Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

A tensor bidiagonalization method for higher-order singular value decomposition with applications

A. El Hachimi[1,2] | K. Jbilou[1,2] | A. Ratnani[1] | L. Reichel*[3]

[1]The UM6P Vanguard Center, Mohammed VI Polytechnic University, Green City, Morocco.
[2]LMPA, Université du Littoral Cote d'Opale, 50 rue F. Buisson, 62228 Calais-Cedex, France
[3]Department of Mathematical Sciences, Kent State University, OH, USA

Correspondence
*Lothar Reichel, USA. Email: reichel@math.kent.edu

Present Address
Present address

Abstract

The need to know a few singular triplets associated with the largest singular values of a third-order tensor arises in data compression and extraction. This paper describes a new method for their computation using the t-product. Methods for determining a couple of singular triplets associated with the smallest singular values also are presented. The proposed methods generalize available restarted Lanczos bidiagonalization methods for computing a few of the largest or smallest singular triplets of a matrix. The methods of this paper use Ritz and harmonic Ritz lateral slices to determine accurate approximations of the largest and smallest singular triplets, respectively. Computed examples show applications to data compression and face recognition.

KEYWORDS:
tensor, t-product, partial tensor bidiagonalization, restarted tensor bidiagonalization singular value decomposition; face recognition

1 | INTRODUCTION

The last 20 years has seen an immense growth of the amount of data that is collected for analysis, but it is a challenging problem to extract useful information from available data. This difficulty arises, e.g., in machine learning, data mining, and deep learning; see, e.g., Arnold et al.[1]. The extraction of useful information from data that is represented by a *matrix* often is facilitated by the singular value decomposition of the matrix. Typically, only a few of the largest singular triplets, i.e., the largest singular values and associated right and left singular vectors, are required to extract useful information from the matrix. A restarted Lanczos bidiagonalization method for computing accurate approximations of these singular triplets is described in[2], and R code written by Bryan W. Lewis is available at[3].

In many recent applications the given data are represented by a multidimensional array. These arrays, known as *tensors*, are natural generalizations of matrices. Several approaches to define tensor-tensor products and tensor-matrix products are described in the literature, including the n-mode product[4,5], the t-product[6,7], and the c-product[8,9]. Generalizations of the singular value decomposition (SVD) to tensors are described in[5] using the n-mode product (the so-called HOSVD), and in[8,6] using the tensor c-product and t-product. The need to compute the SVD or a partial SVD of a tensor arises in a variety of applications, including image restoration, tensor completion[10], robust tensor principal component analysis[11], tensor compression[12], and recognition of color faces[13,14,15]. These applications require knowledge of the largest singular values and associated lateral tensor singular slices.

It is the purpose of the this paper to introduce a new restarted tensor Lanczos bidiagonalization method for third-order tensors using the t-product for approximating a few of the largest singular values and associated lateral tensor singular slices. This method generalizes the approach described in[2] from matrices to tensors. We remark that the Lanczos bidiagonalization method (also

known as the Golub-Kahan bidiagonalization method) for third-order tensors using the t-product has been described in [16,17,6,18]; however, this bidiagonalization method differs from the one of the present paper.

In [2] the authors describe a restarted Lanczos bidiagonalization method for the computation of a few of the smallest singular values and associated singular vectors of a large matrix by determining harmonic Ritz values is presented. This paper presents an analogous scheme for third-order tensors.

The organization of this paper is as follows. Section 2 recalls some properties of the t-product and Section 3 reviews tensor Lanczos bidiagonalization of third-order tensors using the t-product. Restarted tensor Lanczos bidiagonalization methods are presented for the approximation of a few of the largest singular values and associated lateral tensor singular slices by computing lateral tensor Ritz slices, as well as for approximating a few of the smallest singular values and associated lateral tensor singular slices by evaluating harmonic lateral tensor Ritz slices. Section 4 discusses multidimensional principal component analysis using a partial tensor HOSVD with application to face recognition, and Section 5 presents a few computed examples. Concluding remarks can be found in Section 6.

2 | THE TENSOR T-PRODUCT

This section reviews results by Kilmer et al. [6,19] and uses notation employed there and by Kolda and Bader [5]. A third-order tensor is an array $\mathscr{A} = [a_{ijk}] \in \mathbb{R}^{\ell\times p\times n}$. Matrices and vectors are tensors of order two and one, respectively. A *slice* or *frame* of a third-order tensor $\mathscr{A}$ is a section obtained by fixing any one of the three indices. Using MATLAB notation, $\mathscr{A}(i,:,:)$, $\mathscr{A}(:,j,:)$, and $\mathscr{A}(:,:,k)$ denote the ith horizontal, the jth lateral, and the kth frontal slices of $\mathscr{A}$, respectively. The lateral slice $\mathscr{A}(:,j,:)$ also is denoted by $\vec{\mathscr{A}}_j$, and the frontal slice $\mathscr{A}(:,:,k)$ is an $\ell \times p$ matrix that is sometimes denoted by $\mathscr{A}^{(k)}$. A *fiber* of a third order tensor $\mathscr{A}$ is defined by fixing any two of the three indices. The fiber $\mathscr{A}(i,j,:)$ is called a *tube* of $\mathscr{A}$. We will use capital calligraphic letters $\mathscr{A}$ to denote third-order tensors, capital letters A to identify matrices, bold face lower case letters $\boldsymbol{a}$ to denote tubes, and lower case letters a stand for scalars. Further, $\mathbb{K}_n^{\ell\times p} = \mathbb{R}^{\ell\times p\times n}$ denotes the space of third-order tensors of size $\ell\times p\times n$, $\mathbb{K}_n^{\ell} = \mathbb{R}^{\ell\times 1\times n}$ stands for the space of lateral slices of size $\ell \times n$, and $\mathbb{K}_n = \mathbb{R}^{1\times 1\times n}$ denotes the space of tubes with n entries. For a third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$ with frontal slices $\mathscr{A}^{(i)}$, $i = 1,\ldots,n$, we define:

- The block circulant matrix associated with $\mathscr{A}$:

$$\texttt{bcirc}(\mathscr{A}) = \begin{bmatrix} \mathscr{A}^{(1)} & \mathscr{A}^{(n)} & \ldots & \mathscr{A}^{(2)} \\ \mathscr{A}^{(2)} & \mathscr{A}^{(1)} & \ldots & \mathscr{A}^{(3)} \\ \vdots & \ddots & \ddots & \vdots \\ \mathscr{A}^{(n)} & \mathscr{A}^{(n-1)} & \ldots & \mathscr{A}^{(1)} \end{bmatrix} \in \mathbb{K}^{\ell n\times pn}. \tag{1}$$

- The operator `unfold` applied to $\mathscr{A}$ gives the matrix made up of its frontal slices,

$$\texttt{unfold}(\mathscr{A}) = \begin{bmatrix} \mathscr{A}^{(1)} \\ \mathscr{A}^{(2)} \\ \vdots \\ \mathscr{A}^{(n)} \end{bmatrix} \in \mathbb{K}^{\ell n\times p}.$$

  We also will need the inverse operator `fold` such that $\texttt{fold}\,(\texttt{unfold}\,(\mathscr{A})) = \mathscr{A}$.

- The block diagonal matrix associated with $\mathscr{A}$ is defined as

$$\texttt{bdiag}\,(\mathscr{A}) = \begin{bmatrix} \mathscr{A}^{(1)} & & & \\ & \mathscr{A}^{(2)} & & \\ & & \ddots & \\ & & & \mathscr{A}^{(n)} \end{bmatrix} \in \mathbb{K}^{\ell n\times pn}.$$

Definition 1. ([19]) Let $\mathscr{A} \in \mathbb{K}_n^{\ell\times q}$ and $\mathscr{B} \in \mathbb{K}_n^{q\times p}$ be third-order tensors. The t-product of $\mathscr{A}$ and $\mathscr{B}$ is defined by

$$\mathscr{A} \star \mathscr{B} := \texttt{fold}\,(\texttt{bcirc}(\mathscr{A})\,\texttt{unfold}(\mathscr{B})) \in \mathbb{K}_n^{\ell\times p}.$$

The block circulant matrix (1) can be block-diagonalized by using the discrete Fourier transform (DFT) as follows:

$$\texttt{bcirc}(\mathscr{A}) = \left(F_n^H \otimes I_\ell\right) \texttt{bdiag}(\widehat{\mathscr{A}}) \left(F_n \otimes I_p\right),$$

where $F_n \in \mathbb{C}^{n\times n}$ is the discrete Fourier matrix, F_n^H denotes its conjugate transpose, $\widehat{\mathscr{A}}$ stands for the Fourier transform of $\mathscr{A}$ along each tube, $I_\ell \in \mathbb{R}^{\ell\times\ell}$ denotes the identity matrix, and $\otimes$ is the Kronecker product. The matrix $\widehat{\mathscr{A}}$ can be computed with the fast Fourier transform (FFT) algorithm; see [19] for details. Using MATLAB notations, we have

$$\widehat{\mathscr{A}} = \texttt{fft}(\mathscr{A}, [\,], 3).$$

The inverse operation can be evaluated in MATLAB with the command

$$\mathscr{A} = \texttt{ifft}(\widehat{\mathscr{A}}, [\,], 3).$$

Hence, the t-product $\mathscr{C} = \mathscr{A} \star \mathscr{B}$ can be evaluated by using

$$\widehat{\mathscr{C}}^{(i)} = \widehat{\mathscr{A}}^{(i)}\widehat{\mathscr{B}}^{(i)}, \qquad i = 1, 2, \ldots, n, \tag{2}$$

where $\widehat{\mathscr{A}}^{(i)}$, $\widehat{\mathscr{B}}^{(i)}$, and $\widehat{\mathscr{C}}^{(i)}$ are the ith frontal slices of the tensors $\widehat{\mathscr{A}}$, $\widehat{\mathscr{B}}$, and $\widehat{\mathscr{C}}$, respectively.

