Calcolo (2024) 61:27
https://doi.org/10.1007/s10092-024-00582-4

Hosvd-tmpe: an extrapolation method for multidimensional sequences

Abdeslem Hafid Bentbib[1] · Khalide Jbilou[2,3] · Ridwane Tahiri[1]

Received: 3 April 2023 / Revised: 8 April 2024 / Accepted: 11 April 2024

Abstract
Accelerating slowly convergent sequences is one of the main purposes of extrapolation methods. In this paper, we present a new tensor polynomial extrapolation method, which is based on a modified minimisation problem and some ideas leading to the recent Tensor Global Minimal Extrapolation Method (TG-MPE). We discuss the application of our method to fixed-point iterative process. An efficient algorithm via the higher order Singular Value Decomposition (HOSVD) is proposed for its implementation. The numerical tests show clearly the effectiveness and performance of the proposed method.

Keywords Convergence acceleration · High-order SVD · N-mode product · Tensor Extrapolation method

Mathematics Subject Classification 15A69 · 65B05 · 15A29

1 Introduction

Accelerating slowly convergent sequences is one of the primary objectives of extrapolation methods. A common source of such sequences, for example, is the numerical solution of systems of linear or nonlinear equations using fixed-point iterative methods or sequences generated by quadrature techniques for integral computation. Typically,

✉ Ridwane Tahiri
ridwane.tahiri@ced.uca.ma

Abdeslem Hafid Bentbib
a.bentbib@uca.ac.ma

Khalide Jbilou
Khalide.Jbilou@univ-littoral.fr

1 Department of Mathematics, Faculty of Science and Technology Marrakech, University Cadi Ayyad, BP 549, 42 000 Marrakech, Morocco

2 Université du Littoral Cote d'Opale, LMPA, 50 rue F. Buisson, 62228 Calais-Cedex, France

3 University Mohammed VI, Benguerire, Morocco

Published online: 03 June 2024

these sequences exhibit slow convergence towards the desired fixed point. As a result, to attain a satisfactory approximation of the limits of these sequences with a prescribed level of accuracy, a substantial number of iterations is necessary. This, in turn, results in a considerable computational burden.

Formally, the principle of extrapolation methods consists of transforming the terms of a certain slowly convergent sequence (s_n) into a new sequence (t_n) that converges more rapidly to the same limit. In fact, each extrapolation method employs a specific transformation T. For a given number $k+1$ of elements from the basic sequence, the transformation T produces a new term $t_n^{(k)}$ as: $t_n^{(k)} = T(s_n, s_{n+1}, \ldots, s_{n+k})$ such that, under certain assumptions, $\frac{\|t_n^{(k)} - s\|}{\|s_n - s\|} \underset{n\to\infty}{\longrightarrow} 0$ where s is the limit of (s_n) and $\|.\|$ is a suitable norm. As demonstrated in [12], it is worth noting that a universal transformation that accelerates the convergence of all sequences cannot exist. Actually, each transformation is only able to accelerate the convergence of a limited class of sequences. This implies that it will always be important to discover and study new transformations.

For scalars, vector and matrix sequences, many extrapolation methods have been introduced, see for example [7, 27] and the references therein. As for the tensor case that interests us in this work, extrapolation methods have been recently proposed. See for instance [2, 6, 10, 11, 16–18, 20, 22, 24, 25]. The primary reason for the interest in generalizing extrapolation methods to tensor sequences stems from the significant roles they have played in various disciplines, such as color and multispectral image and video processing [28], psychometrics, chemometrics, biomedical signal processing, higher-order statistics (HOS), and econometrics [1, 14]. Consequently, it becomes highly intriguing to search for appropriate transformations that expedite the convergence of tensor sequences.

In this paper, our aim is to introduce a new extrapolation method based on the combination of Higher-Order Singular Value Decomposition (HOSVD) [13], and the tensor global minimal polynomial extrapolation method (TG-MPE) [16]. A numerically stable algorithm is proposed for the implementation of the proposed method.

The outline of this paper is organized as follows. In Sect. 2, we provide the necessary definitions, notations, and some results required for this work. In Sect. 3, after a brief overview of the tensor global MPE (TG-MPE) method [16], we develop the new HOSVD-TMPE (Higher-Order Singular Value Decomposition method which is based on Tensor Minimal Polynomial Extrapolation). Section 4 outlines the algorithm for implementing this method. We establish error analysis results in Sect. 5, and in Sect. 6, we present numerical experiments that confirm the effectiveness of the proposed method.

2 Basic definitions

This section reviews some definitions and proprieties of tensors and uses notations defined by [19]. A tensor is a multidimensional array of data. The elements of a tensor are referred by using multiple indices. The required number of indices defines the order

of a tensor. For a given (N-order) tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ we use the notation

$$\mathcal{A} = (\mathcal{A}_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N})_{1 \leqslant i_n \leqslant I_n; 1 \leqslant n \leqslant N}.$$

The values $\mathcal{A}_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N}$ are the entries of $\mathcal{A}$.

Definition 1 [19] Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ and a matrix $M \in \mathbb{R}^{J_n \times I_n}$, the n-mode product of $\mathcal{A}$ by M, denoted by $\mathcal{A} \times_n M$, is the $(I_1 \times I_2 \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)$-tensor defined by

$$(\mathcal{A} \times_n M)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} \mathcal{A}_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} M_{j_n i_n}.$$

The n-mode product of the tensor $\mathcal{A}$ by a vector $w \in \mathbb{R}^{I_n}$, denoted by $\mathcal{A} \bar{\times}_n w$, is the subtensor of order $(I_1 \times I_2 \ldots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)$ defined as

$$(\mathcal{A} \bar{\times}_n w)_{i_1 i_2 \ldots i_{n-1} i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} \mathcal{A}_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} w_{i_n}.$$

Definition 2 (n-mode unfolding matrix) For a given tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ and $1 \leq n \leq N$, the n-mode unfolding matrix of $\mathcal{A}$, is the $(I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N)$ matrix, denoted by $\mathcal{A}_{(n)}$ such that

$$\mathcal{A}_{(n)}(i_n, j) = \mathcal{A}_{i_1 \ldots i_N},$$

where $j = 1 + \sum_{k=1 k \neq n}^{N} (i_k - 1) L_k$ with $L_k = \prod_{m=1 m \neq n}^{k-1} I_m$. The columns of $\mathcal{A}_{(n)}$ are called the mode-n fibres of $\mathcal{A}$.

Definition 3 Let $\mathcal{A}$ and $\mathcal{B}$ two tensors of the same size $I_1 \times I_2 \times I_3 \times \cdots \times I_N$, the (Frobenius) inner product of $\mathcal{A}$ and $\mathcal{B}$ is the scalar defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_N=1}^{I_N} \cdots \sum_{i_1=1}^{I_1} \mathcal{A}_{i_1 \ldots i_N} \mathcal{B}_{i_1 \ldots i_N}, \text{ and } \| \mathcal{A} \|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}.$$

Proposition 1 [13] *For $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$, $M_1 \in \mathbb{R}^{J_n \times I_n}$, $M_3 \in \mathbb{R}^{J_p \times I_N}$ and $M_2 \in \mathbb{R}^{K_n \times J_n}$ ($n \neq p$), we have the following properties*

1. $(\mathcal{A} \times_n M_1) \times_p M_3 = (\mathcal{A} \times_p M_3) \times_n M_1 = \mathcal{A} \times_n M_1 \times_p M_3$.
2. $(\mathcal{A} \times_n M_1) \times_n M_2 = \mathcal{A} \times_n (M_2 M_1)$.
3. *If M_1 is orthogonal, then* $\| \mathcal{A} \times_n M_1 \|_F = \| \mathcal{A} \|_F$.
4. *If $x \in \mathbb{R}^{I_n}$ and $J_n = I_n$ then* $(\mathcal{A} \times_n M_1) \bar{\times}_n x = \mathcal{A} \bar{\times}_n (M_1^T x)$.
5. *For $x \in \mathbb{R}^{I_n}$,* $\mathcal{A} \bar{\times}_n x = \mathcal{A} \times_n x^T$.

