let it act on t by

$$\phi^{(q)}(t^i) = \mathcal{A}_{q+i}, \quad i = 0, 1, 2, \dots \quad (24)$$

Similarly to what was done for $\mathcal{W}_{n-1}(x)$, we consider the polynomial $\mathcal{W}_q(x)$ associated with $v_n(x)$, and defined by

$$\mathcal{W}_q(x) = \phi^{(q)}\left(\frac{v_n(x) - v_n(t)}{x - t}\right), \quad q = m - n + 1. \quad (25)$$

Set

$$\widetilde{\mathcal{W}}_q(x) = x^{n-1} \mathcal{W}_q(x^{-1}), \quad (26)$$

and define

$$\mathcal{P}_{mn}(x) = \tilde{v}_n(x) \sum_{i=0}^{m-n} \mathcal{A}_i x^i + x^{m-n+1} \widetilde{\mathcal{W}}_q(x), \quad m \geq n. \quad (27)$$

Then we have the following conclusion.

Theorem 13. Let $\tilde{v}_n(0) \neq 0$; then $\mathcal{P}_{mn}(x)/\tilde{v}_n(x) - f(x) = O(x^{m+1})$.

Proof. Let f_{m-n+1} be the formal power series

$$f_{m-n+1}(x) = \sum_{j=0}^{\infty} \mathcal{A}_{m-n+1+j} x^j. \quad (28)$$

Then

$$x^{m-n+1} f_{m-n+1}(x) = f(x) - \sum_{i=0}^{m-n} \mathcal{A}_i x^i. \quad (29)$$

Expanding (25) and using (26) we obtain

$$\widetilde{\mathcal{W}}_q(x) = \sum_{q=0}^{n-1} \left(\sum_{i=0}^q b_{n-q+i} \mathcal{A}_{m-n+1+i} \right) x^q. \quad (30)$$

Computing the product $\tilde{v}_n(x) f_{m-n+1}(x)$, we find that

$$\tilde{v}_n(x) f_{m-n+1}(x) = \sum_{q=0}^{\infty} \left(\sum_{i=0}^q b_{n-q+i} \mathcal{A}_{m-n+1+i} \right) x^q. \quad (31)$$

By Theorem 9, one has

$$\frac{\widetilde{\mathcal{W}}_q(x)}{\tilde{v}_n(x)} = \left(n - \frac{1}{n} \right)_{f_{m-n+1}}(x). \quad (32)$$

Then, for $m \geq n$ we deduce from (27) and (29) that

$$\begin{aligned} & \tilde{v}_n(x) f(x) - \mathcal{P}_{mn}(x) \\ &= \tilde{v}_n(x) \left(\sum_{i=0}^{m-n} \mathcal{A}_i x^i + x^{m-n+1} f_{m-n+1}(x) \right) \\ & \quad - \tilde{v}_n(x) \sum_{i=0}^{m-n} \mathcal{A}_i x^i + x^{m-n+1} \widetilde{\mathcal{W}}_q(x) \\ &= x^{m-n+1} \left(\tilde{v}_n(x) f_{m-n+1} - \widetilde{\mathcal{W}}_q(x) \right) \\ &= x^{m-n+1} \left\{ \sum_{q=n}^{\infty} \left(\sum_{i=0}^q b_{n-q+i} \mathcal{A}_{m-n+1+i} \right) x^q \right\} \\ &= O(x^{m+1}). \end{aligned} \quad (33)$$

□

Now, we can achieve $(m/n)_f(x)$ by the above procedure, and it will be denoted by $\mathcal{P}_{mn}(x)/\tilde{v}_n(x)$.

Definition 14. $R_{m,n}(x) = \mathcal{P}_{mn}(x)/\tilde{v}_n(x)$ is called TPTA with order $m+1$ and is denoted by $(m/n)_f(x)$.

Algorithm 15 (compute $R_{m,n}(x) = \mathcal{P}_{mn}(x)/\tilde{v}_n(x)$ with $v_n(x)$ being arbitrarily chosen).

- (1) Set $q = m - n + 1$ and chose a quasi-monic polynomial $v_n(x)$.
- (2) Use (19) to compute $\tilde{v}_n(x)$.
- (3) Compute $\mathcal{W}_q(x)$ and $\widetilde{\mathcal{W}}_q(x)$ by (25) and (26), respectively.
- (4) Substitute $\tilde{v}_n(x)$ and $\widetilde{\mathcal{W}}_q(x)$ into (27) to compute $\mathcal{P}_{mn}(x)$.
- (5) Set $R_{m,n}(x) = \mathcal{P}_{mn}(x)/\tilde{v}_n(x)$.