As already pointed out by Kilmer et al. [6], one can use symmetry properties of the DFT when applied to real data to reduce the computational effort when evaluating the t-product with the FFT. This is described by the following result, which can be found, e.g., in [20].

Lemma 1. Given a real vector $v \in \mathbb{R}^n$, the associated DFT vector $\widehat{v} = F_n v$ satisfies

$$\widehat{v}_1 \in \mathbb{R}, \quad \texttt{conj}\left(\widehat{v}_i\right) = \widehat{v}_{n-i+2}, \quad i = 2, 3, \ldots, \left\lceil \frac{n+1}{2} \right\rceil,$$

where conj denotes the complex conjugation operator and $\left\lceil \frac{n+1}{2} \right\rceil$ is the integer part of $\frac{n+1}{2}$.

It follows that for a third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$, we have

$$\widehat{\mathscr{A}}^{(1)} \in \mathbb{R}^{\ell\times p}, \quad \texttt{conj}\left(\widehat{\mathscr{A}}^{(i)}\right) = \widehat{\mathscr{A}}^{(n-i+2)}, \quad i = 2, 3, \ldots, \left\lceil \frac{n+1}{2} \right\rceil.$$

This shows that the t-product of two third-order tensors can be determined by evaluating just about half the number of products involved in (2). Algorithm 1 describes the computations.

Algorithm 1 t-product of third-order tensors.

Input: $\mathscr{A} \in \mathbb{K}_n^{\ell\times q}$, $\mathscr{B} \in \mathbb{K}_n^{q\times p}$.
Output: $\mathscr{C} := \mathscr{A} \star \mathscr{B} \in \mathbb{K}_n^{\ell\times p}$.

1: Compute $\widehat{\mathscr{A}} = \texttt{fft}(\mathscr{A}, [\,], 3)$, $\widehat{\mathscr{B}} = \texttt{fft}(\mathscr{B}, [\,], 3)$.
2: **for** $i = 1, \ldots, \left\lceil \frac{n+1}{2} \right\rceil$ **do**
3: $\quad \widehat{\mathscr{C}}^{(i)} = \widehat{\mathscr{A}}^{(i)}\widehat{\mathscr{B}}^{(i)}$.
4: **end for**
5: **for** $i = \left\lceil \frac{n+1}{2} \right\rceil + 1, \ldots, n$ **do**
6: $\quad \widehat{\mathscr{C}}^{(i)} = \texttt{conj}\left(\widehat{\mathscr{C}}^{(n-i+2)}\right)$.
7: **end for**
8: $\mathscr{C} = \texttt{ifft}\left(\widehat{\mathscr{C}}, [\,], 3\right)$.

The following definition is concerned with the t-product of a third-order tensor and a tube.

Definition 2. Let $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$ and $\boldsymbol{b} \in \mathbb{K}_n$. Then $\mathscr{C} := \mathscr{A} \star \boldsymbol{b} \in \mathbb{K}_n^{\ell\times p}$ is obtained by applying the inverse DFT along each tube of $\widehat{\mathscr{C}}$, where each frontal slice is determined by the standard matrix product between each frame of $\widehat{\mathscr{A}}$ and $\widehat{\boldsymbol{b}}$, i.e.,

$$\widehat{\mathscr{C}}^{(i)} = \widehat{\mathscr{A}}^{(i)}\widehat{\boldsymbol{b}}^{(i)} = \widehat{\boldsymbol{b}}^{(i)}\widehat{\mathscr{A}}^{(i)}, \quad i = 1, 2, \ldots, n.$$

A third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$ can be written as

$$\mathscr{A} = \left[\vec{\mathscr{A}}_1, \vec{\mathscr{A}}_2, \ldots, \vec{\mathscr{A}}_p\right],$$

thus, for the tensors $\mathscr{A} \in \mathbb{K}_n^{\ell\times q}$ and $\mathscr{B} \in \mathbb{K}_n^{q\times p}$, the t-product $\mathscr{A} \star \mathscr{B}$ can be expressed as

$$\mathscr{A} \star \mathscr{B} = \left[\mathscr{A} \star \vec{\mathscr{B}}_1, \mathscr{A} \star \vec{\mathscr{B}}_2, \dots, \mathscr{A} \star \vec{\mathscr{B}}_p\right],$$

where

$$\mathscr{A} \star \overrightarrow{\mathscr{B}}_i = \overrightarrow{(\mathscr{A} \star \mathscr{B})}_i, \quad i = 1, 2, \dots, p.$$

The Frobenius norm of a third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$ is given by

$$\|\mathscr{A}\|_F := \sqrt{\sum_{i_1,i_2,i_3=1}^{\ell,p,n} a_{i_1,i_2,i_3}^2},$$

and the inner product of two third-order tensors of the same size $\mathscr{A}, \mathscr{B} \in \mathbb{K}_n^{\ell\times p}$ is defined as

$$\langle \mathscr{A}, \mathscr{B} \rangle := \sum_{i_1,i_2,i_3=1}^{\ell,p,n} a_{i_1,i_2,i_3} b_{i_1,i_2,i_3}.$$

We have the relations

$$\|\mathscr{A}\|_F = \frac{1}{\sqrt{n}} \left\|\widehat{\mathscr{A}}\right\|_F, \qquad \langle \mathscr{A}, \mathscr{B} \rangle = \frac{1}{n} \langle \widehat{\mathscr{A}}, \widehat{\mathscr{B}} \rangle.$$

We recall for later use the definitions of some special tensors and operations:

- The identity tensor $\mathscr{I}_\ell \in \mathbb{K}_n^{\ell\times\ell}$ is the tensor whose first frontal slice is the identity matrix and all other slices have zero entries only.
- The transpose of a real third-order tensor, $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$, denoted by $\mathscr{A}^H \in \mathbb{K}_n^{p\times\ell}$, is the tensor obtained by first transposing each one of the frontal slices of $\mathscr{A}$, and then reversing the order of the transposed frontal slices 2 through n; see [19]. Let the third-order tensors $\mathscr{A}$ and $\mathscr{B}$ be such that the products $\mathscr{A} \star \mathscr{B}$ and $\mathscr{B}^H \star \mathscr{A}^H$ are defined. Then, similarly to the matrix transpose, the tensor transpose satisfies $(\mathscr{A} \star \mathscr{B})^H = \mathscr{B}^H \star \mathscr{A}^H$.
- A tensor $\mathscr{Q} \in \mathbb{K}_n^{\ell\times\ell}$ is said to be orthogonal if and only if

$$\mathscr{Q}^H \star \mathscr{Q} = \mathscr{Q} \star \mathscr{Q}^H = \mathscr{I}_\ell.$$

- A square third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell\times\ell}$ is invertible if there is a third-order tensor $\mathscr{B} \in \mathbb{K}_n^{\ell\times\ell}$ such that

$$\mathscr{A} \star \mathscr{B} = \mathscr{I}_\ell, \quad \mathscr{B} \star \mathscr{A} = \mathscr{I}_\ell.$$

  In this case $\mathscr{B}$ is said to be the inverse of $\mathscr{A}$, and is denoted by $\mathscr{A}^{-1}$.

Definition 3. ([6]) Let $\vec{\mathscr{A}}_i \in \mathbb{K}_n^\ell$ for $i = 1, 2, \dots, p$ be lateral slices of the tensor $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$. A t-linear combination of these slices is defined as

$$\vec{\mathscr{A}}_1 \star \boldsymbol{b}_1 + \vec{\mathscr{A}}_2 \star \boldsymbol{b}_2 + \dots + \vec{\mathscr{A}}_p \star \boldsymbol{b}_p,$$

where the $\boldsymbol{b}_i$ for $i = 1, 2, \dots, p$ are tubes in $\mathbb{K}_n$. Moreover,

$$\mathtt{span}\left\{\vec{\mathscr{A}}_1, \vec{\mathscr{A}}_2, \dots, \vec{\mathscr{A}}_p\right\} = \left\{\sum_{i=1}^{p} \vec{\mathscr{A}}_i \star \boldsymbol{b}_i : \ \boldsymbol{b}_i \in \mathbb{K}_n, \ i = 1, 2, \dots, p\right\}.$$

The tensor singular value decomposition (t-SVD) associated with the t-product, introduced by Kilmer and Martin [19], generalizes the classical SVD of a matrix. It is described in the next theorem.

Theorem 1. ([19]) Let $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$ be a third-order tensor. Then it can be represented as the t-product of three third-order tensors,

$$\mathscr{A} = \mathscr{U} \star \mathscr{S} \star \mathscr{V}^H, \tag{3}$$

where $\mathscr{U} \in \mathbb{K}_n^{\ell\times\ell}$ and $\mathscr{V} \in \mathbb{K}_n^{p\times p}$ are orthogonal tensors, and $\mathscr{S} \in \mathbb{K}_n^{\ell\times p}$ is an f-diagonal tensor, i.e., each frontal slice of the DFT of $\mathscr{S}$ is a diagonal matrix.