Definition 4 [16]. The Einstein product of two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times \ldots \times I_N \times J_1 \times \cdots \times J_M}$ and $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_M \times K_1 \times \cdots \times K_L}$, is the $I_1 \times \ldots \times I_N \times K_1 \times \cdots \times K_L$ tensor denoted by $\mathcal{A} *_M \mathcal{B}$ whose entries are given by

$$(\mathcal{A} *_M \mathcal{B})_{i_1,\ldots i_N k_1,\ldots k_M} = \sum_{j_1 \ldots j_M} \mathcal{A}_{i_1,\ldots i_N j_1 \ldots j_M} \mathcal{B}_{j_1 \ldots j_M k_1,\ldots k_L}$$

Definition 5 [16]. The tensor product denoted $\boxdot^{(N+1)}$ of two $(N+1)$-*mode* tensors $\mathcal{A} = [\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m] \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N \times m}$ and $\mathcal{B} = [\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n] \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N \times n}$, where $\mathcal{A}_i \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ and $\mathcal{B}_i \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$, is the $m \times n$ matrix whose (i, j) entries, $i = 1, \ldots, m$ and $j = 1, \ldots, n$, is given by

$$(\mathcal{A} \boxdot^{(N+1)} \mathcal{B})_{i,j} = \langle \mathcal{A}_i, \mathcal{B}_j \rangle.$$

Proposition 2 *Let* $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N \times I_{N+1}}$, $\mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ *and* $x \in \mathbb{R}^{I_{N+1}}$. *Then we have*

1. $(\mathcal{A} \boxdot^{(N+1)} \mathcal{B})^T = \mathcal{B} \boxdot^{(N+1)} \mathcal{A}$
2. $(\mathcal{A} \boxdot^{(N+1)} \mathcal{B})x = \mathcal{A} \boxdot^{(N+1)} (\mathcal{B} \bar{\times}_{(N+1)} x)$.
3. $\langle (\mathcal{A} \boxdot^{N+1} \mathcal{B})x, x \rangle = \langle \mathcal{B} \bar{\times}_{(N+1)} x, \mathcal{A} \bar{\times}_{(N+1)} x \rangle$.
4. $\langle \mathcal{C}, \mathcal{A} \bar{\times}_{(N+1)} x \rangle = (\mathcal{C} \boxdot^{(N+1)} \mathcal{A})x$

Proof By definitions of the involved tensor products. □

3 The HOSVD-TMPE method

First of all, let us recall the Tensor Global Minimal Extrapolation Method (TG-MPE) proposed in [16] that is the starting point of this work.
Let $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2 \ldots$ be a given sequence of real tensors of the same dimension $I_1 \times I_2 \times I_3 \times \cdots \times I_N$ and set

$$\mathcal{D}_n = \mathcal{S}_{n+1} - \mathcal{S}_n, \quad n = 0, 1, \ldots$$

Define for some fixed n and k, the $I_1 \times I_2 \times I_3 \times \cdots \times I_N \times I_{N+1}$ tensor $\mathbb{D}_k^{(n)}$ by

$$\mathbb{D}_k^{(n)} = \left[\mathcal{D}_n, \mathcal{D}_{n+1}, \ldots, \mathcal{D}_{n+k}\right] \text{ with } I_{N+1} = k + 1,$$

such that the i^{th} frontal slice, obtained by fixing the last index at i, is $\left[\mathbb{D}_k^{(n)}\right]_{:,:,\ldots,:,i} = \mathcal{D}_{n+i}$, $0 \leq i \leq k$.

The TG-MPE method is based on the solution, for $\bar{\alpha}^{(k)} = \left(\alpha_0^{(k)}, \alpha_1^{(k)}, \ldots, \alpha_{k-1}^{(k)}\right) \in \mathbb{R}^k$, of the problem

$$(\mathbb{D}_{k-1}^{(n)} \boxdot^{N+1} \mathbb{D}_{k-1}^{(n)})\bar{\alpha}^{(k)} = -(\mathbb{D}_{k-1}^{(n)} \boxdot^{N+1} \mathcal{D}_{n+k}), \tag{1}$$

for which the authors adopt a tensor QR based approach to obtain the solution (see [16] for details).
Once $\alpha_0^{(k)}, \alpha_1^{(k)}, \ldots, \alpha_{k-1}^{(k)}$ have been determined, set $\alpha_k^{(k)} = 1$. Provided that $\sum_{i=0}^{k} \alpha_i^{(k)} \neq 0$, compute the coefficients $\delta_0^{(k)}, \delta_1^{(k)}, \ldots, \delta_k^{(k)}$ as

$$\delta_j^{(k)} = \frac{\alpha_j^{(k)}}{\sum_{i=0}^{k} \alpha_i^{(k)}} \qquad for \quad j = 0, \ldots, k. \tag{2}$$

Finally we set

$$\mathcal{T}_k^{(n)} = \sum_{j=0}^{k} \delta_j^{(k)} \mathcal{S}_{n+j} \tag{3}$$

as an approximation to the limit of the sequence $(\mathcal{S}_n)$.
Now, let us outline the approach we followed in developing our proposed method. First, using assertion 1 of Proposition 2, we rewrite the problem in (1) as follows

$$\mathbb{D}_{k-1}^{(n)} \boxdot^{N+1} (\mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} \bar{\alpha}^{(k)} + \mathcal{D}_{n+k}) = 0. \tag{4}$$

The next proposition allows us to transform the above problem to an equivalent minimisation one.

Proposition 3 *The equation* (4) *is equivalent to the minimization least squares tensor problem*

$$\bar{\alpha}^{(k)} = \underset{x \in \mathbb{R}^k}{argmin} \parallel \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} x + \mathcal{D}_{n+k} \parallel_F \tag{5}$$

Proof We set $g(x) = \parallel \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} x + \mathcal{D}_{n+k} \parallel_F^2$. For a vector $h \in \mathbb{R}^k$ and a non zero scalar t, we have

$$\begin{aligned} g(x+th) &= \parallel \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} (x+th) + \mathcal{D}_{n+k} \parallel_F^2 \\ &= g(x) + 2t \langle \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} x + \mathcal{D}_{n+k}, \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} h \rangle \\ &\quad + t^2 \parallel \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} h \parallel_F^2 \\ &= g(x) + t \langle \mathbb{D}_{k-1}^{(n)} \boxdot^{N+1} (\mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} x + \mathcal{D}_{n+k}), h \rangle \\ &\quad + t^2 \parallel \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} h \parallel_F^2 \end{aligned}$$

Therefore, the gradient of g at x is as $\nabla g(x) = \mathbb{D}_{k-1}^{(n)} \boxdot^{N+1} (\mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} x + \mathcal{D}_{n+k})$. It is easy to check that the matrix $\mathbb{D}_{k-1}^{(n)} \boxdot^{N+1} \mathbb{D}_{k-1}^{(n)}$ is positive semi-definite, which guarantees the convexity of g. As a result

$$\bar{\alpha}^{(k)} = \underset{x \in \mathbb{R}^k}{\texttt{argming}}(x) \iff \nabla g(\bar{\alpha}^{(k)}) = 0.$$

□

Therefore, the problem in (4) is equivalent to

$$\min_{\bar{\alpha}^{(k)} \in \mathbb{R}^k} \| \mathbb{D}_{k-1}^{(n)} \bar{\times}_{(N+1)} \bar{\alpha}^{(k)} + \mathcal{D}_{n+k} \|_F, \tag{6}$$

which can be expressed as the following constrained problem

$$\min_{\alpha^{(k)} \in \mathbb{R}^{k+1}} \|\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} \alpha^{(k)}\|_F \quad \text{subject to } \alpha_k^{(k)} = 1, \ \alpha^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_k^{(k)})^T. \tag{7}$$

Rewriting the basic problem as a constrained minimization problem is indeed the core concept of this work. Specifically, for our proposed method (HOSVD-TMPE), we have replaced the constraint $\alpha_k^{(k)} = 1$ in (7) with the new constraint $\| \alpha^{(k)} \|_2 = 1$, resulting in the new constrained minimization problem

$$\begin{aligned} &\min_{\alpha^{(k)} \in \mathbb{R}^{k+1}} \| \mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} \alpha^{(k)} \|_F \quad \text{subject to } \| \alpha^{(k)} \|_2 = 1, \ \alpha^{(k)} \\ &= (\alpha_0^{(k)}, \alpha_1^{(k)}, \ldots, \alpha_k^{(k)})^T. \end{aligned} \tag{8}$$

Again, with $\alpha_0^{(k)}, \alpha_1^{(k)}, \ldots, \alpha_k^{(k)}$ determined and provided $\sum_{i=0}^k \alpha_i^{(k)} \neq 0$, we compute similarly the scalars $\delta_0^{(k)}, \delta_1^{(k)}, \ldots, \delta_k^{(k)}$ as in (2) and the new sequence term $\mathcal{T}_k^{(n)}$ as in (3).

Remark 1 The purpose of this new constraint choice is not to obtain a unit solution, as it may seem, since we can achieve that simply through the normalization of the result obtained by TG-MPE. Instead, as will be demonstrated, it enables us to easily characterize the desired approximate solution and develop an efficient algorithm for its determination.

The next theorem recalls the HOSVD decomposition [13], which is a highly valuable tool for establishing the main result in this work.

Theorem 4 (HOSVD) [13] *Let $\mathcal{A}$ be a real tensor of dimension $I_1 \times I_2 \times I_3 \times \cdots \times I_N$. Then $\mathcal{A}$ can be decomposed as*

$$\mathcal{A} = \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)}, \tag{9}$$

where $V^{(n)} = \left[v_1^{(n)} v_2^{(n)} \ldots v_{I_n}^{(n)} \right]$ is a unitary $I_n \times I_n$ matrix $(1 \leq n \leq N)$, Σ is a real tensor of the same dimension as $\mathcal{A}$ and the subtensors $\Sigma_{i_n=m}$ $(1 \leq m \leq I_n)$ obtained by fixing the n^{th} index to m, have the following properties

- *All-orthogonality $\langle \Sigma_{i_n=m}, \Sigma_{i_n=m'} \rangle = 0 \quad for \ \ m \neq m'$.*
- *$\| \Sigma_{i_n=1} \|_F \geqslant \| \Sigma_{i_n=2} \|_F \geqslant \cdots \| \Sigma_{i_n=I_n} \|_F \geqslant 0$ for all possible values of n.*

The quantity $\| \Sigma_{i_n=i} \|_F$ is called the i^{th} n-mode singular value of $\mathcal{A}$, it is symbolized by $\sigma_i^{(n)}$; and the vector $v_i^{(n)}$ is the i^{th} n-mode singular vector.

In the following main result, we provide a characterization of the solution $\alpha^{(k)}$ of the minimisation problem (8).