Example 16. Let

$$\begin{aligned} f(x) &= \left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 2 & -1 & 2 \end{array} \middle| \begin{array}{cc} 1 & -1 \\ 2 & 1 \end{array} \right) \\ & \quad + \left(\begin{array}{cc|cc} 1 & 0 & 2 & 1 \\ 1 & 2 & -1 & 2 \end{array} \middle| \begin{array}{cc} 1 & -2 \\ 3 & 1 \end{array} \right) x \\ & \quad + \left(\begin{array}{cc|cc} 1 & 2 & 1 & 1 \\ 0 & 2 & -1 & 3 \end{array} \middle| \begin{array}{cc} 0 & -1 \\ 2 & 0 \end{array} \right) x^2 + \dots \\ &= \mathcal{A}_0 + \mathcal{A}_1 x + \mathcal{A}_2 x^2 + \dots \end{aligned} \quad (34)$$

Now we apply Algorithm 15 to compute TPTA of type $(2/2)$ for this example.

(1) Chose $v_2(x) = x^2 - 2x + 4$, $q = m - n + 1 = 1$.

(2) Use (19) to compute $\tilde{v}_2(x)$:

$$\tilde{v}_2(x) = 1 - 2x + 4x^2. \quad (35)$$

(3) By using (25) and (26) we get

$$\begin{aligned} \mathcal{W}_1(x) &= \left(\begin{array}{cc|cc|cc} x-1 & 2 & 2x+1 & x-1 & x-2 & 1-2x \\ x & 2x-2 & -x+1 & 2x-1 & 3x-2 & x-2 \end{array} \right), \end{aligned} \quad (36)$$

and

$$\begin{aligned} \widetilde{\mathcal{W}}_1(x) &= \left(\begin{array}{cc|cc|cc} 1-x & 2x & x & 1-x & 1-2x & x-1 \\ 0 & 2-2x & x-1 & 2-x & 2-2x & 1-2x \end{array} \right). \end{aligned} \quad (37)$$

(4) Substituting $\tilde{v}_2(x)$ and $\widetilde{\mathcal{W}}_1(x)$ into (27), we obtain

$$\begin{aligned} \mathcal{P}_{22}(x) &= \tilde{v}_2(x) \mathcal{A}_0 + x \widetilde{\mathcal{W}}_1(x) \\ &= \left(\begin{array}{cc|cc|cc} 3x^2-x+1 & 2x^2 & -3x^2+2x & 3x^2-x+1 & 2x^2-x+1 & -x^2-1 \\ -2x^2+x & 6x^2-2x+2 & -3x^2+x-1 & 7x^2-2x+2 & 4x^2-x+2 & 2x^2-x+1 \end{array} \right). \end{aligned} \quad (38)$$

(5) Set $R_{22}(x) = \mathcal{P}_{22}(x)/\tilde{v}_2(x)$. It is easy to verify that

$$\frac{\mathcal{P}_{22}(x)}{\tilde{v}_2(x)} - f(x) = O(x^3). \quad (39)$$

3.2. Algorithm for Computing TPTA. Generally, the precision of TPTA is limited, since the denominator polynomials of TPTA are arbitrarily prescribed. In this subsection, in order to improve the precision of approximation, we propose an algorithm for computing the denominator polynomials and illustrate the efficiency of this algorithm in next section.

First, we give the following conclusion.

Theorem 17 (error formula).

$$f(x) - \left(n - \frac{1}{n}\right)_f(x) = \frac{x^n}{\tilde{v}_n(x)} \phi\left(\frac{v_n(t)}{1-tx}\right). \quad (40)$$

Proof. Note that ϕ is a linear functional on $\mathbf{P}$, only acting on t . From (18) and (20) we deduced that

$$\begin{aligned} \widetilde{\mathcal{W}}_{n-1}(x) &= x^{n-1} \mathcal{W}_{n-1}(x^{-1}) \\ &= x^{n-1} \phi\left(\frac{x v_n(x^{-1}) - x v_n(t)}{1-tx}\right) \\ &= \phi\left(\frac{x^n v_n(x^{-1}) - x^n v_n(t)}{1-tx}\right) \\ &= \phi\left(\frac{x^n v_n(x^{-1})}{1-tx}\right) - x^n \phi\left(\frac{v_n(t)}{1-tx}\right) \\ &= \tilde{v}_n(x) f(x) - x^n \phi\left(\frac{v_n(t)}{1-tx}\right), \end{aligned} \quad (41)$$

and then this error formula holds. $\square$

In terms of the error formula, it holds that

$$\begin{aligned} f(x) - \left(n - \frac{1}{n}\right)_f(x) &= \frac{x^n}{\tilde{v}_n(x)} \phi\left(\frac{v_n(t)}{1-tx}\right) = \frac{x^n}{\tilde{v}_n(x)} \\ &\cdot \phi(v_n(t) + v_n(t)tx + v_n(t)t^2x^2 + \dots) \\ &= \frac{x^n}{\tilde{v}_n(x)} (\phi(v_n(t)) + \phi(v_n(t)t)x \\ &\quad + \phi(v_n(t)t^2)x^2 + \dots). \end{aligned} \quad (42)$$