Algorithm 2 The t-SVD of a third-order tensor.

Input: $\mathscr{A} \in \mathbb{K}_n^{\ell \times p}$.
Output: $\mathscr{U} \in \mathbb{K}_n^{\ell \times \ell}$, $\mathscr{S} \in \mathbb{K}_n^{\ell \times p}$, $\mathscr{V} \in \mathbb{K}_n^{p \times p}$.

1: $\widehat{\mathscr{A}} = \texttt{fft}(\mathscr{A}, [\,], 3)$.
2: **for** $i = 1, \ldots, \left\lceil \frac{n+1}{2} \right\rceil$ **do**
3: $\quad [\widehat{\mathscr{U}}^{(i)}, \widehat{\mathscr{S}}^{(i)}, \widehat{\mathscr{V}}^{(i)}] = \texttt{svd}(\widehat{\mathscr{A}}^{(i)})$.
4: **end for**
5: **for** $i = 1, \ldots, \left\lceil \frac{n+1}{2} \right\rceil + 1$ **do**
6: $\quad \widehat{\mathscr{U}}^{(i)} = \texttt{conj}\left(\widehat{\mathscr{U}}^{(n-i+2)}\right)$, $\widehat{\mathscr{S}}^{(i)} = \texttt{conj}\left(\widehat{\mathscr{S}}^{(n-i+2)}\right)$, and $\widehat{\mathscr{V}}^{(i)} = \texttt{conj}\left(\widehat{\mathscr{V}}^{(n-i+2)}\right)$.
7: **end for**
8: Compute $\mathscr{U} = \texttt{ifft}(\widehat{\mathscr{U}}, [\,], 3)$, $\mathscr{S} = \texttt{ifft}(\widehat{\mathscr{S}}, [\,], 3)$, and $\mathscr{V} = \texttt{ifft}(\widehat{\mathscr{V}}, [\,], 3)$.

Algorithm 2 summarizes the computation of the t-SVD of a third-order tensor with the aid of the FFT.

The factorization (3) can be expressed as

$$\mathscr{A} = \mathscr{U} \star \mathscr{S} \star \mathscr{V}^H = \sum_{i=1}^{\min\{\ell,p\}} \vec{\mathscr{U}}_i \star \boldsymbol{s}_i \star \vec{\mathscr{V}}_i^H,$$

where the $\boldsymbol{s}_i = \mathscr{S}(i,i,:)$ are singular tubes, and $\vec{\mathscr{U}}_i = \mathscr{U}(:,i,:)$ and $\vec{\mathscr{V}}_i = \mathscr{U}(:,i,:)$ are right and left lateral tensor singular slices, respectively, for $i = 1, 2, \ldots, \min\{\ell, p\}$. The triplets $\{\boldsymbol{s}_i, \vec{\mathscr{U}}_i, \vec{\mathscr{V}}_i\}_{i=1:\min(\ell,p)}$ will be referred to as singular triplets of the tensor $\mathscr{A}$. The singular tubes are ordered so that their norms $\sigma_i = \|\boldsymbol{s}_i\|_F$ are decreasing with i, i.e.,

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min\{\ell,p\}} \geq 0.$$

Note that we also have the relations

$$\mathscr{A} \star \vec{\mathscr{V}}_i = \vec{\mathscr{U}}_i \star \boldsymbol{s}_i, \quad \mathscr{A}^H \star \vec{\mathscr{U}}_i = \vec{\mathscr{V}}_i \star \boldsymbol{s}_i, \quad i = 1, 2, \ldots, \min\{\ell, p\}.$$

We note for future reference that

$$\mathscr{S}(i,i,1) = \sum_{j=1}^{n} \frac{1}{n} \widehat{\mathscr{S}}(i,i,j). \tag{4}$$

In the following, we will need the notion of rank of a third-order tensor.

Definition 4. Let $\mathscr{A} \in \mathbb{K}_n^{\ell \times p}$ be a third-order tensor. Then its tubal rank is defined as

$$\texttt{rank}_t(\mathscr{A}) = \texttt{card}\left\{\sigma_i > 0, \;\; i = 1, 2, \ldots, \min\{\ell, p\}\right\},$$

where σ_i is the norm of the singular tube $\boldsymbol{s}_i$ of $\mathscr{A}$ and card stands for the cardinality.

The next result generalizes the Eckart-Young theorem for matrices to third-order tensors. It is important in the context of data compression.

Theorem 2. ([12,19]) Let the t-SVD of a third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell \times p}$ be given by $\mathscr{A} = \mathscr{U} \star \mathscr{S} \star \mathscr{V}^H$. For $1 \leq k \leq \min\{\ell, p\}$, define the truncated t-SVD by

$$\mathscr{A}_k = \sum_{i=1}^{k} \vec{\mathscr{U}}_i \star \boldsymbol{s}_i \star \vec{\mathscr{V}}_i^H.$$

Then

$$\mathscr{A}_k = \underset{\widetilde{\mathscr{A}} \in \mathbb{M}}{\arg\min} \left\| \mathscr{A} - \widetilde{\mathscr{A}} \right\|_F,$$

where $\mathbb{M}$ is the set given by $\mathbb{M} = \{\mathscr{X} \star \mathscr{Y};\ \text{with } \mathscr{X} \in \mathbb{K}_n^{\ell \times k},\ \mathscr{Y} \in \mathbb{K}_n^{k \times p}\}$.

The matrix QR factorization also can be generalized to tensors.

Theorem 3. ([19]) Let $\mathscr{A} \in \mathbb{K}_n^{\ell \times p}$. Then $\mathscr{A}$ can be factored as

$$\mathscr{A} = \mathscr{Q} \star \mathscr{R}, \tag{5}$$

where $\mathcal{Q} \in \mathbb{K}_n^{\ell\times\ell}$ is an orthogonal tensor and $\mathcal{R} \in \mathbb{K}_n^{\ell\times p}$ is an f-upper triangular tensor, i.e., each frontal slice of the DFT of $\mathcal{R}$ is an upper triangular matrix. The factorization (5) is referred to as the t-QR factorization of $\mathcal{A}$.

Algorithm 3 summarizes the computation of the t-QR factorization (5). The function `qr` in line 3 of the algorithm computes a QR factorization of the matrix $\widehat{\mathcal{A}^{(i)}} \in \mathbb{C}^{\ell\times p}$; thus $\widehat{\mathcal{A}^{(i)}} = \widehat{\mathcal{Q}}^{(i)}\widehat{\mathcal{R}}^{(i)}$, where the matrix $\widehat{\mathcal{Q}}^{(i)} \in \mathbb{C}^{\ell\times\ell}$ is orthogonal and the matrix $\widehat{\mathcal{R}}^{(i)} \in \mathbb{C}^{\ell\times p}$ has an upper triangular leading principal submatrix of order ℓ.

Algorithm 3 t-QR factorization of a third-order tensor.

Input: $\mathcal{A} \in \mathbb{K}_n^{\ell\times p}$.
Output: $\mathcal{Q} \in \mathbb{K}_n^{\ell\times\ell}$, $\mathcal{R} \in \mathbb{K}_n^{\ell\times p}$.

1: $\widehat{\mathcal{A}} = \texttt{fft}(\mathcal{A}, [\,], 3)$.
2: **for** $i = 1 \ldots, \left\lceil \frac{n+1}{2} \right\rceil$ **do**
3: $[\widehat{\mathcal{Q}}^{(i)}, \widehat{\mathcal{R}}^{(i)}] = \texttt{qr}(\widehat{\mathcal{A}^{(i)}})$.
4: **end for**
5: **for** $i = \left\lceil \frac{n+1}{2} \right\rceil + 1 \ldots, n$ **do**
6: $\widehat{\mathcal{Q}}^{(i)} = \texttt{conj}\left(\widehat{\mathcal{Q}}^{(n-i+2)}\right)$ and $\widehat{\mathcal{R}}^{(i)} = \texttt{conj}\left(\widehat{\mathcal{R}}^{(n-i+2)}\right)$.
7: **end for**
8: Compute $\mathcal{Q} = \texttt{ifft}(\widehat{\mathcal{Q}}, [\,], 3)$ and $\mathcal{R} = \texttt{ifft}(\widehat{\mathcal{R}}, [\,], 3)$.

Following Kilmer et al.[6], we define orthogonality of lateral tensor slices. Let $\vec{\mathcal{X}}$ and $\vec{\mathcal{Y}}$ be two lateral tensor slices in $\mathbb{K}_n^{\ell}$ and define the inner product of these slices as

$$\left\langle \vec{\mathcal{X}}, \vec{\mathcal{Y}} \right\rangle := \vec{\mathcal{X}}^H \star \vec{\mathcal{Y}} \in \mathbb{K}_n.$$

The lateral slices in the set

$$\left\{ \vec{\mathcal{X}}_1, \vec{\mathcal{X}}_2, \ldots, \vec{\mathcal{X}}_p \right\}, \tag{6}$$

with $p \geq 2$, are said to be *orthogonal* if

$$\left\langle \vec{\mathcal{X}}_i, \vec{\mathcal{X}}_j \right\rangle = \begin{cases} \alpha_i \mathbf{e} & \text{if } i = j, \\ \mathbf{0} & \text{if } i \neq j, \end{cases} \tag{7}$$

where $\mathbf{e}$ is the tube in $\mathbb{K}_n$, whose first element is 1 and the remaining elements vanish, and the α_i, $i = 1, 2, \ldots, p$, are nonvanishing scalars. Furthermore, if $\alpha_i = 1$ for all $i = 1, 2, \ldots, p$, then the set (6) is said to be *orthonormal*.