Theorem 5 *Let $\sigma_1^{(N+1)}, \sigma_2^{(N+1)}, \ldots, \sigma_{k+1}^{(N+1)}$ be the $(N+1)$-mode singular values of $\mathbb{D}_k^{(n)}$ ordered as in*

$$\sigma_1^{(N+1)} \geq \sigma_2^{(N+1)} \geq \ldots \geq \sigma_{k+1}^{(N+1)} \geqslant 0,$$

and let $\{v_i^{(N+1)}\}_{i=1}^{k+1}$ be the corresponding $(N+1)$-mode singular vectors obtained from the HOSVD,

$$\mathbb{D}_k^{(n)} = \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_{(N+1)} V^{(N+1)}. \tag{10}$$

If the smallest $(N+1)$-mode singular value $\sigma_{k+1}^{(N+1)}$ of $\mathbb{D}_k^{(n)}$, is simple, in the sense that $\sigma_k^{(N+1)} \neq \sigma_{k+1}^{(N+1)}$, then the solution $\alpha^{(k)}$ to the minimization problem (8) is unique (up to a multiplicative constant ρ, $\mid \rho \mid = 1$) and is given as $\alpha^{(k)} = v_{k+1}^{(N+1)}$.

Proof First, let us show that

$$\|\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} v_{k+1}^{(N+1)}\|_F = \sigma_{k+1}^{(N+1)}.$$

In fact, by Assertion 4 of Proposition 1, we have

$$\begin{aligned}\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} v_{k+1}^{(N+1)} &= \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \\ &\quad \times_{(N+1)} V^{(N+1)} \bar{\times}_{(N+1)} v_{k+1}^{(N+1)} \\ &= \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \\ &\quad \times_N V^{(N)} \bar{\times}_{(N+1)} ((V^{(N+1)})^T v_{k+1}^{(N+1)}).\end{aligned}$$

Since $V^{(N+1)}$ is unitary, one has $(V^{(N+1)})^T v_{k+1}^{(N+1)} = (0, 0, \ldots, 1)^T = e_{k+1} \in \mathbb{R}^{k+1}$. Therefore, using Assertion 1 and 5 of Proposition 1,

$$\begin{aligned}\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} v_{k+1}^{(N+1)} &= \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)} \bar{\times}_{(N+1)} e_{k+1} \\ &= (\Sigma \bar{\times}_{(N+1)} e_{k+1}) \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)} \\ &= \Sigma_{i_{(N+1)}=k+1} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)}.\end{aligned}$$

As a result, using Assertion 3 of Proposition 1, we get

$$\|\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} v_{k+1}^{(N+1)}\|_F = \|\Sigma_{i_{(N+1)}=k+1}\|_F = \sigma_{k+1}^{(N+1)}.$$

Now, let β be an arbitrary vector in $\mathbb{R}^{k+1}$ such that $\|\beta\|_2 = 1$. Since $V^{(N+1)} = \left[v_1^{(N+1)} v_2^{(N+1)} \ldots v_{I_N+1}^{(N+1)}\right]$ is a unitary matrix in $\mathbb{R}^{(k+1)\times(k+1)}$, then there exist scalars

$\beta_1, \ldots, \beta_{k+1}$ such that

$$\beta = \sum_{i=1}^{k+1} \beta_i v_i^{(N+1)} \quad \text{with} \quad \|\beta\|^2 = \sum_{i=1}^{k+1} (\beta_i)^2 = 1.$$

Taking into account the orthogonality of $\{\Sigma_i^{(N+1)}\}_{i=1,\ldots,k+1}$, and the fact that $\sigma_{k+1}^{(N+1)}$ is the smallest $(N+1)$-mode singular value, we can easily establish that

$$\begin{aligned}
\|\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} \beta\|_F^2 &= \| \sum_{j=1}^{k+1} \beta_j \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_{(N+1)} V^{(N+1)} \\
&\quad \bar{\times}_{(N+1)} v_j^{(N+1)} \|_F^2 \\
&= \| \sum_{j=1}^{k+1} \beta_j \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)} \bar{\times}_{(N+1)} e_j \|_F^2 \\
&= \| \sum_{j=1}^{k+1} \beta_j (\Sigma \bar{\times}_{(N+1)} e_j) \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)} \|_F^2 \\
&= \| \sum_{j=1}^{k+1} \beta_j \Sigma_{i_{(N+1)}=j} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)} \|_F^2 \\
&= \langle \sum_{j=1}^{k+1} \beta_j \Sigma_{i_{(N+1)}=j}, \sum_{j=1}^{k+1} \beta_j \Sigma_{i_{(N+1)}=j} \rangle \quad (\textit{Orthognality of} \quad V^{(n)}, s) \\
&= \sum_{j=1}^{k+1} \beta_j^2 \langle \Sigma_{i_{(N+1)}=j}, \Sigma_{i_{(N+1)}=j} \rangle \quad (\textit{Orthognality of} \quad \Sigma_{i_{(N+1)}=j}, s) \\
&= \sum_{j=1}^{k+1} \beta_j^2 (\sigma_j^{(N+1)})^2 \geq (\sigma_{k+1}^{(N+1)})^2 (\sum_{j=1}^{k+1} \beta_j^2) = (\sigma_{k+1}^{(N+1)})^2.
\end{aligned}$$

Therefore

$$\|\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} \beta\|_F \geq \sigma_{k+1}^{(N+1)},$$

which leads to

$$\min_{\|\alpha^{(k)}\|=1} \|\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} \alpha^{(k)}\|_F = \|\mathbb{D}_k^{(n)} \bar{\times}_{(N+1)} v_{k+1}^{(N+1)}\|_F = \sigma_{k+1}^{(N+1)}.$$

□

4 Implementation of HOSVD-TMPE

As demonstrated in the previous section, determining the sequence term $\mathcal{T}n^{(k)}$ in (3) requires the computation of $vk+1$, the $(k+1)^{th}$ vector of the $(N+1)$-mode matrix $V^{(N+1)}$ corresponding to the smallest $(N+1)$-mode singular value $\sigma_{k+1}^{(N+1)}$ from the decomposition (10). Obtaining the matrix $V^{(N+1)}$ directly through HOSVD applied to $\mathbb{D}k^{(n)}$ can be expensive. However, in this case, we employ a less computationally intensive method to obtain the matrix $V^{(N+1)}$ and consequently, the vector $vk+1^{(N+1)}$. To begin, let us introduce the following lemma, which was proven in [3].

Lemma 1 [3] *Let $\mathcal{A}$ be a real tensor of dimension $I_1 \times I_2 \times I_3 \times \cdots \times I_N$, and consider its HOSVD decomposition as*

$$\mathcal{A} = \Sigma \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \cdots \times_N V^{(N)}. \tag{11}$$

Then a matrix representation of (11) can be obtained, for $n \in \{1, \cdots, N\}$, as follows

$$\mathcal{A}_{(n)} = V^{(n)} \Sigma_{(n)} (V^{(n+1)} \otimes V^{(n+2)} \otimes \cdots \otimes V^{(N)} \otimes V^{(1)} \cdots \otimes V^{(n-1)})^T, \tag{12}$$

where $\otimes$ stands for the Kronecker product of matrices.

Proposition 6 *For each $n \in \{1, \ldots, N\}$ we have:*

$$\mathcal{A}_{(n)} \mathcal{A}_{(n)}^T = V^{(n)} \Sigma_{(n)} \Sigma_{(n)}^T (V^{(n)})^T \tag{13}$$

Proof We use (12) and the orthogonality of the matrices $V^{(n)}$, $1 \leq n \leq N$. □

The decomposition (13) is, in fact, the spectral decomposition of the symmetric matrix $\mathcal{A}_{(n)}\mathcal{A}_{(n)}^T$. Therefore, to get the n-mode matrix $V^{(n)}$ appearing the HOSVD-decomposition of the tensor $\mathcal{A}$, we can compute it throughout the eigenvalue decomposition (EVD) of the $I_n \times I_n$ matrix $\mathcal{A}_{(n)}\mathcal{A}_{(n)}^T$.

Now, we adopt this process to determine the matrix $V^{(N+1)}$ in (10). We first compute the unfolding matrix $(\mathbb{D}_k^{(n)})_{(N+1)}$ and the matrix $M = (\mathbb{D}_k^{(n)})_{(N+1)}(\mathbb{D}_k^{(n)})_{(N+1)}^T$ of size $(k+1) \times (k+1)$. Therefore, applying the EVD decomposition on M allows us to obtain the matrix $V^{(N+1)}$ (we have $M = V^{(N+1)}\Sigma(V^{(N+1)})^T$). Then, the $(k+1)^{th}$ vector (solution $\alpha^{(k)}$) $v_{k+1} = V^{(N+1)}e_{k+1}$ where $e_{k+1} = (0, 0, \ldots, 1)^T \in \mathbb{R}^{k+1}$.

Remark 2 In practice, the integer k is not typically large. Therefore, applying the Eigenvalue Decomposition EVD to the symmetric positive semi definite $(k+1) \times (k+1)$ matrix $M = (\mathbb{D}_k^{(n)})_{(N+1)}(\mathbb{D}_k^{(n)})_{(N+1)}^T$ is less computationally expensive compared to performing the HOSVD decomposition of the tensor $\mathbb{D}_k^{(n)}$.

After obtaining the solution $\alpha^{(k)}$, we used (2) and (3) for computing the extrapolated tensor $\mathcal{T}_n^{(k)}$. We summarize the different steps of the implementation of HOSVD-TMPE in Algorithm 1.