If we impose that $v_n(t)$ satisfies the condition $\phi(v_n(t)) = 0$, then the first term of (42) disappears, and the order of approximation becomes $n+1$. If, in addition, we also impose the condition $\phi(v_n(t)t) = 0$, the second term in the expansion of the error also disappears, and the order of approximation becomes $n+2$, and so on. We indicate that $v_n(x)$ depends on $n+1$ arbitrary constants; however, on the other side, a rational function is defined apart from a multiplying factor in its numerator and its denominator. It implies that $(n-1/n)_f(x)$ depends on n arbitrary constants. So let us take $v_n(t)$ such that

$$\phi(v_n(t)t^k) = 0, \quad k = 0, 1, 2, \dots, n-1. \quad (43)$$

Definition 18. $v_n(t)$ in (43) is called an orthogonal polynomial with respect to the linear functional ϕ and $(n-1/n)_f(x)$ in (42) is also called a TPTA for the given power series (15) when (43) is satisfied.

From (43) we obtain

$$\phi(v_n(t)t^k) = \sum_{i=0}^n b_i \mathcal{A}_{i+k} = 0, \quad k = 0, 1, 2, \dots, n-1. \quad (44)$$

Let $b_n = 1$ in (44); then it follows that

$$\sum_{i=0}^{n-1} b_i \mathcal{A}_{i+k} = -\mathcal{A}_{k+n}, \quad k = 0, 1, 2, \dots, n-1. \quad (45)$$

Forming the scalar product of both sides of (45) with $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_{n-1}$, respectively, we get

$$\sum_{i=0}^{n-1} b_i (\mathcal{A}_{i+k}, \mathcal{A}_k) = -(\mathcal{A}_{k+n}, \mathcal{A}_k), \quad (46)$$

$$k = 0, 1, 2, \dots, n-1.$$

Denote

$$H_n(\mathcal{A}_0) = \begin{pmatrix} (\mathcal{A}_0, \mathcal{A}_0) & (\mathcal{A}_1, \mathcal{A}_0) & \cdots & (\mathcal{A}_{n-1}, \mathcal{A}_0) \\ (\mathcal{A}_1, \mathcal{A}_1) & (\mathcal{A}_2, \mathcal{A}_1) & \cdots & (\mathcal{A}_n, \mathcal{A}_1) \\ \vdots & \vdots & \ddots & \vdots \\ (\mathcal{A}_{n-1}, \mathcal{A}_{n-1}) & (\mathcal{A}_n, \mathcal{A}_{n-1}) & \cdots & (\mathcal{A}_{2n-2}, \mathcal{A}_{n-1}) \end{pmatrix}, \quad (47)$$

$$\vec{h}_n = \begin{pmatrix} -(\mathcal{A}_n, \mathcal{A}_0) \\ -(\mathcal{A}_{n+1}, \mathcal{A}_1) \\ \vdots \\ -(\mathcal{A}_{2n-1}, \mathcal{A}_{n-1}) \end{pmatrix},$$

and call $\det(H_n(\mathcal{A}_0))$ the Hankel determinant of $f(x)$ with respect to the coefficients $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_{n-1}$.

Then (46) is converted into

$$H_n(\mathcal{A}_0) x = \vec{h}_n, \quad x = (b_0, b_1, \dots, b_{n-1})^T. \quad (48)$$

In the case of TPTA, $v_n(x)$ is not arbitrarily chosen any more but is determined by the preceding system. The choice of $v_n(x)$ can help to improve the accuracy of approximation, but unfortunately we have not been able to guarantee that the solution of system (48) comes into existence, so far. We only give the following basic theorem about system (48) on the basis of linear algebra.