Following[6], we observe that any lateral slice $\vec{\mathcal{X}} \in \mathbb{K}_n^{\ell}$ can be normalized as

$$\vec{\mathcal{X}} = \vec{\mathcal{Y}} \star \mathbf{a} \tag{8}$$

with $\vec{\mathcal{Y}} \in \mathbb{K}_n^{\ell}$, $\left\|\vec{\mathcal{Y}}\right\| = 1$, and $\mathbf{a} \in \mathbb{K}_n$. Here the tensor norm is defined as

$$\left\|\vec{\mathcal{Y}}\right\| = \frac{\left\|\left\langle \vec{\mathcal{Y}}, \vec{\mathcal{Y}} \right\rangle\right\|_F}{\left\|\vec{\mathcal{Y}}\right\|_F}.$$

Note that $\vec{\mathcal{Y}}$ has unit norm if and only if $\left\langle \vec{\mathcal{Y}}, \vec{\mathcal{Y}} \right\rangle = \boldsymbol{e}$; see[6] for more details. Algorithm 4 summarizes the normalization process. The MATLAB function `randn` in the algorithm generates a vector in $\mathbb{R}^{\ell}$ with normally distributed pseudorandom entries with mean zero and variance one.

3 | TENSOR LANCZOS BIDIAGONALIZATION FOR COMPUTING THE LARGEST AND SMALLEST SINGULAR TRIPLETS

This section describes the Lanczos bidiagonalization process for tensors using the t-product, and discusses how approximations of the largest and smallest singular triplets of a large third-order tensor $\mathcal{A} \in \mathbb{K}_n^{\ell\times p}$ can be computed.

Algorithm 4 `Normalize`($\vec{\mathscr{X}}$).

Input: $\vec{\mathscr{X}} \in \mathbb{K}_n^\ell$.
Output: $\vec{\mathscr{Y}} \in \mathbb{K}_n^\ell$ of unit norm and $\mathbf{a} \in \mathbb{K}_n$ that satisfy (8).

$\vec{\widehat{\mathscr{Y}}} = \texttt{fft}(\vec{\mathscr{X}}, [\,], 3)$.
for $i = 1, \ldots, \left\lceil \frac{n+1}{2} \right\rceil$ **do**
  $\widehat{\mathbf{a}}^{(i)} = \left\| \vec{\widehat{\mathscr{Y}}}^{(i)} \right\|_F$.
  if $\widehat{\mathbf{a}}^{(i)} > 0$ **then**
    $\vec{\widehat{\mathscr{Y}}}^{(i)} = \dfrac{\vec{\widehat{\mathscr{Y}}}^{(i)}}{\widehat{\mathbf{a}}^{(i)}}$
  else
    $\vec{\widehat{\mathscr{Y}}}^{(i)} = \texttt{randn}(\ell, 1)$; $\mathbf{b}^{(i)} = \left\| \vec{\widehat{\mathscr{Y}}}^{(i)} \right\|_F$, and $\vec{\widehat{\mathscr{Y}}}^{(i)} = \dfrac{\vec{\widehat{\mathscr{Y}}}^{(i)}}{\mathbf{b}^{(i)}}$.
  end if
end for
for $i = \left\lceil \frac{n+1}{2} \right\rceil + 1, \ldots, n$ **do**
  $\vec{\widehat{\mathscr{Y}}}^{(i)} = \texttt{conj}\left(\vec{\widehat{\mathscr{Y}}}^{(n-i+2)}\right)$, $\widehat{\mathbf{a}}^{(i)} = \texttt{conj}\left(\widehat{\mathbf{a}}^{(n-i+2)}\right)$.
end for
$\vec{\mathscr{Y}} = \texttt{ifft}(\vec{\widehat{\mathscr{Y}}}, [\,], 3)$, $\mathbf{a} = \texttt{ifft}(\widehat{\mathbf{a}}, [\,], 3)$.

3.1 | The tensor Lanczos bidiagonalization algorithm

The Lanczos bidiagonalization process was introduced for matrices by Golub and Kahan[21] and therefore sometimes is referred to as the Golub-Kahan bidiagonalization process. For a matrix $A \in \mathbb{R}^{\ell \times p}$, this process is closely related to symmetric Lanczos process applied to the real symmetric matrices AA^T and A^TA, or alternatively to the symmetric matrix

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

Lanczos bidiagonalization algorithms have been applied to solve numerous problems such as large-scale least squares problem[22], the approximation of the largest or smallest singular triplets of a large matrix[2,23,24], and in Tikhonov regularization of large linear discrete ill-posed problems; see, e.g.,[25,26]. We note that the bidiagonalization method described in[22] and applied in[25,26] reduces a large matrix A to a small lower bidiagonal matrix, while in[2] the matrix A is reduced to a small upper bidiagonal matrix. We will review the latter approach.

Application of $m \ll \min\{\ell, p\}$ steps of the Lanczos bidiagonalization process to the matrix $A \in \mathbb{R}^{\ell \times p}$ with the initial unit vector $p_1 \in \mathbb{R}^\ell$ generically produces two matrices

$$P_m = \left[p_1, p_2, \ldots, p_m\right] \in \mathbb{R}^{p \times m}, \quad Q_m = \left[q_1, q_2, \ldots, q_m\right] \in \mathbb{R}^{\ell \times m}.$$

The columns of P_m and Q_m form orthonormal bases for the Krylov subspaces

$$\begin{aligned} \mathscr{K}_m\left(A^TA, p_1\right) &= \texttt{span}\{p_1, A^TAp_1, \left(A^TA\right)^2 p_1, \ldots, \left(A^TA\right)^{m-1} p_1\}, \\ \mathscr{K}_m\left(AA^T, q_1\right) &= \texttt{span}\{q_1, AA^Tq_1, \left(AA^T\right)^2 q_1, \ldots, \left(AA^T\right)^{m-1} q_1\}, \end{aligned}$$

respectively, where $q_1 = Ap_1/\|Ap_1\|_2$. A matrix interpretation of the recursion relations of the Lanczos process gives the matrix relations

$$AP_m = Q_mB_m, \tag{9}$$
$$A^TQ_m = P_mB_m^T + r_me_m^T, \tag{10}$$

where $e_m = [0, \dots, 0, 1]^T \in \mathbb{R}^m$, and $r_m \in \mathbb{R}^p$ which satisfies $P_m^T r_m = 0$. The matrix $B_m \in \mathbb{R}^{m\times m}$ is upper bidiagonal and satisfies $B_m = Q_m^T A P_m$. Equation (10) can be rewritten as

$$A^T Q_m = P_{m+1} B_{m,m+1}^T,$$

where $B_{m,m+1} = \left[B_m, \beta_m e_m\right] \in \mathbb{R}^{m\times(m+1)}$, $P_{m+1} = \left[P_m, p_{m+1}\right] \in \mathbb{R}^{p\times(m+1)}$, and $r_m = \beta_m p_{m+1}$ with $\beta_m = \|r_m\|_F$.

When considering bidiagonalization of a third-order tensor $\mathscr{A}$ using the t-product, the scalars and the columns of the matrices P_m and Q_m in the matrix decompositions (9) and (10) become tubes and lateral slices, respectively, in the decompositions determined by the tensor Lanczos bidiagonalization process. The application of m steps of tensor Lanczos bidiagonalization to the third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$ generically computes two tensors

$$\mathscr{P}_m = \left[\vec{\mathscr{P}}_1, \vec{\mathscr{P}}_2, \dots, \vec{\mathscr{P}}_m\right] \in \mathbb{K}_n^{p\times m} \text{ and } \mathscr{Q}_m = \left[\vec{\mathscr{Q}}_1, \vec{\mathscr{Q}}_2, \dots, \vec{\mathscr{Q}}_m\right] \in \mathbb{K}_n^{\ell\times m},$$

whose lateral slices form bases for the tensor Krylov subspaces $\mathscr{K}_m\left(\mathscr{A}^H \star \mathscr{A}, \vec{\mathscr{P}}_1\right)$ and $\mathscr{K}_m\left(\mathscr{A} \star \mathscr{A}^H, \vec{\mathscr{Q}}_1\right)$, respectively. They are defined by

$$\mathscr{K}_m\left(\mathscr{A}^H \star \mathscr{A}, \vec{\mathscr{P}}_1\right) = \mathtt{span}\{\vec{\mathscr{P}}_1, \left(\mathscr{A}^H \star \mathscr{A}\right) \star \vec{\mathscr{P}}_1, \dots, \left(\mathscr{A}^H \star \mathscr{A}\right)^{m-1} \star \vec{\mathscr{P}}_1\},$$
$$\mathscr{K}_m\left(\mathscr{A} \star \mathscr{A}^H, \vec{\mathscr{Q}}_1\right) = \mathtt{span}\{\vec{\mathscr{Q}}_1, \left(\mathscr{A} \star \mathscr{A}^H\right) \star \vec{\mathscr{Q}}_1, \dots, \left(\mathscr{A} \star \mathscr{A}^H\right)^{m-1} \star \vec{\mathscr{Q}}_1\},$$

where $\vec{\mathscr{P}}_1 \in \mathbb{K}_n^p$ is a lateral slice of unit norm, and the lateral slice $\vec{\mathscr{Q}}_1 \in \mathbb{K}_n^\ell$ is of unit norm and proportional to $\mathscr{A} \star \vec{\mathscr{P}}_1$. Algorithm 5 describes the tensor Lanczos bidiagonalization algorithm.