Algorithm 1 HOSVD-TMPE

Input: n and k and $\mathcal{S}_n, \mathcal{S}_{n+1}, \ldots, \mathcal{S}_{n+k+1}$.
Output:. $\mathcal{T}_n^{(k)}$
Compute the tensors $\mathcal{D}_n, \mathcal{D}_{n+1}, \ldots, \mathcal{D}_{n+k}$ and form the tensor $\mathbb{D}_k^{(n)} = \left[\mathcal{D}_n, \mathcal{D}_{n+1}, \ldots, \mathcal{D}_{n+k}\right]$.
Compute the unfolding matrix $(\mathbb{D}_k^{(n)})_{(N+1)}$.
Compute the square matrix $M := (\mathbb{D}_k^{(n)})_{(N+1)}((\mathbb{D}_k^{(n)})_{(N+1)})^T \in \mathbb{R}^{(k+1)\times(k+1)}$.
Compute the EVD of M : $M = V\Sigma V^T$.
Determine $\alpha^{(k)} = Ve_{k+1}$ with $e_{k+1} = (0, 0, \ldots, 1)^T \in \mathbb{R}^{k+1}$.
Determine $\delta^{(k)} = (\delta_0^{(k)}, \delta_1^{(k)}, ..., \delta_k^{(k)})$ via (2).
Set $\mathcal{T}_n^{(k)} = \sum_{i=0}^{k} \delta_i^{(k)} \mathcal{S}_{n+i}$.

It is clear that Algorithm 1 outlines only the computation of a single extrapolated tensor $\mathcal{T}_n^{(k)}$ (with fixed values for n and k), and does not provide any information about the sequence $(\mathcal{T}_n^{(k)})_n$ as n varies. To address this aspect, in the experimental section of this work, we employ two distinct schemes following the manner in which the sequence is generated.

Algorithm 2 HOSVD-TMPE for fixed-point problems

Input: $iter = 0, iter_{max}, \epsilon$, k and $\mathcal{S}_n$.
Output:. $\mathcal{T}_n^{(k)}$
for $j = 1 : k + 1$ **do**
  $\mathcal{S}_{n+j} = \mathcal{F}(\mathcal{S}_{n+j-1})$
end for
With $\mathcal{S}_n, \mathcal{S}_{n+1}, \ldots, \mathcal{S}_{n+k+1}$ available, compute $\mathcal{T}_n^{(k)}$ by Algorithm 1.
if $\texttt{error}(\mathcal{T}_n^{(k)}) < \epsilon$ or $iter > iter_{max}$ **then** stop
else
  set $\mathcal{S}_n = \mathcal{F}(\mathcal{T}_n^{(k)})$, $iter = iter + 1$ and go to Step 3.
end if

In case where the sequence $(\mathcal{S}_n)$ is produced by a some fixed-point process

$$\mathcal{S}_{n+1} = \mathcal{F}(\mathcal{S}_n), \tag{14}$$

we apply our proposed algorithm 1 in the restarted acceleration scheme explained in [10, 11]. That is, starting from an iterate $\mathcal{S}_n$, we generate via the fixed-point process 14 a sequence of iterates $\{\mathcal{S}_n\}_{i=0}^{k+1}$ and then use Algorithm 1 to produce $\mathcal{T}_n^{(k)}$. Next, we use $\mathcal{F}(\mathcal{T}_n^{(k)})$ as the starting point of (14) and the process continues. In the absence of a specific convergence measurement tool, it is generally acceptable to stop computations after a fixed number of iterations $iter_{max}$ or when the relative residual error

$$\texttt{res}(\mathcal{T}_n^{(k)}) := \frac{\| \mathcal{T}_n^{(k)} - \mathcal{T}_{n-1}^{(k)} \|_F}{\| \mathcal{T}_n^{(k)} \|_F} \tag{15}$$

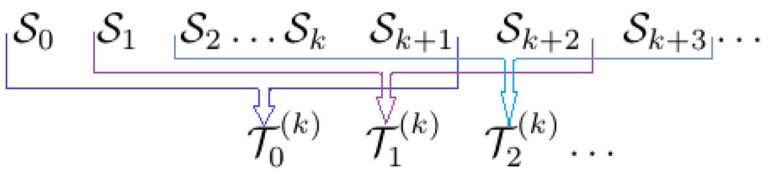

Fig. 1 The acceleration scheme

is less than a given threshold ϵ.

We summarized that in Algorithm 2. The `error` in Step 7 can either represent the relative residual `res` in (15) or some specific error depending on the studied problem.

Now, in cases where the sequence is not of a fixed-point type or when we lack information about its production process, we can apply our method in the acceleration scheme (see Fig. 1) as follows: we apply the extrapolation on the first $k+2$ iterates to produce the first term $\mathcal{T}_0^{(k)}$. Next, we start from the second iterate $\mathcal{S}_1$ and consider a set of $k+2$ iterates to produce the second term $\mathcal{T}_1^{(k)}$ via extrapolation and this process continues. The following figure illustrates this acceleration scheme.

Now, through the following result, we reveal a class of sequences on which the application of HOSVD-TMPE produces, just from a certain finite number of terms and in one time, the sought limit $\mathcal{S}$. Such property is known as finite termination property, see [27].

Theorem 7 *Let $\mathcal{S}$ be the limit of the tensor sequence $(\mathcal{S}_n)$ and let $\mathcal{E}_n = \mathcal{S}_n - \mathcal{S}$ and $\mathcal{D}_n = \mathcal{S}_{n+1} - \mathcal{S}_n$. Assume that there exists a fixed number k such that, for all $n = 0, 1, \ldots$*

$$\sum_{i=0}^{k} \alpha_i^{(k)} \mathcal{E}_{n+i} = 0 \qquad with \qquad \sum_{i=0}^{k} \alpha_i^{(k)} \neq 0 \quad and \quad \alpha_0^{(k)} \alpha_k^{(k)} \neq 0, \tag{16}$$

and the tensors $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{k-1}\}$ are linearly independent. Then the tensor $\mathcal{T}_k^{(n)}$ obtained by applying HOSVD-TMPE to the $k+2$ tensors $\mathcal{S}_n, \mathcal{S}_{n+1}, \ldots, \mathcal{S}_{n+k+1}$, satisfies $\mathcal{T}_k^{(n)} = \mathcal{S}$ for all $n = 0, 1, \ldots$.

Proof Let us first show by induction on n the linear independence of $\mathcal{D}_n, \mathcal{D}_{n+1}, \ldots, \mathcal{D}_{n+k-1}$. For $n = 0$, it is true by the hypothesis. Assume now that that $\mathcal{D}_n, \mathcal{D}_{n+1}, \ldots, \mathcal{D}_{n+k-1}$ are linearly independent, and show that so is for $\mathcal{D}_{n+1}, \mathcal{D}_{n+2}, \ldots, \mathcal{D}_{n+k}$. From (16), we get

$$\sum_{i=0}^{k} \alpha_i^{(k)} \mathcal{D}_{n+i} = 0, \qquad n = 1, 2, \ldots. \tag{17}$$

Since $\alpha_k^{(k)} \neq 0$, let us set $\alpha_k^{(k)} = -1$. Then (17) becomes

$$\mathcal{D}_{n+k} = \sum_{i=0}^{k-1} \alpha_i^{(k)} \mathcal{D}_{n+i} \tag{18}$$

To show that $\mathcal{D}_{n+1}, \mathcal{D}_{n+2}, \ldots, \mathcal{D}_{n+k}$ are independent, assume that there exist scalars $\beta_1, \beta_2, \ldots, \beta_k$ such that

$$\sum_{i=1}^{k} \beta_i \mathcal{D}_{n+i} = 0. \tag{19}$$

Substituting (18) into (19) and rearranging terms leads to

$$\beta_k \alpha_0^{(k)} \mathcal{D}_n + \sum_{i=1}^{k-1} (\beta_i + \beta_k \alpha_i^{(k)}) \mathcal{D}_{n+i} = 0. \tag{20}$$

The induction hypothesis implies that

$$\beta_k \alpha_0^{(k)} = 0 \quad \text{and} \quad \beta_i + \beta_k \alpha_i^{(k)} = 0 \quad \text{for} \quad i = 1, \ldots, k-1.$$

Since $\alpha_0^{(k)} \neq 0$, it follows that

$$\beta_1 = \beta_2 = \ldots = \beta_k = 0$$

which implies the linear independence of $\mathcal{D}_{n+1}, \mathcal{D}_{n+2}, \ldots, \mathcal{D}_{n+k}$.
On the other hand, using (16), we obtain

$$\sum_{i=0}^{k} \alpha_i^{(k)} (\mathcal{S}_{n+i} - \mathcal{S}) = \sum_{i=0}^{k} \alpha_i^{(k)} \mathcal{S}_{n+i} - (\sum_{i=0}^{k} \alpha_i^{(k)}) \mathcal{S} = 0,$$

from which it follows that

$$\mathcal{S} = \frac{\sum_{i=0}^{k} \alpha_i^{(k)} \mathcal{S}_{n+i}}{\sum_{i=0}^{k} \alpha_i^{(k)}}.$$

Therefore, setting $\check{\delta}_i^{(k)} = \dfrac{\alpha_i^{(k)}}{\sum_{i=0}^{k} \alpha_i^{(k)}}$, gives

$$\mathcal{S} = \sum_{i=0}^{k} \check{\delta}_i^{(k)} \mathcal{S}_{n+i}, \text{ with } \sum_{i=0}^{k} \check{\delta}_i^{(k)} = 1. \tag{21}$$