Theorem 19. *The solution of (48) exists if and only if $\text{rank}(H_n(\mathcal{A}_0)) = \text{rank}(H_n(\mathcal{A}_0) : \vec{h}_n)$. Moreover, the solution is unique if $\det(H_n(\mathcal{A}_0)) \neq 0$.*

Proof. The proof of the assertion follows from the simple fact that, for a system of linear equations, described by $Ax = b$, where A is matrix and x, b are vectors, the solution of system (48) comes into existence for x if and only if $\text{rank}(A) = \text{rank}(A : b)$; i.e., the right-hand vector must be in the vector space spanned by the columns of the coefficient matrix A . Moreover, if $\det(H_n(\mathcal{A}_0)) \neq 0$, according to Cramer's rule, the solution is unique. $\square$

Theorem 20 (existence). *Let $f(x)$ be given power series (15); then $(n-1/n)_f$ and $(m/n)_f$ exist and are unique if and only if $\det(H_n(\mathcal{A}_0)) \neq 0$.*

Proof. “ $\Leftarrow$ ” By Theorem 19, if $\det(H_n(\mathcal{A}_0)) \neq 0$, it means that nonhomogeneous equation (48) exists as a unique solution $b_0, b_1, \dots, b_{n-1}$ for $v_n(x)$. From (19), it also means that $\tilde{v}_n(x)$ and $\tilde{\mathcal{W}}_{n-1}(x)$ exist. Hence, by the construction of $(n-1/n)_f$, existence holds.

“ $\Rightarrow$ ” Let $(n-1/n)_f = \tilde{\mathcal{W}}_{n-1}(x)/\tilde{v}_{n-1}(x)$ exist and be unique; then it implies that

$$\text{rank}(H_n(\mathcal{A}_0)) = \text{rank}(H_n(\mathcal{A}_0) : \vec{h}_n), \quad (49)$$

and if $\det(H_n(\mathcal{A}_0)) = 0$, then $\text{rank}(H_n(\mathcal{A}_0) : \vec{h}_n) < n$; the fact that equation (46) has $n - \text{rank}(H_n(\mathcal{A}_0) : \vec{h}_n)$ solutions, namely, that we can construct $n - \text{rank}(H_n(\mathcal{A}_0) : \vec{h}_n)$ generating polynomials, which is contradictory to the uniqueness of $(n-1/n)_f$, holds.

The proof of existence and uniqueness of $(m/n)_f$ is similar to the preceding process. $\square$

Theorem 21. *Let $\det(H_n(\mathcal{A}_0)) \neq 0$; then $(n-1/n)_f(x) = \tilde{\mathcal{W}}_{n-1}(x)/\tilde{v}_n(x)$, where the generating polynomial $v_n(x)$ is given by*

$$v_n(x) = \det \begin{pmatrix} H_n(\mathcal{A}_0) & \vec{\gamma} \\ \vec{\delta}^T & x^n \end{pmatrix}, \quad (50)$$

where $\vec{\gamma} = ((\mathcal{A}_n, \mathcal{A}_0), (\mathcal{A}_{n+1}, \mathcal{A}_1), \dots, (\mathcal{A}_{2n-1}, \mathcal{A}_{n-1}))^T$, $\vec{\delta}^T = (1, x, \dots, x^{n-1})$, and $\tilde{v}_n(x)$ and $\tilde{\mathcal{W}}_{n-1}(x)$ are given by (19) and (20), respectively.

Now, we can derive an algorithm to calculate $(m/n)_f(x)$ using (26), (27), and (50).

Algorithm 22 (compute $(m/n)_f(x) = \mathcal{P}_{mn}(x)/\tilde{v}_n(x)$).

- (1) Use (14) to calculate $H_n(\mathcal{A}_0)$ and $\vec{\gamma}$.
- (2) Use (50) and (19) to compute $v_n(x)$ and $\tilde{v}_n(x)$, respectively.
- (3) Set $q = m - n + 1$ and compute $\mathcal{W}_q(x)$ and $\tilde{\mathcal{W}}_q(x)$ by

$$\mathcal{W}_q(x) = \phi^{(m-n+1)} \left(\frac{v_n(x) - v_n(t)}{x - t} \right), \quad (51)$$

$$\tilde{\mathcal{W}}_q(x) = x^{n-1} \mathcal{W}_q(x^{-1}).$$

- (4) Compute the numerator of TPTA by

$$\mathcal{P}_{mn}(x) = \tilde{v}_n(x) \sum_{i=0}^{m-n} \mathcal{A}_i x^i + x^{m-n+1} \tilde{\mathcal{W}}_q(x). \quad (52)$$

- (5) Obtain $(m/n)_f(x) = \mathcal{P}_{mn}(x)/\tilde{v}_n(x)$.

4. Application for Computing the Tensor Exponential Function

The method of truncated infinite series has abroad applications in finite single crystal plasticity for computing tensor exponential function [16]. However, the accuracy and effectiveness of such algorithm are limited by round-off and choice of termination criterion. In this section, we will utilize the

method of TPTA to compute tensor exponential function. We start by briefly reviewing some basic equations that model the behaviour of single crystals in the finite strain range [16].

Consider a single crystal model

$$F = F^e F^p, \quad (53)$$

where F^e and F^p denote elastic part and plastic part, respectively.