Algorithm 5 Tensor Lanczos bidiagonalization using the t-product.

Input: $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$, number of steps $m \le \min\{\ell, p\}$, $\vec{\mathscr{P}}_1 \in \mathbb{K}_n^p$ with unit norm.
Output: $\mathscr{P}_m = [\vec{\mathscr{P}}_1, \vec{\mathscr{P}}_2, \dots, \vec{\mathscr{P}}_m] \in \mathbb{K}_n^{p\times m}$ and $\mathscr{Q}_m = [\vec{\mathscr{Q}}_1, \vec{\mathscr{Q}}_2, \dots, \vec{\mathscr{Q}}_m] \in \mathbb{K}_n^{\ell\times m}$ with orthonormal lateral slices, $\mathscr{B}_m \in \mathbb{K}_n^{m\times m}$ a bidiagonal tensor, and $\vec{\mathscr{R}}_m \in \mathbb{K}_m^\ell$.

$\mathscr{P}_1 = \left[\vec{\mathscr{P}}_1\right]$.
$\vec{\mathscr{Q}}_1 = \mathscr{A} \star \vec{\mathscr{P}}_1$.
$[\vec{\mathscr{Q}}_1, \boldsymbol{\alpha}_1] = \mathtt{Normalize}(\vec{\mathscr{Q}}_1)$.
$\mathscr{Q}_1 = \left[\vec{\mathscr{Q}}_1\right]$, $\mathscr{B}_m(1, 1, :) = \boldsymbol{\alpha}_1$.
for $i = 1$ to m **do**
  $\vec{\mathscr{R}}_i = \mathscr{A}^H \star \vec{\mathscr{Q}}_i - \boldsymbol{\alpha}_i \star \vec{\mathscr{P}}_i$.
  Reorthogonalization $\vec{\mathscr{R}}_i = \vec{\mathscr{R}}_i - \mathscr{P}_i \star (\mathscr{P}_i^H \star \vec{\mathscr{R}}_i)$.
  if $i < m$ **then**
    $[\vec{\mathscr{P}}_{i+1}, \boldsymbol{\beta}_i] = \mathtt{Normalize}(\vec{\mathscr{R}}_i)$.
    $\mathscr{P}_{i+1} = \left[\mathscr{P}_i, \vec{\mathscr{P}}_{i+1}\right]$, $\mathscr{B}_m(i, i+1, :) = \boldsymbol{\beta}_i$.
    $\vec{\mathscr{Q}}_{i+1} = \mathscr{A} \star \vec{\mathscr{P}}_{i+1} - \boldsymbol{\beta}_i \star \vec{\mathscr{Q}}_i$.
    Reorthogonalization $\vec{\mathscr{Q}}_{i+1} = \vec{\mathscr{Q}}_{i+1} - \mathscr{Q}_i \star (\mathscr{Q}_i^H \star \vec{\mathscr{Q}}_{i+1})$.
    $[\vec{\mathscr{Q}}_{i+1}, \boldsymbol{\alpha}_{i+1}] = \mathtt{Normalize}(\vec{\mathscr{Q}}_{i+1})$.
    $\mathscr{Q}_{i+1} = \left[\mathscr{Q}_i, \vec{\mathscr{Q}}_{i+1}\right]$, $\mathscr{B}_m(i+1, i+1, :) = \boldsymbol{\alpha}_{i+1}$.
  end if
end for

We remark that Algorithm 5 differs from the tensor bidiagonalization algorithms described in [6,18] in that the former produces an upper bidiagonal tensor $\mathscr{B}_m$, while the latter determine a lower bidiagonal tensor. The use of an upper bidiagonal tensor in the present paper is inspired by the choices in [2,21]. Algorithm 5 is said to break down when one of the tensor slices $\vec{\mathscr{R}}_i$ or $\vec{\mathscr{Q}}_{i+1}$ vanishes. We comment below on this situation, but note that breakdown is exceedingly rare.

Theorem 4. Generically, Algorithm 5 determines the decompositions

$$\mathscr{A} \star \mathscr{P}_m = \mathscr{Q}_m \star \mathscr{B}_m, \tag{11}$$

$$\mathscr{A}^H \star \mathscr{Q}_m = \mathscr{P}_m \star \mathscr{B}_m^H + \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H, \tag{12}$$

with $\mathscr{P}_m \in \mathbb{K}_n^{p\times m}$, $\mathscr{Q}_m \in \mathbb{K}_n^{\ell\times m}$, where $\mathscr{P}_m^H \star \mathscr{P}_m = \mathscr{I}_m$ and $\mathscr{Q}_m^H \star \mathscr{Q}_m = \mathscr{I}_m$. The tensor $\vec{\mathscr{E}}_m \in \mathbb{K}_n^m$ is the canonical lateral slice whose elements are zero except for the first element of the mth tube, which equals 1, and $\vec{\mathscr{R}}_m \in \mathbb{K}_n^p$ is determined by steps 6 and 7 of Algorithm 5 such that $\mathscr{P}_m^H \star \vec{\mathscr{R}}_m = 0$. The tensor $\mathscr{B}_m \in \mathbb{K}_n^{m\times m}$ is upper bidiagonal, each of whose frontal slices is an upper bidiagonal matrix. Thus,

$$\mathscr{B}_m = \begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\beta}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\alpha}_2 & \boldsymbol{\beta}_2 & \mathbf{0} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \boldsymbol{\alpha}_{m-1} & \boldsymbol{\beta}_{m-1} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \boldsymbol{\alpha}_m \end{bmatrix},$$

where $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ are tubes in $\mathbb{K}_n$.

Proof. The relations (11) and (12) follow immediately from the recursion relations of Algorithm 5. The orthonormality of the lateral slices of $\mathscr{P}_m$ and $\mathscr{Q}_m$ can be shown by induction. The proof is closely related to the proof of the existence of the relations (9) and (10), and the properties of the matrices involved. The latter relations are used in[2]. □

The Lanczos bidiagonalization process may suffer from loss of orthogonality of the lateral slices of the tensors $\mathscr{P}_m$ and $\mathscr{Q}_m$. Therefore, reorthogonalization is carried out in Lines 7 and 12 in Algorithm 5. We remark that reorthogonalization makes the algorithm more costly both in terms of storage and arithmetic floating point operations. The extra cost may be acceptable as long as the number of steps m is fairly small; see[2,27,28] for discussions of the matrix case.

Let $\vec{\mathscr{R}}_m$ be the tensor whose lateral slices are defined in Line 5. Then

$$[\vec{\mathscr{P}}_{m+1}, \boldsymbol{\beta}_m] = \texttt{Normalize}\left(\vec{\mathscr{R}}_m\right). \tag{13}$$

In the rare event that some $\boldsymbol{\beta}_j$, $1 \le j < m$, vanishes $\left(\boldsymbol{\beta}_j = \mathbf{0}\right)$, Algorithm 5 breaks down. Then the singular tubes of $\mathscr{B}_j$ are singular tubes of $\mathscr{A}$, and the left and right lateral tensor singular slices are obtained as described below. When no breakdown takes place, we can express equation (12) as

$$\mathscr{A}^H \star \mathscr{Q}_m = \mathscr{P}_{m+1} \star \mathscr{B}_{m,m+1}^H,$$

where $\mathscr{P}_{m+1}$ is obtained from $\mathscr{P}_m$ by appending the lateral slice $\vec{\mathscr{P}}_{m+1}$, defined in (13), to get $\mathscr{P}_{m+1} = \left[\mathscr{P}_m, \vec{\mathscr{P}}_{m+1}\right] \in \mathbb{K}_n^{p\times(m+1)}$, and $\mathscr{B}_{m,m+1} \in \mathbb{K}_n^{m\times(m+1)}$ is obtained by appending the lateral slice $\boldsymbol{\beta}_m \star \vec{\mathscr{E}}_m$ to $\mathscr{B}_m$, i.e., $\mathscr{B}_{m,m+1} = \left[\mathscr{B}_m, \boldsymbol{\beta}_m \star \vec{\mathscr{E}}_m\right]$.