Again from (16), we have

$$\mathbb{D}_k^{(n)} \times_{(N+1)} \alpha^{(k)} = \sum_{i=0}^{k} \alpha_i^{(k)} \mathcal{D}_{n+i} = \sum_{i=0}^{k} \alpha_i^{(k)} \mathcal{E}_{n+i+1} - \sum_{i=0}^{k} \alpha_i^{(k)} \mathcal{E}_{n+i} = 0.$$

The linear independence of $\{\mathcal{D}_n, \mathcal{D}_{n+1}, \ldots, \mathcal{D}_{n+k-1}\}$ implies that the problem $\mathbb{D}_k^{(n)} \times_{(N+1)} \alpha^{(k)} = 0$ along with $\|\alpha^{(k)}\| = 1$ has a unique solution (up to a multiplicative constant ρ, $|\ \rho\ | = 1$). That is, the scalars $\check{\delta}_i^{(k)}$ in (21) are the same as $\delta_i^{(k)}$ in $\mathcal{T}_n^{(k)} = \sum_{i=0}^{k} \delta_i^{(k)} \mathcal{S}_{n+i}$ resulting from HOSVD-TMPE, and as a result $\mathcal{T}_n^{(k)} = \mathcal{S}$. □

5 Error analysis

In this section, we consider a convergent tensor sequence $(\mathcal{S}_n)$ with limit $\mathcal{S}$ obtained via the fixed-point process (14):

$$\mathcal{S}_{n+1} = \mathcal{F}(\mathcal{S}_n),$$

where $\mathcal{S}_0$ is a given tensor and $\mathcal{F}$ is assumed to be a differentiable function from $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. We associate with the iterate $\mathcal{S}_n$, the residual error given as:

$$\mathcal{R}(\mathcal{S}_n) = \mathcal{F}(\mathcal{S}_n) - \mathcal{S}_n = \mathcal{S}_{n+1} - \mathcal{S}_n. \tag{22}$$

The limit $\mathcal{S}$ satisfies the equations $\mathcal{S} = \mathcal{F}(\mathcal{S})$ and $\mathcal{R}(\mathcal{S}) = 0$.
As examples of $\mathcal{F}$, we consider the following cases:

1. $\mathcal{F}$ is linear:
   - $\mathcal{F}(\mathcal{X}) = \mathcal{X} \times_N A + \mathcal{B}$, with $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $A \in \mathbb{R}^{I_N \times I_N}$ such that $(I - A)$ is non-singular.
   - $\mathcal{F}(\mathcal{X}) = \mathcal{A} *_N \mathcal{X} + \mathcal{B}$ with $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times I_1 \times I_2 \times \cdots \times I_N}$ and $*_N$ stands for the Einstein product (Definition 4).
2. $\mathcal{F}$ is not linear: $\mathcal{F}(\mathcal{X}) = \mathcal{X}^T *_N \mathcal{A} *_N \mathcal{X} + \mathcal{X}^T *_N \mathcal{C}$ with $\mathcal{A}, \mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times I_1 \times I_2 \times \cdots \times I_N}$. Notice that in this case, one has

$$\mathcal{F}(\mathcal{X}) = \frac{\partial \mathcal{F}}{\partial \mathcal{X}}(\mathcal{S}) *_N (\mathcal{X} - \mathcal{S}) + \mathcal{F}(\mathcal{S}) + \mathbf{O}(\| \mathcal{X} - \mathcal{S} \|_F^2) \quad \text{as} \quad \mathcal{X} \longrightarrow \mathcal{S}$$

where the tensor $\dfrac{\partial \mathcal{F}}{\partial \mathcal{X}}(\mathcal{S}) = \mathcal{A} *_N \mathcal{S} - \mathcal{C}$ is the gradient of $\mathcal{F}$ at $\mathcal{S}$. Thus for a large n, the sequence $(\mathcal{S}_n)$ can be generated as follows

$$\mathcal{S}_{n+1} = \mathcal{F}(\mathcal{S}_n) \approx \frac{\partial \mathcal{F}}{\partial \mathcal{X}}(\mathcal{S}) *_N (\mathcal{S}_n - \mathcal{S}) + \mathcal{F}(\mathcal{S}). \tag{23}$$

That is, the sequence $(\mathcal{S}_n)$ behaves as if it were being generated by approximate linear system of the form $(\mathcal{I} - \mathcal{P}) *_N \mathcal{X} \approx \mathcal{B}$ through

$$\mathcal{X}_{n+1} = \mathcal{P} *_N \mathcal{X}_n + \mathcal{B}$$

with $\mathcal{P} = \dfrac{\partial \mathcal{F}}{\partial \mathcal{X}}(\mathcal{S})$ and $\mathcal{B} = (\mathcal{I} - \mathcal{P}) *_N \mathcal{S}$.

Coming back to the general case, assume that $\mathcal{F}$ has a unique fixed point denoted by $\mathcal{S}$. Applying HOSVD-TMPE on $(\mathcal{S}_n)$, the residual $\mathcal{R}(\mathcal{T}_n^{(k)}) = \mathcal{T}_{n+1}^{(k)} - \mathcal{T}_n^{(k)}$ can be expressed as

$$\mathcal{R}(\mathcal{T}_n^{(k)}) \cong \sum_{i=0}^{k} \delta_i^{(k)} \mathcal{D}_{n+i} = \mathbb{D}_k^{(n)} \times_{(N+1)} \delta^{(k)}, \text{ with } \delta^{(k)} = \left(\delta_0^{(k)}, \delta_2^{(k)}, \ldots, \delta_k^{(k)}\right)^T, \tag{24}$$

where '$\cong$' represents exact equality '$=$' when the sequence $(\mathcal{S}_n)$ is generated linearly, while, it is just an approximation '$\approx$' when $\mathcal{F}$ is non-linear.

Assume that $(\mathcal{S}_n)$ is generated linearly, the subsequent theorem demonstrates that the computation of $\|\mathcal{R}(\mathcal{T}_n^{(k)})\|_F$ can be achieved without incurring any additional computational expense, relying solely on the quantities derived from our algorithm, and without the need to directly compute $\mathcal{R}(\mathcal{T}_n^{(k)})$. The result remains true for the nonlinear case when replacing $=$ with $\approx$.

Theorem 8 *The residual $\mathcal{R}(\mathcal{T}_n^{(k)})$ in (24) satisfies*

$$\|\mathcal{R}(\mathcal{T}_n^{(k)})\|_F = \frac{\sigma_{k+1}^{(N+1)}}{|\sum_{i=0}^{k} \alpha_i^{(k)}|}, \tag{25}$$

where $\sigma_{k+1}^{(N+1)}$ is the smallest $(N+1)$-mode singular value of $\mathbb{D}_k^{(n)}$.

Proof Using the relation (2), it follows that

$$\mathbb{D}_k^{(n)} \times_{(N+1)} \delta^{(k)} = \frac{(\mathbb{D}_k^{(n)} \times_{(N+1)} \alpha^{(k)})}{\sum_{i=0}^{k} \alpha_k^{(k)}}$$

with $\alpha^{(k)} = \left(\alpha_0^{(k)}, \alpha_2^{(k)}, \ldots, \alpha_k^{(k)}\right)^T$ is the solution of (8). Then, using Theorem 5, we obtain

$$\|\mathcal{R}(\mathcal{T}_n^{(k)})\|_F = \frac{1}{|\sum_{i=0}^{k} \alpha_k^{(k)}|} \|\mathbb{D}_k^{(n)} \times_{(N+1)} \alpha^{(k)}\|_F = \frac{\sigma_{k+1}^{(N+1)}}{|\sum_{i=0}^{k} \alpha_k^{(k)}|}.$$

□

6 Numerical experiments

In this section, we will illustrate, via some numerical experiments, the effectiveness of our proposed method (HOSVD-TMPE). We compared it in the examples below, in addition to the basic sequences (Basic-Iterations), with the TG-MPE method and the Nesterov's transformation [23] defined by

$$\mathcal{T}_{n+1}^{Nv} = \mathcal{X}_{n+1} + \frac{\eta_n + 1}{\eta_{n+1}}(\mathcal{X}_{n+1} - \mathcal{X}_n),$$

where $\{\eta_n\}$ is the scalar sequence given by $\eta_{n+1} = \frac{\sqrt{4(\eta_n)^2 + 1) + 1}}{2}$ with $\eta_1 = 1$. For Examples 1, 2, and 4, we implemented the HOSVD-TMPE and TG-MPE methods using the restarted acceleration scheme outlined in Algorithm 2. Since the exact solution is known, we employed the relative error

$$\in(\mathcal{T}_n^{(k)}) = \frac{\| \mathcal{T}_n^{(k)} - \mathcal{S} \|_F}{\| \mathcal{S} \|_F}, \tag{26}$$

to illustrate the behaviour of the produced sequences.
For Example 3, we adopted the acceleration scheme outlined in Fig. 1, utilizing the residual error-norm

$$\mathcal{R}e(\mathcal{T}_n^{(k)}) = \| \mathcal{B} - \mathcal{A} *_2 \mathcal{T}_n^{(k)} \|, \tag{27}$$

to assess the convergence behaviour of the methods.