For a single crystal with a total number n_{syst} of slip systems, the evolution of the inelastic deformation gradient, F^p , is defined by means of the following rate form:

$$\dot{F}^p F^{p-1} = \sum_{\alpha=1}^{n_{syst}} \dot{\gamma}^\alpha s_0^\alpha \otimes m_0^\alpha, \quad (54)$$

where $\dot{\gamma}^\alpha$ denotes the contribution of slip system α to the total inelastic rate of deformation. The vectors s_0^α and m_0^α denote, respectively, the slip direction and normal direction of slip system α .

The above tensor differential equation can be discretized in an implicit fashion with use of the tensor exponential function. The implicit exponential approximation to the inelastic flow equation results in the following discrete form:

$$F_{n+1}^p = \exp \left(\sum_{\alpha=1}^{n_{syst}} \Delta \gamma^\alpha s_0^\alpha \otimes m_0^\alpha \right) F_n^p. \quad (55)$$

The above formula is analogous to the exact solution of initial value problem (1) and it is necessary to calculate

$$\exp \left(\sum_{\alpha=1}^{n_{syst}} \Delta \gamma^\alpha s_0^\alpha \otimes m_0^\alpha \right). \quad (56)$$

In [7], the author used Algorithm 23 to calculate (56).

Algorithm 23 (truncated infinite series method [7] (p.749)).

- (1) Given tensor $\mathcal{X}$, initialise $n = 0$ and $\exp(\mathcal{X}) := \mathcal{I}$.
- (2) Increment counter $n := n + 1$.
- (3) Compute $n!$ and $\mathcal{X}^n$.
- (4) Add new term to the series $\exp(\mathcal{X}) := \exp(\mathcal{X}) + (1/n!) \mathcal{X}^n$.
- (5) Check convergence, if

$$\frac{\|\mathcal{X}^n\|}{n!} < \epsilon_{tol}, \quad \text{then exit, else goto (1)}. \quad (57)$$

Example 24. Consider a tensor exponential function $\exp(\mathcal{A}x)$; the entries of $\mathcal{A}$ are $a_{121} = 1$, $a_{221} = -2$, $a_{122} = 2$, $a_{222} = -1$, and zero elsewhere.

To find a tensor Padé-type approximation of type (3/3) for the tensor exponential function, first we should expand $\exp(\mathcal{A}x)$ into power series by means of Definition 7. We can obtain

$$\begin{aligned} \exp(\mathcal{A}x) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 2 \\ 0 & 0 & -1 \end{pmatrix} x \\ &+ \begin{pmatrix} 0 & -2 & 0 \\ 0 & 5 & 2 \\ 0 & 2 & 0 \end{pmatrix} x^2 \\ &+ \begin{pmatrix} 0 & \frac{13}{6} & 0 \\ 0 & \frac{14}{6} & \frac{14}{6} \\ 0 & -\frac{14}{6} & -\frac{13}{6} \end{pmatrix} x^3 \\ &+ \begin{pmatrix} 0 & -\frac{40}{24} & 0 \\ 0 & \frac{24}{41} & \frac{24}{40} \\ 0 & \frac{24}{41} & -\frac{40}{24} \end{pmatrix} x^4 + \dots \\ &= \mathcal{A}_0 + \mathcal{A}_1 x + \mathcal{A}_2 x^2 + \mathcal{A}_3 x^3 + \mathcal{A}_4 x^4 \\ &+ \dots \end{aligned} \quad (58)$$

By Algorithm 22, the following can be done.

- (1) Use (14) to compute $H_3(\mathcal{A}_0)$ and $\vec{\gamma}$:

$$\begin{aligned} H_3(\mathcal{A}_0) &= \begin{pmatrix} 2 & -2 & \frac{5}{2} \\ 10 & -14 & \frac{41}{3} \\ \frac{41}{2} & -\frac{61}{3} & \frac{365}{24} \end{pmatrix}, \\ \vec{\gamma} &= \begin{pmatrix} -\frac{7}{3} \\ \frac{61}{6} \\ -\frac{547}{60} \end{pmatrix}. \end{aligned} \quad (59)$$

- (2) Use (50) to calculate $v_3(x)$:

$$v_3(x) = \frac{15041}{1080} + \frac{3947}{60}x + \frac{1189}{10}x^2 + \frac{1493}{18}x^3, \quad (60)$$

and compute $\tilde{v}_3(x)$ by (19), so we get

$$\tilde{v}_3(x) = \frac{15041}{1080}x^3 + \frac{3947}{60}x^2 + \frac{1189}{10}x + \frac{1493}{18}. \quad (61)$$