We turn to the connection between the partial Lanczos bidiagonalization of a third-order tensor $\mathscr{A}$ and the partial Lanczos tridiagonalization process of the tensor $\mathscr{A}^H \star \mathscr{A}$. This connection will be used later. Multiplying (11) from the left by $\mathscr{A}^H$, we get

$$\begin{aligned} \mathscr{A}^H \star \mathscr{A} \star \mathscr{P}_m &= \mathscr{A}^H \star \mathscr{Q}_m \star \mathscr{B}_m \\ &= \mathscr{P}_m \star \mathscr{B}_m^H \star \mathscr{B}_m + \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \mathscr{B}_m \\ &= \mathscr{P}_m \star \mathscr{B}_m^H \star \mathscr{B}_m + \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \boldsymbol{\alpha}_m. \end{aligned} \tag{14}$$

Let $\mathscr{T}_m$ be the symmetric tridiagonal tensor defined by

$$\mathscr{T}_m = \mathscr{B}_m^H \star \mathscr{B}_m \in \mathbb{K}_n^{m\times m}.$$

Then (14) is a partial tensor Lanczos tridiagonalization of $\mathscr{A}^H \star \mathscr{A}$ with initial lateral slice $\vec{\mathscr{P}}_1 = \mathscr{P}_m \star \vec{\mathscr{E}}_1$. The lateral slices of $\mathscr{P}_m$ form an orthonormal basis for the tensor Krylov subspace

$$\mathscr{K}_m\left(\mathscr{A}^H \star \mathscr{A}, \vec{\mathscr{P}}_1\right) = \texttt{span}\left\{\vec{\mathscr{P}}_1, \mathscr{A}^H \star \mathscr{A} \star \vec{\mathscr{P}}_1, \left(\mathscr{A}^H \star \mathscr{A}\right)^2 \star \vec{\mathscr{P}}_1, \dots, \left(\mathscr{A}^H \star \mathscr{A}\right)^{m-1} \star \vec{\mathscr{P}}_1\right\}.$$

Similarly, multiplying (12) from the left by $\mathscr{A}$, we obtain

$$\mathscr{A} \star \mathscr{A}^H \star \mathscr{Q}_m = \mathscr{Q}_m \star \mathscr{B}_m \star \mathscr{B}_m^H + \mathscr{A} \star \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H.$$

It follows that the lateral slices of $\mathscr{Q}_m$ form an orthonormal basis for the Krylov subspace

$$\mathscr{K}_m\left(\mathscr{A}\star\mathscr{A}^H,\vec{\mathscr{Q}}_1\right)=\mathtt{span}\left\{\vec{\mathscr{Q}}_1,\mathscr{A}\star\mathscr{A}^H\star\vec{\mathscr{Q}}_1,\left(\mathscr{A}\star\mathscr{A}^H\right)^2\star\vec{\mathscr{Q}}_1,\ldots,\left(\mathscr{A}\star\mathscr{A}^H\right)^{m-1}\star\vec{\mathscr{Q}}_1\right\}.$$

3.2 | Approximating singular tubes and singular lateral slices

We describe an approach to approximate the largest or smallest singular triplets (singular tubes and associated left and right lateral singular slices) of a large tensor $\mathscr{A}\in\mathbb{K}_n^{\ell\times p}$, $(\ell\geq p)$, using restarted partial tensor Lanczos bidiagonalization. Since the tensor $\mathscr{A}$ is large, computing its k largest or smallest singular triplets by determining the t-SVD of $\mathscr{A}$ is very expensive. The idea is to approximate the extreme singular triplets of the tensor $\mathscr{A}$ by determining the extreme singular triplets of the bidiagonal tensor $\mathscr{B}_m$, where m is small. Let $\{\boldsymbol{s}_{i,m},\vec{\mathscr{U}}_{i,m},\vec{\mathscr{V}}_{i,m}\}$, $1\leq i\leq m$, denote the singular triplets of $\mathscr{B}_m$. They satisfy

$$\mathscr{B}_m\star\vec{\mathscr{V}}_{i,m}=\boldsymbol{s}_{i,m}\star\vec{\mathscr{U}}_{i,m}\ \text{ and }\ \mathscr{B}_m^H\star\vec{\mathscr{U}}_{i,m}=\boldsymbol{s}_{i,m}\star\vec{\mathscr{V}}_{i,m}.$$

In the following, to simplify the notation, we will use $\{\boldsymbol{s}_i,\vec{\mathscr{U}}_i,\vec{\mathscr{V}}_i\}$ instead of $\{\boldsymbol{s}_{i,m},\vec{\mathscr{U}}_{i,m},\vec{\mathscr{V}}_{i,m}\}$ as a singular triplet of the tensor $\mathscr{B}_m$.

The $k\leq m$ largest singular triplets of $\mathscr{A}$ are approximated by the triplets $\{\boldsymbol{s}_{i,m}^{\mathscr{A}},\vec{\mathscr{U}}_{i,m}^{\mathscr{A}},\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}\}$ defined by

$$\boldsymbol{s}_{i,m}^{\mathscr{A}}=\boldsymbol{s}_i,\quad\vec{\mathscr{U}}_{i,m}^{\mathscr{A}}=\mathscr{Q}_m\star\vec{\mathscr{U}}_i,\quad\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}=\mathscr{P}_m\star\vec{\mathscr{V}}_i,\quad i=1,2,\ldots,k.\tag{15}$$

For $i=1,2,\ldots,k$, we have

$$\begin{aligned}\mathscr{A}\star\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}&=\mathscr{A}\star\mathscr{P}_m\star\vec{\mathscr{V}}_i\\&=\mathscr{Q}_m\star\mathscr{B}_m\star\vec{\mathscr{V}}_i\\&=\mathscr{Q}_m\star\boldsymbol{s}_i\star\vec{\mathscr{U}}_i\\&=\mathscr{Q}_m\star\vec{\mathscr{U}}_i\star\boldsymbol{s}_i\\&=\vec{\mathscr{U}}_{i,m}^{\mathscr{A}}\star\boldsymbol{s}_{i,m}^{\mathscr{A}}.\end{aligned}$$

Similarly,

$$\begin{aligned}\mathscr{A}^H\star\vec{\mathscr{U}}_{i,m}^{\mathscr{A}}=\mathscr{A}^H\star\mathscr{Q}_m\star\vec{\mathscr{U}}_i&=\left(\mathscr{P}_m\star\mathscr{B}_m+\vec{\mathscr{R}}_m\star\vec{\mathscr{E}}_m^H\right)\star\vec{\mathscr{U}}_i\\&=\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}\star\boldsymbol{s}_{i,m}^{\mathscr{A}}+\vec{\mathscr{R}}_m\star\vec{\mathscr{E}}_m^H\star\vec{\mathscr{U}}_i.\end{aligned}\tag{16}$$

To accept $\{\boldsymbol{s}_{i,m}^{\mathscr{A}},\vec{\mathscr{U}}_{i,m}^{\mathscr{A}},\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}\}$ as an approximate singular triplet of $\mathscr{A}$, the remainder term $\vec{\mathscr{R}}_m\star\vec{\mathscr{E}}_m^H\star\vec{\mathscr{U}}_i$ should be small enough. We can bound the remainder term according to

$$\begin{aligned}\left\|\vec{\mathscr{R}}_m\star\vec{\mathscr{E}}_m^H\star\vec{\mathscr{U}}_i\right\|_F&=\frac{1}{\sqrt{n}}\left\|\mathtt{bdiag}\left(\widehat{\vec{\mathscr{R}}_m}\right)\mathtt{bdiag}\left(\widehat{\left(\vec{\mathscr{E}}_m^H\right)}\right)\mathtt{bdiag}\left(\widehat{\vec{\mathscr{U}}_i}\right)\right\|_F\\&\leq\frac{1}{\sqrt{n}}\left\|\mathtt{bdiag}\left(\widehat{\vec{\mathscr{R}}_m}\right)\right\|_F\left\|\mathtt{bdiag}\left(\widehat{\left(\vec{\mathscr{E}}_m^H\right)}\right)\mathtt{bdiag}\left(\widehat{\vec{\mathscr{U}}_i}\right)\right\|_F\\&=\left\|\mathtt{bdiag}\left(\vec{\mathscr{R}}_m\right)\right\|_F\left\|\mathtt{bdiag}\left(\widehat{\left(\vec{\mathscr{E}}_m^H\right)}\right)\mathtt{bdiag}\left(\widehat{\vec{\mathscr{U}}_i}\right)\right\|_F\\&=\left\|\boldsymbol{\beta}_m\right\|_F\sum_{s=1}^{n}\left|\widehat{\left(\vec{\mathscr{E}}_m^H\right)}^{(s)}\widehat{\vec{\mathscr{U}}_i}^{(s)}\right|.\end{aligned}$$

Analogously as in[2], we require for $1\leq s\leq n$ that

$$\left|\widehat{\left(\vec{\mathscr{E}}_m^H\right)}^{(s)}\widehat{\vec{\mathscr{U}}_i}^{(s)}\right|\leq\delta'\left\|\widehat{\mathscr{A}}^{(s)}\right\|=\delta'\left(\boldsymbol{s}_{1,m}^{\widehat{\mathscr{A}}^{(s)}}\right)=\delta\left(\boldsymbol{s}_{1,m}^{\widehat{\mathscr{A}}}\right)^{(s)},$$

for a user-chosen parameter $\delta'>0$, where $\left(\boldsymbol{s}_{j,m}^{\widehat{\mathscr{A}}}\right)^{(s)}$ denotes the sth element of the jth approximate singular tube of $\widehat{\mathscr{A}}$. We obtain from eq. (4) that

$$\left\|\vec{\mathscr{R}}_m\star\vec{\mathscr{E}}_m^H\star\vec{\mathscr{U}}_i\right\|_F\leq\delta'\left\|\boldsymbol{\beta}_m\right\|_F\sum_{s=1}^{n}\left(\boldsymbol{s}_1^{\widehat{\mathscr{A}}}\right)^{(s)}=n\delta'\left\|\boldsymbol{\beta}_m\right\|_F\left(\boldsymbol{s}_1^{\mathscr{A}}\right)^{(1)}=n\delta''\left(\boldsymbol{s}_1^{\mathscr{A}}\right)^{(1)},$$

where $\delta'' = \delta' \left\|\boldsymbol{\beta}_m\right\|_F$. The computed approximate singular triplets $\{\boldsymbol{s}_{i,m}^{\mathscr{A}}, \vec{\mathscr{U}}_{i,m}^{\mathscr{A}}, \vec{\mathscr{V}}_{i,m}^{\mathscr{A}}\}$, $i = 1, 2, \ldots, k$, of $\mathscr{A}$ are accepted as singular triplets of $\mathscr{A}$ if

$$\left\|\vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\mathscr{U}}_i\right\|_F \leq \delta \left(\boldsymbol{s}_{1,m}^{\mathscr{A}}\right)^{(1)}, \quad i = 1, 2, \ldots k, \tag{17}$$

for some user-specified parameter $\delta > 0$.