In addition, in Example 4, which addresses image restoration problems, we used the PSNR (Peak Signal-to-Noise Ratio) to assess the quality of the restored images. Furthermore, in this specific example, we evaluated the convergence rates of TG-MPE and HOSVD-TMPE by considering the ratio $\frac{\|\mathcal{T}_n^{method} - \mathcal{S}\|_F}{\|\mathcal{S}_n - \mathcal{S}\|_F}$.

As shown in Algorithm 2, the iteration process is terminated as soon as the norm of the used error is less than predefined ϵ, or the number of iterations reaches a given number $iter_{max}$. We carried out the computations using MATLAB 2023a in HP computer running Windows 11 with Intel Core i7 and 16 GB RAM.

Example 1

We consider tensor sequence $(\mathcal{S}_n)$ with $\mathcal{S}_n \in \mathbb{R}^{d \times d \times d}$, obtained from the linear iterative scheme

$$\mathcal{S}_{n+1} = \mathcal{S}_n \times_3 A + \mathcal{B}, \qquad n = 0, 1, \ldots$$

$A = V \Sigma V^T \in \mathbb{R}^{d \times d}$ where V is a random orthogonal matrix, and Σ is a diagonal matrix such that $\Sigma(i, i) = 1 - 0.95 \times \frac{i}{d}$, $i = 1, .., d$. The tensor $\mathcal{B}$ is such that the

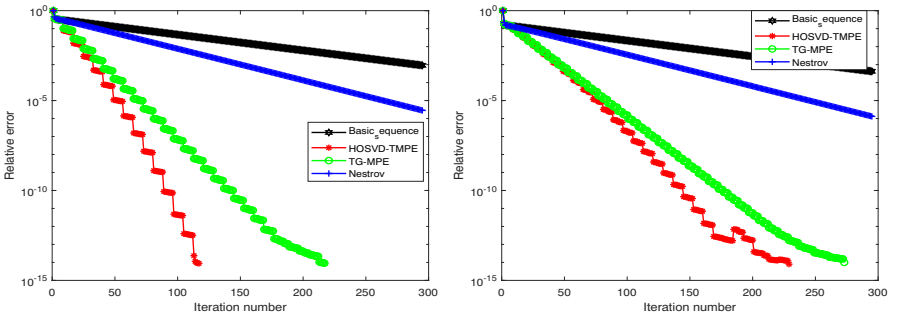

Fig. 2 The relative error for dimensions $d = 25$ (left) and $d = 40$ (right) for the Basic-iteration, Nesterov, TG-MPE and HOSVD-TMPE (Example 1)

limit (exact solution) is the tensor $\mathcal{S}$ with all elements equal to one:

$$\mathcal{B} = \mathcal{S} \times_3 (I - A).$$

Fig. 2 illustrates the relative error-norm versus the iteration index n for both the Basic iteration and Nesterov process, TG-MPE, and our proposed method HOSVD-TMPE with $k = 3$, considering different dimensions ($d = 25$ and 40). We set ϵ to 10^{-14}. The curves clearly reveal the acceleration effect of our method, showcasing its superiority compared to TG-MPE and Nesterov.

Example 2

In this example, we consider the tensor sequence $(\mathcal{S}_n)$ from $\mathbb{R}^{N\times N\times N}$ obtained by the following linear iterative scheme

$$\mathcal{S}_{n+1} = \mathcal{A} *_N \mathcal{S}_n + \mathcal{B}, \qquad n = 0, 1, \ldots$$

with $\mathcal{A} \in \mathbb{R}^{N\times N\times N\times N\times N\times N}$ such that

$$\mathcal{A}_{i,j,k,i,j,k} = (i + j + k)/(3 * N) \quad \textit{and} \quad \mathcal{A}_{i,i,i,i,i,i} = 1 - 0.01 * i.$$

The tensor right-hand $\mathcal{B}$ is such that the limit $\mathcal{S}$ is the tensor whose all elements equal to one, and is given by

$$\mathcal{B} = (I - \mathcal{A}) *_N \mathcal{S}.$$

Fig. 3 illustrates the behavior of the relative error-norm for the basic sequence, the Nesterov process, TG-MPE, and our method HOSVD-TMPE ($k = 3$) across different dimensions ($N = 10$ and $N = 15$). We set ϵ to 10^{-13} as a stopping criterion. It is evident that the application of TG-MPE and HOSVD-TMPE results in much faster convergence compared to Basic Iterations and the Nesterov process. As observed in the

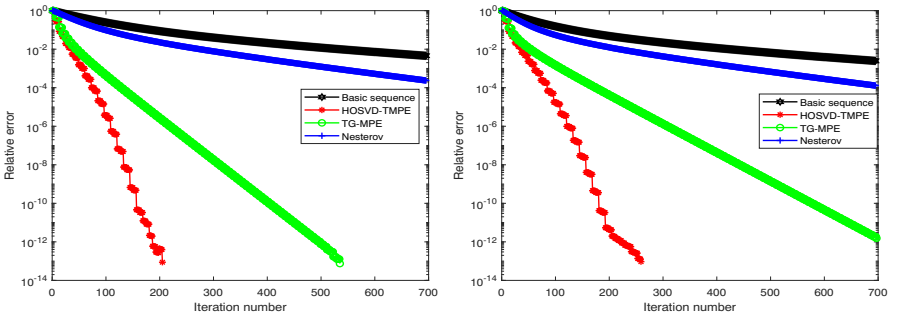

Fig. 3 The relative error-norm for dimensions $N = 10$ (left) and $N = 15$ (right) for the Basic-iteration, TG-MPE, Nesterov and HOSVD-TMPE (Example 2)

previous example, the HOSVD-TMPE method the HOSVD-TMPE method exhibits rapid convergence.

Example 3

In this example, we consider the partial differential equation

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0 & \Omega =]0,1[\times]0,1[\,, \\ u(0,y) = 0, \quad u(1,y) = \sin 3\pi y, \\ u(x,0) = 0, \quad u(x,1) = \sin 3\pi x \cos \pi x. \end{cases} \tag{28}$$

The central finite difference discretization of (28) leads to the following tensor equation with Einstein product (see Definition 4)

$$\mathcal{A} *_2 \mathcal{X} = \mathcal{B}, \qquad \mathcal{X} = (x_{kl}) \in \mathbb{R}^{N \times N}, \tag{29}$$

with $\mathcal{A} = (\alpha_{ijkl}) \in \mathbb{R}^{N \times N \times N \times N}$ and $\mathcal{B} = (\beta_{ij}) \in \mathbb{R}^{N \times N}$ such that

$$\begin{aligned} &\alpha_{ijkl} = q_{mn}, \quad m = ivec([i,j],[N,N]), \quad n = ivec([k,l],[N,N]), \\ &\beta_{ij} = b_t, \quad t = ivec([i,j],[N,N]) \text{ and } Q = [q_{mn}] := (F \otimes I_N + I_N \otimes E), \end{aligned}$$

where $\otimes$ denotes the Kronecker product, $ivec$ is the index mapping function defined as follows: For given integers j_1, j_2 and dimensions N_1, N_2, the integer $ivec([i,j],[N_1,N_2])$ is given by

$$ivec([j_1,j_2],[N_1,N_2]) = j_1 + (j_2 - 1)N_1 N_2.$$

$F = tridiag(1, -4, 1)$ and $E = [e_{ij}]$ are tridiagonal and diagonal matrices of order N and $b = (b_i) \in \mathbb{R}^{N^2}$, defined by

$$e_{ij} = \begin{cases} 1, & \text{if } \mid i - j \mid = 1, \\ 0, & \text{otherwise,} \end{cases}$$

$$b_t = \begin{cases} -\sin\dfrac{3t}{N+1}\pi & \text{if } t = vN, \quad v = 1, \ldots, N, \\ -\sin\dfrac{3t}{N+1}\pi \cos\dfrac{t}{N+1}\pi & \text{if } t = (N-1)N + w, \quad w = 1, \ldots, N-1, \\ -\sin\dfrac{3N}{N+1}\pi(1 + \cos\dfrac{N}{N+1}\pi) & \text{if } t = N^2, \\ 0 & \text{otherwise.} \end{cases}$$

We solve (29) using the following iterative scheme [29]:

$$\begin{cases} \mathcal{X}_{n+1} = \mathcal{X}_n + \dfrac{Tr(\mathcal{P}_n^T *_2 \mathcal{R}_n)}{Tr(\mathcal{P}_n^T *_2 \mathcal{A} *_2 \mathcal{P}_n)}\mathcal{P}_n, \\ \mathcal{R}_{n+1} = \mathcal{B} - \mathcal{A} *_2 \mathcal{X}_{n+1}, \\ \mathcal{P}_{n+1} = \mathcal{R}_{n+1} - \dfrac{Tr(\mathcal{P}_n^T *_2 \mathcal{A} *_2 \mathcal{R}_{n+1})}{Tr(\mathcal{P}_n^T *_2 \mathcal{A} *_2 \mathcal{P}_n)}\mathcal{P}_n. \end{cases} \tag{30}$$

with $\mathcal{P}_0 = \mathcal{R}_0 = \mathcal{B} - \mathcal{A} *_2 \mathcal{X}_0$ where $\mathcal{X}_0$ is an initial tensor guess.