- (3) Set $q = m - n + 1 = 1$ and compute $\mathcal{W}_1(x)$, $\widetilde{\mathcal{W}}_1(x)$:

$$\mathcal{W}_1(x) = \begin{pmatrix} 0 & \frac{1493}{18}x^2 - \frac{4229}{90}x + \frac{1039}{135} & 0 \\ 0 & -\frac{1493}{9}x^2 - \frac{5479}{180}x - \frac{15041}{540} & 0 \\ 0 & -\frac{1493}{18}x^2 + \frac{5479}{90}x - \frac{15041}{135} & 0 \end{pmatrix}, \quad (62)$$

TABLE 1: Numerical results of Example 24 at different points by using Algorithm 22.

x		(1, 2, 1)	(2, 2, 1)	(1, 2, 2)	(2, 2, 2)	Res
0.1	$e^{\mathcal{A}x}$	0.08200959	0.82282781	0.17717281	-0.08200959	8.34e-12
	$(3/3)_{e^{\mathcal{A}x}}(x)$	0.08200778	0.82282688	0.17717311	-0.08200778	
0.2	$e^{\mathcal{A}x}$	0.13495955	0.68377119	0.31622880	-0.13495955	1.62e-9
	$(3/3)_{e^{\mathcal{A}x}}(x)$	0.13493452	0.68375764	0.31624235	-0.13493452	
0.3	$e^{\mathcal{A}x}$	0.16712428	0.57369394	0.42630605	-0.16712428	3.14e-8
	$(3/3)_{e^{\mathcal{A}x}}(x)$	0.16701602	0.57363058	0.42636941	-0.16701602	
0.4	$e^{\mathcal{A}x}$	0.18456291	0.48575712	0.51424287	-0.18456291	2.38e-7
	$(3/3)_{e^{\mathcal{A}x}}(x)$	0.18427224	0.48557038	0.51442961	-0.18427224	

TABLE 2: The exact value of $\exp(\mathcal{A})$.

	(1, 2, 1)	(2, 2, 1)	(1, 2, 2)	(2, 2, 2)
$\exp(\mathcal{A})$	0.15904705	0.20883238	0.79116761	-0.15904705

and

$$\widetilde{\mathcal{W}}_1(x) = \begin{pmatrix} 0 & \frac{1493}{18} - \frac{4229}{90}x + \frac{1039}{135}x^2 & 0 & \frac{1493}{9} + \frac{5479}{180}x + \frac{15041}{540}x^2 \\ 0 & -\frac{1493}{9} - \frac{5479}{180}x - \frac{15041}{540}x^2 & 0 & -\frac{1493}{18} + \frac{4229}{90}x - \frac{1039}{135}x^2 \end{pmatrix}. \quad (63)$$

(4) Substitute $\tilde{v}_3(x)$ and $\widetilde{\mathcal{W}}_1(x)$ into (27) to compute $\mathcal{P}_{33}(x)$:

$$\mathcal{P}_{33}(x) = \begin{pmatrix} \frac{15041}{1080}x^3 + \frac{3947}{60}x^2 + \frac{1189}{10}x + \frac{1493}{18} & \frac{1039}{135}x^3 - \frac{4229}{90}x^2 + \frac{1493}{18}x & 0 & \frac{15041}{540}x^3 + \frac{5479}{180}x^2 + \frac{1493}{9}x \\ 0 & -\frac{15041}{1080}x^3 + \frac{3181}{90}x^2 - \frac{4229}{90}x + \frac{1493}{18} & 0 & -\frac{1039}{135}x^3 + \frac{4229}{90}x^2 - \frac{1493}{18}x \end{pmatrix}. \quad (64)$$

(5) Obtain $(m/n)_f(x) = \mathcal{P}_{33}(x)/\tilde{v}_3(x)$.

In Table 1 we compare the number of exact figures given by the method of TPTA of type (3/3) with corresponding exact value of $\exp(\mathcal{A}x)$ referring to the entries of (1, 2, 1), (2, 2, 1), (1, 2, 2), and (2, 2, 2). We also compute the norm of absolute residual tensor (denoted by Res). Here,

$$Res(x_i) = \left\| \exp(\mathcal{A}x_i) - \left(\frac{3}{3}\right)_{e^{\mathcal{A}x}}(x_i) \right\|^2, \quad (65)$$

where the operation $\|\cdot\|$ is defined by (13).

From Table 1, it is observed that the estimates from TPTA can reach the desired accuracy.

Example 25. Let $f(x)$ be given by Example 24.

By Algorithm 22 for preceding example again, we calculate $(m/m)_f(1)$, $m = 1, 2, 3, 4, 5$. The exact value and approximant value associated with the entries of (1, 2, 1), (2, 2, 1), (1, 2, 2), and (2, 2, 2) are listed in Tables 2 and 3, respectively, where $x = 1$.