To keep the storage requirement fairly small for large-scale problems, we would like the number of steps m of the tensor Lanczos bidiagonalization process to be small. However, when m is small, it may not be possible to approximate the desired singular triplets sufficiently accurately using the available Krylov subspaces $\mathscr{K}_m\left(\mathscr{A}^H \star \mathscr{A}, \vec{\mathscr{Q}}_1\right)$ and $\mathscr{K}_m\left(\mathscr{A} \star \mathscr{A}^H, \vec{\mathscr{P}}_1\right)$. A remedy for this situation is to restart the tensor Lanczos bidiagonalization process. The idea is to repeatedly update the initial lateral slices used for the tensor Lanczos bidiagonalization process, and in this way determine a sequence of increasingly more appropriate Krylov subspaces, until the k desired singular triplets have been found with required accuracy. We remark that restarting techniques have been used for computing a few desired singular triplets or eigenvalue-eigenvector pairs of a large matrix, where properties of Ritz vectors, harmonic Ritz vectors, and refined Ritz vectors have been exploited; see, e.g.,[2,23,29,30,31] for details.

3.3 | Augmentation by Ritz lateral slices

Assume that we would like to approximate the k largest singular triplets of $\mathscr{A} \in \mathbb{K}_n^{\ell \times p}$. To this end, we carry out $m > k$ steps of tensor Lanczos bidiagonalization as described in the previous subsection. The approximate right singular lateral slice $\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}$ is a Ritz lateral slice of $\mathscr{A}^H \star \mathscr{A}$ associated with the Ritz tube $\left(\boldsymbol{s}_{i,m}^{\mathscr{A}}\right)^2 = \boldsymbol{s}_{i,m}^{\mathscr{A}} \star \boldsymbol{s}_{i,m}^{\mathscr{A}}$ for $i \in \{1, 2, \ldots, m\}$, and we have

$$\begin{aligned}\mathscr{A}^H \star \mathscr{A} \star \vec{\mathscr{V}}_{i,m}^{\mathscr{A}} = \mathscr{A}^H \star \vec{\mathscr{U}}_{i,m}^{\mathscr{A}} \star \boldsymbol{s}_{i,m}^{\mathscr{A}} &= \left(\vec{\mathscr{V}}_{i,m}^{\mathscr{A}} \star \boldsymbol{s}_{i,m}^{\mathscr{A}} + \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\mathscr{U}}_i\right) \star \boldsymbol{s}_{i,m}^{\mathscr{A}} \\ &= \vec{\mathscr{V}}_{i,m}^{\mathscr{A}} \star \left(\boldsymbol{s}_{i,m}^{\mathscr{A}}\right)^2 + \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\mathscr{U}}_i \star \boldsymbol{s}_{i,m}^{\mathscr{A}}.\end{aligned}$$

In what follows we will show some results that will help us to approximate the largest or smallest singular triplets of a third-order tensor. The idea behind these results is to find equations that are analogous to (11) and (12), and such that the reduced tensor will contain the k approximate singular tubes among its first k elements on the diagonal, and the right projection tensor will contain the k right Ritz lateral slices among its first k lateral slices, and the left projection tensor will contain the k left Ritz lateral slices among its first k lateral slices. The following theorem will be helpful.

Theorem 5. Assume that m steps of Algorithm 5 have been applied to the third-order tensor $\mathscr{A} \in \mathbb{K}_n^{\ell \times p}$, and suppose that $\boldsymbol{\beta}_m$ in (13) is nonvanishing. Then for $k < m$, we have

$$\mathscr{A} \star \widetilde{\mathscr{P}}_{k+1} = \widetilde{\mathscr{Q}}_{k+1} \star \widetilde{\mathscr{B}}_{k+1}, \tag{18}$$

$$\mathscr{A}^H \star \widetilde{\mathscr{Q}}_{k+1} = \widetilde{\mathscr{P}}_{k+1} \star \widetilde{\mathscr{B}}_{k+1}^H + \widetilde{\boldsymbol{\beta}}_{k+1} \star \vec{\widetilde{\mathscr{P}}}_{k+2} \star \vec{\mathscr{E}}_{k+1}^H, \tag{19}$$

where $\widetilde{\mathscr{P}}_{k+1} \in \mathbb{K}_n^{p \times (k+1)}$ and $\widetilde{\mathscr{Q}}_{k+1} \in \mathbb{K}_n^{\ell \times (k+1)}$ have orthonormal lateral slices, and the first k lateral slices of $\widetilde{\mathscr{P}}_m$ are the first k Ritz lateral slices of $\mathscr{A}$, $\widetilde{\mathscr{B}}_{k+1} \in \mathbb{K}_n^{(k+1) \times (k+1)}$ is an upper triangular tensor, $\vec{\widetilde{\mathscr{P}}}_{k+2} \in \mathbb{K}_n^p$ is a lateral slice that is orthogonal to $\widetilde{\mathscr{P}}_{k+1}$, $\widetilde{\boldsymbol{\beta}}_{k+1} \in \mathbb{K}_n$, and $\vec{\mathscr{E}}_{k+1} \in \mathbb{K}_n^{k+1}$ is the canonical element under the t-product.

Proof. Let the Ritz lateral slices $\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}$ for $1 \leq i \leq k$ be associated with the k Ritz tubes of $\mathscr{A}$. Introduce the tensor

$$\widetilde{\mathscr{P}}_{k+1} = \left[\vec{\mathscr{V}}_{1,m}^{\mathscr{A}}, \vec{\mathscr{V}}_{2,m}^{\mathscr{A}}, \ldots, \vec{\mathscr{V}}_{k,m}^{\mathscr{A}}, \vec{\mathscr{P}}_{m+1}\right] \in \mathbb{K}_n^{p \times (k+1)}, \tag{20}$$

where $\vec{\mathscr{P}}_{m+1}$ is given by (13). Then, using the fact that $\mathscr{A} \star \vec{\mathscr{V}}_{i,m}^{\mathscr{A}} = \vec{\mathscr{U}}_{i,m}^{\mathscr{A}} \star \boldsymbol{s}_{i,m}^{\mathscr{A}}$ for $i = 1, 2, \ldots, k$, we obtain

$$\begin{aligned}\mathscr{A} \star \widetilde{\mathscr{P}}_{k+1} &= \left[\mathscr{A} \star \vec{\mathscr{V}}_{1,m}^{\mathscr{A}}, \mathscr{A} \star \vec{\mathscr{V}}_{2,m}^{\mathscr{A}}, \ldots, \mathscr{A} \star \vec{\mathscr{V}}_{k,m}^{\mathscr{A}}, \mathscr{A} \star \vec{\mathscr{P}}_{m+1}\right] \\ &= \left[\vec{\mathscr{U}}_{1,m}^{\mathscr{A}} \star \boldsymbol{s}_{1,m}^{\mathscr{A}}, \vec{\mathscr{U}}_{2,m}^{\mathscr{A}} \star \boldsymbol{s}_{2,m}^{\mathscr{A}}, \ldots, \vec{\mathscr{U}}_{k,m}^{\mathscr{A}} \star \boldsymbol{s}_{k,m}^{\mathscr{A}}, \mathscr{A} \star \vec{\mathscr{P}}_{m+1}\right].\end{aligned} \tag{21}$$

Orthogonalizing the expression $\mathscr{A} \star \vec{\mathscr{P}}_{m+1}$ against $\{\vec{\mathscr{U}}_{i,m}^{\mathscr{A}}\}_{i=1:k}$ gives

$$\mathscr{A} \star \vec{\mathscr{P}}_{m+1} = \sum_{i=1}^{k} \boldsymbol{\rho}_i \star \vec{\mathscr{U}}_{i,m}^{\mathscr{A}} + \vec{\widetilde{\mathscr{R}}}_k, \tag{22}$$

where $\vec{\widetilde{\mathscr{R}}}_k$ is orthogonal to $\{\vec{\mathscr{U}}^{\mathscr{A}}_{i,m}\}_{i=1:k}$, and the $\boldsymbol{\rho}_i$ for $i \in \{1, 2, \ldots, k\}$ are given by

$$\begin{aligned}
\boldsymbol{\rho}_i = \left(\vec{\mathscr{U}}^{\mathscr{A}}_{i,m}\right)^H \star \left(\mathscr{A} \star \vec{\mathscr{P}}_{m+1}\right) &= \left(\mathscr{A}^H \star \vec{\mathscr{U}}^{\mathscr{A}}_{i,m}\right)^H \star \vec{\mathscr{P}}_{m+1} \\
&= \left(\vec{\mathscr{V}}^{\mathscr{A}}_{i,m} \star \boldsymbol{s}^{\mathscr{A}}_{i,m} + \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}^H_m \star \vec{\mathscr{U}}_i\right)^H \star \vec{\mathscr{P}}_{m+1} \\
&= \boldsymbol{\beta}^H_m \star \left(\vec{\mathscr{U}}_i{}^H \star \vec{\mathscr{E}}_m \star \vec{\mathscr{P}}^H_{m+1}\right) \star \vec{\mathscr{P}}_{m+1} \\
&= \boldsymbol{\beta}_m \star \vec{\mathscr{U}}_i{}^H \star \vec{\mathscr{E}}_m \\
&= \boldsymbol{\beta}_m \star \left\langle \vec{\mathscr{U}}_i, \vec{\mathscr{E}}_m \right\rangle,
\end{aligned}$$

because $\boldsymbol{\beta}_m = \boldsymbol{\beta}^H_m$.