In Table 1, we present the iteration number n, the residual error-norm $\mathcal{R}e(\mathcal{T}_n^{(k)}) = \| \mathcal{B} - \mathcal{A} *_2 \mathcal{T}_n^{(k)} \|_F$, and the CPU time for the four methods for different dimensions $N = 10, 40$ and 100. Note that the stopping criterion ϵ is adjusted according to the dimension N ($\epsilon = 10^{-16}$ for $N = 10$, $\epsilon = 10^{-8}$ for $N = 40$ and $\epsilon = 10^{-2}$ for $N = 100$). The results highlight the acceleration effect of our proposed method (HOSVD-TMPE). Across various dimensions ($N = 10$, 40, and 100), both the number of iterations and the associated CPU time for HOSVD-TMPE and TG-MPE are lower compared to those obtained with the initial sequence (Basic Iteration) and the Nesterov process. We can also note the effectiveness of HOSVD-TMPE in comparison to TG-MPE method, particularly when considering the case of $N = 100$.

In Fig. 4, we have plotted the residual error $\mathcal{R}e$ versus the iteration number n for four methods: Basic Iterations, Nesterov, TG-MPE, and HOSVD-TMPE, considering two different values of dimensions, $N = 30$ (left) and $N = 50$ (right). Iterations are terminated when the residual $\mathcal{R}e$ reaches 10^{-8}. The acceleration effect of HOSVD-TMPE is evident, showcasing its competitiveness with TG-MPE for both $N = 30$ and $N = 50$. They exhibit nearly similar behaviors, with a slight advantage observed for HOSVD-TMPE.

Table 1 Comparison of the required iteration number, residual error $\mathcal{R}e$, and CPU time for Basic Iterations (Scheme (30)), Nesterov, TG-MPE, and HOSVD-TMPE (Example 3)

Dimensions	Algorithms	Iter	Error $\mathcal{R}e$	CPU-time (s)
$N = 10$	scheme (30) (Basic-Iteration)	1575	9.78e−17	2.27
	Nesterov	3249	9.83e−17	2.45
	TG-MPE ($k = 4$)	644	9.53e−17	1.04
	HOSVD-TMPE ($k = 4$)	646	7.95e−17	1.01
N=40	scheme (30)	5040	9.96e−09	14.61
	Nesterov	9993	9.98e−09	28.83
	TG-MPE ($k = 4$)	1199	9.99e−09	4.87
	HOSVD-TMPE ($k = 4$)	1200	9.77e−09	5.11
N=100	scheme (30)	7873	9.95e−03	402.02
	Nesterov	19,123	9.96e−03	733.06
	TG-MPE ($k = 4$)	1096	9.92e−03	85.06
	HOSVD-TMPE ($k = 4$)	1095	9.92e−03	83.02

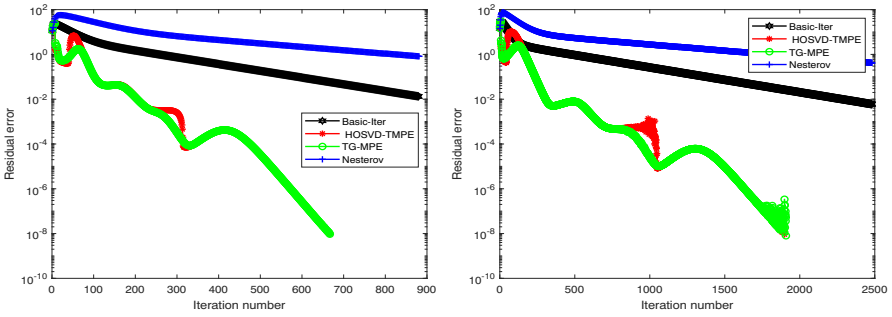

Fig. 4 The residual error $\mathcal{R}e$ for dimensions $N = 30$ (left) and $N = 50$ (right) for the Basic-iteration (30), Nesterov, TG-MPE and HOSVD-TMPE (Example 3)

Example 4

Image inpainting is the process of reconstructing missing regions in an image. This task is crucial in computer vision and serves as an essential functionality in numerous imaging and graphics applications; for more details, refer to [28].
The tensor total variation is one of the techniques employed for this purpose. In this example, we address the regularization tensor-least squares problem

$$\min_{\mathcal{X}}\{\|\mathcal{P}_{\mathsf{E}}(\mathcal{X}) - \mathcal{B}\|_F + \lambda\|\nabla\mathcal{X}\|_1\} \tag{31}$$

Fig. 5 The original 'coloredChips.png' image (left) and its text-added one (right)

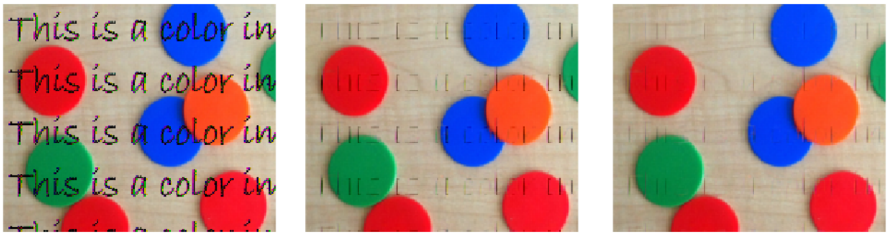

Fig. 6 The recovered images by TDPG (left), TG-MPE (middle) and HOSVD-TMPE (right) in 500 iterations

where $\mathcal{P}_{\mathsf{E}}$ is the tensor projection on $\mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$, such that for $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ the entries of $\mathcal{P}_{\mathsf{E}}(\mathcal{X})$ are given as

$$(\mathcal{P}_{\mathsf{E}}(\mathcal{X}))_{i_1,i_2,\ldots,i_N} = \begin{cases} \mathcal{X}_{i_1,i_2,\ldots,i_N} & \text{if } (i_1, i_2, \ldots, i_N) \in \mathsf{E}, \\ 0 & \text{otherwise,} \end{cases}$$

with $\mathsf{E} = \{(i_1, i_2, \ldots, i_N) : \mathcal{X}_{i_1,i_2,\ldots,i_N} \text{ observed } \}$ (see [8]).
Let $\mathcal{X}^{exact}$ be the original image, and the tensor $\mathcal{B}$ represents its text-added version. Utilizing the tensorial double proximal gradient algorithm (TDPG) proposed in [4], we iteratively solve the problem (31).

For the first experiment, we consider the image 'coloredChips.png' from the MATLAB R2020a collection, with dimensions of $(150 \times 190 \times 3)$. We corrupted it by adding text, as shown in Fig. 5.
Fig. 6 displays the restored images using TDPG, TG-MPE, and the proposed method (HOSVD-TMPE). It is evident that our proposed method successfully removed nearly all the added text, a result not achieved by the basic iteration TDPG or the TG-MPE method.

Table 2 presents the iteration number, relative error-norm, recovered image PSNR, and the required CPU time for TDPG (Basic Iterations), TG-MPE, and HOSVD-TMPE. We use $\epsilon = 5 \times 10^{-2}$ as a stopping criterion for computations. The results obtained confirm the superiority of our proposed method in terms of PSNR and relative error-norm, as well as in terms of CPU time.

Table 2 Comparison of the required iterations number, relative error, PSNR and CPU time for TDPG (Basic-Iterations), TG-MPE and HOSVD-TMPE for the 'coloredChips' image

Data size	Algorithms	Iter	Error	PSNR	CPU-time (s)
$150 \times 190 \times 3$	TDPG (Basic-iterations)	1771	4.99e−02	30.27	21.44
	TG-MPE ($k = 4$)	733	4.96e−02	30.30	10.06
	HOSVD-TMPE ($k = 4$)	541	4.98e−02	31.64	7.22

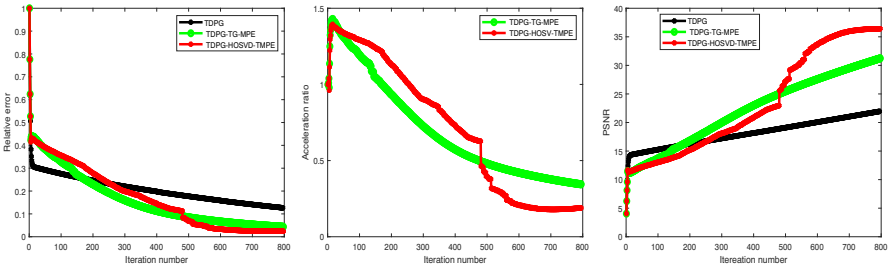

Fig. 7 The relative error-norm, the acceleration rate and the PSNR curves for 'coloredChips' image

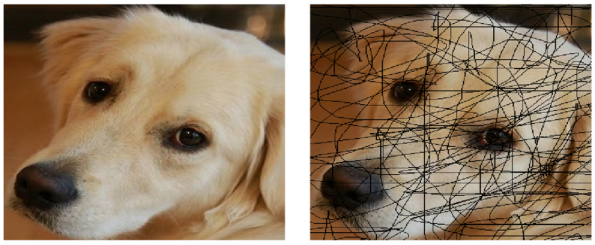

Fig. 8 The original "sherlock" image (left) and its painted one (right)

Figure 7 displays the relative error-norm (on the left), the rate of acceleration (in the middle), and PSNR (on the right) versus the iteration index for the three methods: TDPG, TG-MPE, and HOSVD-TMPE. Compared to TDPG (Basic Iterations) and TG-MPE, the curves confirm that our proposed algorithm is more effective.

Now, let us consider another color image obtained from 'sherlock.jpg' in the MATLAB R2020a collection. We corrupted it by adding a scribble, as illustrated in Fig. 8.