From Table 3, we can see that $(3/3)_{e^{\mathcal{A}x}}$ has the best approximation for this example. We also compute $\exp(\mathcal{A})$ by using Algorithm 23, and the corresponding numerical results are listed in Table 4. By comparison of Table 3 with Table 4, we find that it requires at most 6 coefficients (since $m = 3$) of power series expansion of $\exp(\mathcal{A}x)$ to achieve an error of 10^{-5} in Algorithm 22, while requiring 11 coefficients in Algorithm 23. It is straightforward to understand that Algorithm 23 is more expensive than Algorithm 22 especially for higher order tensor exponential function. In practical applications, only few coefficients of the series may be known,

TABLE 3: Numerical approximations of $\exp(\mathcal{A})$ using Algorithm 22 for Example 25.

m	(1, 2, 1)	(2, 2, 1)	(1, 2, 2)	(2, 2, 2)	Res
1	0.50000000	0.00000000	1.00000000	-0.50000000	3.19e-1
2	-2.12500000	0.75000000	0.25000000	2.12500000	1.10e+1
3	0.15503865	0.20377270	0.79622729	-0.15503865	8.33e-5
4	0.17454584	0.22682585	0.77317481	-0.17454584	1.12e-3
5	0.17625313	0.19112365	0.80887636	-0.17625313	1.21e-3

TABLE 4: Numerical results of Example 25 by using Algorithm 23.

n_{max}	(1, 2, 1)	(2, 2, 1)	(1, 2, 2)	(2, 2, 2)	Res
1	1	-1	2	2	4.33
2	-1	1.5	-0.5	1	6.02
3	1.16666666	-0.83333333	1.83333333	-1.16666666	4.20
4	-0.50000000	0.87500000	0.12500000	0.50000000	1.75
5	0.50833333	-0.14166666	1.14166666	-0.50833333	4.89e-1
6	0.00277777	0.36527777	0.63472222	-0.00277777	9.77e-2
7	0.21964285	0.14821428	0.85178571	-0.21964285	1.47e-2
8	0.13829365	0.22958829	0.77041170	-0.13829365	1.72e-3
9	0.16541280	0.20246638	0.79753361	-0.16541280	1.62e-4
10	0.15735119	0.21060267	0.78939732	-0.15735119	1.22e-5
11	0.15957013	0.20838371	0.79161628	-0.15957013	8.77e-7

so, we may get the desired results by means of TPTA. Thus the effectiveness of the proposed Algorithm 22 is verified.

5. Conclusion

In this paper, we presented tensor Padé-type approximant method for computing tensor exponential function; the expression of TPTA is of the form of tensor numerator and scalar denominator. In order to have a tensor Padé-type approximant with the higher possible precision of approximation, we proposed an algorithm for computing denominator polynomials of TPTA, and its effectiveness has been investigated in one example of tensor exponential function. The key to the TPTA to be applied to the tensor exponential function is that it can be expanded into power series with the same order tensors coefficients by means of t-product. Of course, there are several ways to multiply tensors [26–30], but the order of the resulting tensor may be changed. For example, if $\mathcal{A}$ is $n_1 \times n_2 \times n_3$ and $\mathcal{B}$ is $n_1 \times m_2 \times m_3$, then the **contracted product** [26] of $\mathcal{A}$ and $\mathcal{B}$ is $n_2 \times n_3 \times m_2 \times m_3$. So, the choice of the multiplication of two tensors is an open question for expanding tensor exponential function, and the corresponding tensor Padé approximant theoretic is a subject of further research.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work is supported by National Natural Science Foundation (11371243) and Key Disciplines of Shanghai Municipality (S30104).

References

- [1] S. Dolgov and B. Khoromskij, “Simultaneous state-time approximation of the chemical master equation using tensor product formats,” *Numerical Linear Algebra with Applications*, vol. 22, no. 2, pp. 197–219, 2015.
- [2] N. G. van Kampen, *Stochastic Process in Physics and Chemistry*, North-Holland Personal Library, Elsevier B.V., 3rd edition, 2007.
- [3] P. Gelß, S. Matera, and C. Schütte, “Solving the master equation without kinetic Monte Carlo: tensor train approximations for a CO oxidation model,” *Journal of Computational Physics*, vol. 314, pp. 489–502, 2016.
- [4] W. Ding, K. Liu, E. Belyaev, and F. Cheng, “Tensor-based linear dynamical systems for action recognition from 3D skeletons,” *Pattern Recognition*, pp. 75–86, 2017.
- [5] P. Gelß, S. Klus, S. Matera, and C. Schütte, “Nearest-neighbor interaction systems in the tensor-train format,” *Journal of Computational Physics*, vol. 341, pp. 140–162, 2017.
- [6] C. Gu, Y. Z. Huang, and Z. B. Chen, *Continued Fractional Recurrence Algorithm for Generalized Inverse Tensor Padé Approximation*, Control and Decision, 2018, <http://kns.cnki.net/kcms/detail/21.1124.TP.20180416.0932.036.html>.
- [7] E. A. de Souza, D. Peric, and D. R. J. Owen, *Computational methods for plasticity: Theory and Applications*, Wiley, 2008.
- [8] H. Chen, Y. Chen, G. Li, and L. Qi, “A semidefinite program approach for computing the maximum eigenvalue of a class of structured tensors and its applications in hypergraphs and copositivity test,” *Numerical Linear Algebra with Applications*, vol. 25, no. 1, e2125, 16 pages, 2018.

- [9] G. Zhou, G. Wang, L. Qi, and M. Alqahtani, "A fast algorithm for the spectral radii of weakly reducible nonnegative tensors," *Numerical Linear Algebra with Applications*, vol. 25, no. 2, e2134, 10 pages, 2018.
- [10] H. Chen and Y. Wang, "On computing minimal H -eigenvalue of sign-structured tensors," *Frontiers of Mathematics in China*, vol. 12, no. 6, pp. 1289–1302, 2017.
- [11] G. Wang, G. Zhou, and L. Caccetta, "Z-eigenvalue inclusion theorems for tensors," *Discrete and Continuous Dynamical Systems - Series B*, vol. 22, no. 1, pp. 187–198, 2017.
- [12] K. Zhang and Y. Wang, "An H -tensor based iterative scheme for identifying the positive definiteness of multivariate homogeneous forms," *Journal of Computational and Applied Mathematics*, vol. 305, pp. 1–10, 2016.
- [13] Y. Wang, K. Zhang, and H. Sun, "Criteria for strong H -tensors," *Frontiers of Mathematics in China*, vol. 11, no. 3, pp. 577–592, 2016.
- [14] H. Chen, L. Qi, and Y. Song, "Column sufficient tensors and tensor complementarity problems," *Frontiers of Mathematics in China*, vol. 13, no. 2, pp. 255–276, 2018.
- [15] Y. Wang, L. Caccetta, and G. Zhou, "Convergence analysis of a block improvement method for polynomial optimization over unit spheres," *Numerical Linear Algebra with Applications*, vol. 22, no. 6, pp. 1059–1076, 2015.
- [16] E. A. de Souza Neto, "The exact derivative of the exponential of an unsymmetric tensor," *Computer Methods Applied Mechanics and Engineering*, vol. 190, no. 18–19, pp. 2377–2383, 2001.
- [17] A. Cuitino and M. Ortiz, "A material-independent method for extending stress update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics," *Engineering Computations*, vol. 9, no. 4, pp. 437–451, 1992.
- [18] A. L. Eterovic and K. Bathe, "A hyperelastic-based large strain elasto-plastic constitutive formulation with combined isotropic-kinematic hardening using the logarithmic stress and strain measures," *International Journal for Numerical Methods in Engineering*, vol. 30, no. 6, pp. 1099–1114, 1990.
- [19] J. C. Simo, "Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory," *Computer Methods Applied Mechanics and Engineering*, vol. 99, no. 1, pp. 61–112, 1992.
- [20] C. Brezinski, *Padé-type approximation and general orthogonal polynomials*, vol. 50 of *International Series of Numerical Mathematics*, Birkhauser, Basel, Switzerland, 1980.
- [21] C. Gu, "Matrix Padé-type approximant and directional matrix Padé approximant in the inner product space," *Journal of Computational and Applied Mathematics*, vol. 164, pp. 365–385, 2004.
- [22] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.
- [23] C. D. Martin, R. Shafer, and B. Larue, "An order- p tensor factorization with applications in imaging," *SIAM Journal on Scientific Computing*, vol. 35, no. 1, pp. A474–A490, 2013.
- [24] M. Itskov, "Computation of the exponential and other isotropic tensor functions and their derivatives," *Computer Methods Applied Mechanics and Engineering*, vol. 192, no. 35–36, pp. 3985–3999, 2003.
- [25] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [26] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.
- [27] B. W. Bader and T. G. Kolda, "Algorithm 862: MATLAB tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software*, vol. 32, no. 4, pp. 635–653, 2006.
- [28] H. A. L. Kiers, "Towards a standardized notation and terminology in multiway analysis," *Journal of Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.
- [29] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [30] D. Liu, W. Li, and S.-W. Vong, "The tensor splitting with application to solve multi-linear systems," *Journal of Computational and Applied Mathematics*, vol. 330, pp. 75–94, 2018.

Submit your manuscripts at
www.hindawi.com