Figure 9 illustrates the restored images obtained using TDPG, TG-MPE, and the proposed algorithm (HOSVD-TMPE). It is evident that our proposed method and TG-MPE yield nearly identical results, with a slight advantage observed for HOSVD-TMPE. Compared to the basic iterative TDPG method, both HOSVD-TMPE and TG-MPE methods have effectively removed most of the superimposed scribble.

Table 3 presents various values, including the iteration number, relative error, recovered image PSNR, and required CPU time for TDPG (Basic Iterations), TG-MPE, and HOSVD-TMPE. For the size $150 \times 160 \times 3$, we stopped computations when the relative error was less than $\epsilon = 8 \times 10^{-2}$, while for the size $(325 \times 300 \times 3)$, we set $\epsilon = 10^{-1}$. The obtained results confirm the advantage of our proposed method in

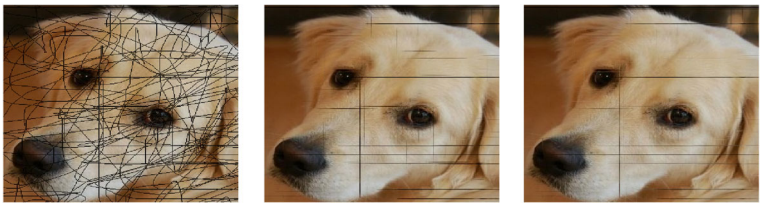

Fig. 9 The recovered images by TDPG (left), TG-MPE (middle) and HOSVD-TMPE (right) in 520 iterations

Table 3 Comparison of the required iteration number, relative error, PSNR and CPU time for TDPG (Basic-Iterations), TG-MPE and HOSVD-TMPE for different sizes of 'sherlok' image

Data size	Algorithms	Iter	Error	PSNR	CPU-time (s)
$(150 \times 160 \times 3)$	TDPG (Basic-iterations)	1296	7.99e−02	27.58	13.71
	TG-MPE ($k = 6$)	295	7.98e−02	27.59	3.55
	HOSVD-TMPE ($k = 6$)	235	7.92e−02	27.65	2.73
$(325 \times 300 \times 3)$	TDPG (Basic-iterations)	1850	1.02e−01	23.77	131.13
	TG-MPE ($k = 6$)	1087	9.99-02	18.95	82.25
	HOSVD-TMPE ($k = 6$)	589	9.99e−02	25.80	41.70

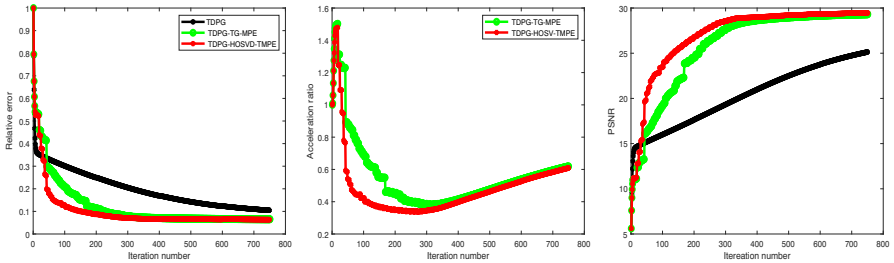

Fig. 10 The relative error (left), the acceleration rate (middle) and the PSNR (right) curves for 'sherlok' image

terms of PSNR, relative error, as well as the required number of iterations and CPU time.

The curves depicted in Fig. 10 illustrate the superior performance of our proposed method compared to TG-MPE. It is evident that HOSVD-TMPE excels in terms of PSNR, relative error-norm, and convergence rate, highlighting its effectiveness

7 Conclusion

The acceleration of slowly convergent sequences is an interesting purpose of extrapolation methods. In this paper, we introduced a new tensor extrapolation method for tensor sequences, namely HOSVD-TMPE. We presented an efficient algorithm that is based on high-order singular value decomposition for implementing the proposed

extrapolation method. The presented numerical tests show the performance and effectiveness of the algorithm. Additionally, we provide some applications in color image restoration and completion.

Acknowledgements Authors thank the anonymous referee for his very careful review of the paper and for his comments, corrections and suggestions that improved the paper.
The work was partially supported by the Moroccan Ministry of Higher Education, Scientific Research and Innovation and the OCP Foundation through the APRD research program.

Author Contributions The third author conceived of the presented idea, developed the theory and performed the computations. The first and the second authors verified the analytical methods and supervised the findings of this work. All authors discussed the results and contributed to the final manuscript.

References

1. Amblard, P.O., Gaeta, M., Lacoume, J.L.: Statistics for complex variables and signals-Part I: variables. Signal Process. **53**, 1–13 (1996)
2. Beik, F.P.A., El Ichi, A., Jbilou, K., Sadaka, R.: Tensor extrapolation methods with applications. Numer. Algor. **87**, 1421–1444 (2021)
3. Bellman, R.: Introduction to matrix analysis. SIAM (1970)
4. Benchettou, O., Bentbib, A.H., Bouhamidi, A.: An accelerated tensorial double proximal Gradient Method for Total variaitional regularization problem. J. Optim. Theory Appl. **198**, 111–134 (2023)
5. Bentbib, A.H., Khouia, A., Sadok, H.: The LSQR method for solving tensor least-squares problems. Electron. Trans. Numer. Anal. **55**, 92–111 (2021)
6. Bentbib, A.H., Jbilou, K., Tahiri, R., Bentbib, A.H., Jbilou, K., Tahiri, R.: N-mode minimal tensor extrapolation methods. Numer. Algor. **95**, 665–691 (2024)
7. Brezinski, C., Redivo-Zaglia, M.: Extrapolation methods: theory and practice. Elsevier (2013)
8. Candes, E.J., Tao, T.: The power of convex relaxation: near-optimal matrix completion. IEEE Trans. Inf. Theory. **56**, 2053–2080 (2010)
9. Chen, J., Saad, Y.: On the tensor SVD and the optimal low rank orthogonal approximation of tensors. SIAM J. Matrix Anal. Appl. **30**, 1709–1734 (2009)
10. Cipolla, S., Redivo-Zaglia, M., Tudisco, F.: Extrapolation methods for fixed? Point multilinear PageRank computations. Numer. Linear Algebra Appl. **27**, e2280 (2020)
11. Cipolla, S., Redivo-Zaglia, M., Tudisco, F.: Shifted and extrapolated power methods for tensor ℓ^p-eigenpairs. Electron. Trans. Numer. Anal. **53**, 1–27 (2020)
12. Delahaye, J.P., Germain-Bonne, B.: The set of logarithmically convergent sequences cannot be accelerated. SIAM J. Numer. Anal. **19**, 840–844 (1982)
13. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal Appl. **21**, 1253–1278 (2000)
14. Duminil, S., Sadok, H., Silvester, D.: Fast solvers for discretized Navier–Stokes problems using vector extrapolation. Numer Algor. **66**, 89–104 (2014)
15. El Guide, M., El Ichi, A., Jbilou, K., Beik, F.P.A.: Tensor Krylov subspace methods via the Einstein product with applications to image and video processing. Appl. Numer. Math. **181**, 347–363 (2022)
16. El Ichi, A., Jbilou, K., Sadaka, R.: Tensor global extrapolation methods using the n-mode and the Einstein products. Mathematics. **8**, 1298 (2020)
17. He, H., Xi, Y., Ho, J. C.: Accelerated SGD for tensor decomposition of sparse count data. Int. Conf. Data Mining Works, pp 284-291 (2020). https://doi.org/10.1109/ICDMW51313.2020.00047
18. He, H., Xi, Y., Ho, J. C.: Fast and accurate tensor decomposition without a high performance computing machine. IEEE Int. Conf. Big Data, pp 163-170 (2020). https://doi.org/10.1109/BigData50022.2020.9378111
19. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**, 455–500 (2009)
20. Lai, F., Li, W., Peng, X., Chen, Y.: Anderson accelerated fixed-point iteration for multilinear PageRank. Numer. Linear Algebra Appl. **30**, e2499 (2023)
21. Liang, M.L., Zheng, B., Zhao, R.J.: Tensor inversion and its application to the tensor equations with Einstein product. Linear Multilinear Algebra. **67**, 843–870 (2019)

22. Liu, D., Liu, X.: Restarted nonnegativity preserving tensor splitting methods via relaxed anderson acceleration for solving multi-linear systems (2022). arXiv preprint arXiv:2211.10857
23. Nesterov, Y.: Gradient methods for minimizing composite functions. Math. Program. **140**, 125–161 (2013)
24. Pollock, S., Shroff, R.: Accelerating the Computation of Tensor Z-eigenvalues (2023). arXiv preprint arXiv:2307.11908
25. Schosser, J.: Tensor extrapolation: an adaptation to data sets with missing entries. J. Big Data. **9**, 26 (2022)
26. Scieur, D., d'Aspremont, A., Bach, F.: Regularized nonlinear acceleration. Math. Program. **179**, 47–83 (2020)
27. Sidi, A.: Vector extrapolation methods with applications. SIAM (2017)
28. Sridevi, G., Srinivas-Kumar, S.: Image inpainting based on fractional order nonlinear diffusion for image reconstruction. Int. J. Circ. Syst. Signal Process. **38**, 3802–3817 (2019)
29. Wang, Q.W., Xu, X.: Iterative algorithms for solving some tensor equations. Linear Multilinear Algebra. **67**, 1325–1349 (2019)
30. Yuan, L., Zhao, Q., Gui, L., Cao, J.: High-order tensor completion via gradient-based optimization under tensor train format. Signal Process. Image Commun. **73**, 53–61 (2019)