Let $\vec{\widetilde{\mathscr{R}}}_k = \vec{\widetilde{\mathscr{R}}'}_k \star \widetilde{\boldsymbol{\alpha}}_{k+1}$ be a normalization of $\vec{\widetilde{\mathscr{R}}}_k$, and introduce the tensors

$$\widetilde{\mathscr{Q}}_{k+1} = \left[\vec{\mathscr{U}}^{\mathscr{A}}_{1,m}, \vec{\mathscr{U}}^{\mathscr{A}}_{2,m}, \ldots, \vec{\mathscr{U}}^{\mathscr{A}}_{k,m}, \vec{\widetilde{\mathscr{R}}'}_k\right] \in \mathbb{K}^{\ell \times (k+1)}_n \tag{23}$$

and

$$\widetilde{\mathscr{B}}_{k+1} = \begin{bmatrix}
\boldsymbol{s}^{\mathscr{A}}_{1,m} & \mathbf{0} & \ldots & \mathbf{0} & \boldsymbol{\rho}_1 \\
\mathbf{0} & \boldsymbol{s}^{\mathscr{A}}_{2,m} & \ldots & \mathbf{0} & \boldsymbol{\rho}_2 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\mathbf{0} & \ldots & \mathbf{0} & \boldsymbol{s}^{\mathscr{A}}_{k,m} & \boldsymbol{\rho}_k \\
\mathbf{0} & \ldots & \ldots & \mathbf{0} & \widetilde{\boldsymbol{\alpha}}_{k+1}
\end{bmatrix} \in \mathbb{K}^{(k+1)\times(k+1)}_n. \tag{24}$$

Then, from (21) and (22), we obtain

$$\begin{aligned}
\mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+1} &= \left[\vec{\mathscr{U}}^{\mathscr{A}}_{1,m} \star \boldsymbol{s}^{\mathscr{A}}_{1,m}, \vec{\mathscr{U}}^{\mathscr{A}}_{2,m} \star \boldsymbol{s}^{\mathscr{A}}_{2,m}, \ldots, \vec{\mathscr{U}}^{\mathscr{A}}_{k,m} \star \boldsymbol{s}^{\mathscr{A}}_{k,m}, \sum_{i=1}^{k} \boldsymbol{\rho}_i \star \vec{\mathscr{U}}^{\mathscr{A}}_{i,m} + \vec{\widetilde{\mathscr{R}}}_k\right] \\
&= \widetilde{\mathscr{Q}}_{k+1} \star \widetilde{\mathscr{B}}_{k+1}.
\end{aligned} \tag{25}$$

On the other hand, as

$$\mathscr{A}^H \star \widetilde{\mathscr{Q}}_{k+1} = \left[\mathscr{A}^H \star \vec{\mathscr{U}}^{\mathscr{A}}_{1,m}, \mathscr{A}^H \star \vec{\mathscr{U}}^{\mathscr{A}}_{2,m}, \ldots, \mathscr{A}^H \star \vec{\mathscr{U}}^{\mathscr{A}}_{k,m}, \mathscr{A}^H \star \vec{\widetilde{\mathscr{R}}'}_k\right],$$

using (16), we get

$$\begin{aligned}
\mathscr{A}^H \star \vec{\mathscr{U}}^{\mathscr{A}}_{i,m} &= \vec{\mathscr{V}}^{\mathscr{A}}_{i,m} \star \boldsymbol{s}^{\mathscr{A}}_{i,m} + \vec{\mathscr{R}}_m \star \vec{\mathscr{E}}^H_m \star \vec{\mathscr{U}}_i \\
&= \vec{\mathscr{V}}^{\mathscr{A}}_{i,m} \star \boldsymbol{s}^{\mathscr{A}}_{i,m} + \vec{\mathscr{P}}_{m+1} \star \boldsymbol{\beta}_m \star \vec{\mathscr{E}}^H_m \star \vec{\mathscr{U}}_i \\
&= \vec{\mathscr{V}}^{\mathscr{A}}_{i,m} \star \boldsymbol{s}^{\mathscr{A}}_{i,m} + \vec{\mathscr{P}}_{m+1} \star \boldsymbol{\rho}^H_i.
\end{aligned}$$

Since

$$\left\langle \mathscr{A}^H \star \vec{\widetilde{\mathscr{R}}}'_k, \vec{\mathscr{V}}^{\mathscr{A}}_{i,m} \right\rangle = \left(\vec{\widetilde{\mathscr{R}}}'_k\right)^H \star \mathscr{A} \star \vec{\mathscr{V}}^{\mathscr{A}}_{i,m} = \boldsymbol{s}^{\mathscr{A}}_{i,m} \star \left(\vec{\widetilde{\mathscr{R}}}'_k\right)^H \star \vec{\mathscr{U}}^{\mathscr{A}}_{i,m} = \mathbf{0},$$

the tensor $\mathscr{A}^H \star \vec{\widetilde{\mathscr{R}}'}_k$ is orthogonal to $\vec{\mathscr{V}}^{\mathscr{A}}_{i,m}$. Moreover, in view of that $\vec{\mathscr{V}}^{\mathscr{A}}_{i,m}$ is orthogonal to $\vec{\mathscr{P}}_{m+1}$, we obtain

$$\mathscr{A}^H \star \vec{\widetilde{\mathscr{R}}'}_k = \boldsymbol{\gamma} \star \vec{\mathscr{P}}_{m+1} + \vec{\mathscr{F}}_{k+1}, \tag{26}$$

where $\vec{\mathscr{F}}_{k+1}$ is orthogonal to $\vec{\mathscr{P}}_{m+1}$ as well as to $\vec{\mathscr{V}}^{\mathscr{A}}_{i,m}$. Due to the orthogonality of $\vec{\widetilde{\mathscr{R}}}_k$ (or $\vec{\widetilde{\mathscr{R}}'}_k$) to $\left\{\vec{\mathscr{U}}^{\mathscr{A}}_{i,m}\right\}_{i=1:k}$, the parameter $\boldsymbol{\gamma}$ in (26) is given by

$$\begin{aligned}
\boldsymbol{\gamma} = \left\langle \vec{\mathscr{P}}_{m+1}, \mathscr{A}^H \star \vec{\widetilde{\mathscr{R}}'}_k \right\rangle &= \left\langle \mathscr{A} \star \vec{\mathscr{P}}_{m+1}, \vec{\widetilde{\mathscr{R}}'}_k \right\rangle \\
&= \left\langle \sum_{i=1}^{k} \boldsymbol{\rho}_i \star \vec{\mathscr{U}}^{\mathscr{A}}_{i,m} + \vec{\widetilde{\mathscr{R}}}_k, \vec{\widetilde{\mathscr{R}}'}_k \right\rangle \\
&= \left\langle \vec{\widetilde{\mathscr{R}}}_k, \vec{\widetilde{\mathscr{R}}'}_k \right\rangle = \widetilde{\boldsymbol{\alpha}}_{k+1}.
\end{aligned}$$

Consequently,

$$\begin{aligned}\mathscr{A}^H \star \widetilde{\mathscr{Q}}_{k+1} &= \left[\vec{\mathscr{V}}_{1,m}^{\mathscr{A}} \star \boldsymbol{s}_{1,m}^{\mathscr{A}} + \vec{\mathscr{P}}_{m+1} \star \boldsymbol{\rho}_1^H, \ldots, \vec{\mathscr{V}}_{k,m}^{\mathscr{A}} \star \boldsymbol{s}_{k,m}^{\mathscr{A}} + \vec{\mathscr{P}}_{m+1} \star \boldsymbol{\rho}_k^H, \widetilde{\boldsymbol{\alpha}}_{k+1} \star \vec{\mathscr{P}}_{m+1} + \vec{\mathscr{F}}_{k+1}\right] \\ &= \widetilde{\mathscr{P}}_{k+1} \star \widetilde{\mathscr{B}}_{k+1}^H + \vec{\mathscr{F}}_{k+1} \star \vec{\mathscr{E}}_{k+1}^H \\ &= \widetilde{\mathscr{P}}_{k+1} \star \widetilde{\mathscr{B}}_{k+1}^H + \widetilde{\boldsymbol{\beta}}_{k+1} \star \vec{\widetilde{\mathscr{P}}}_{k+2} \star \vec{\mathscr{E}}_{k+1}^H, \end{aligned} \tag{27}$$

where $\widetilde{\boldsymbol{\beta}}_{k+1}$ and $\vec{\widetilde{\mathscr{P}}}_{k+2}$ are determined by the normalization of $\vec{\mathscr{F}}_{k+1}$, i.e., $\vec{\mathscr{F}}_{k+1} = \widetilde{\boldsymbol{\beta}}_{k+1} \star \vec{\widetilde{\mathscr{P}}}_{k+2}$, because

$$\widetilde{\mathscr{B}}_{k+1}^H = \begin{bmatrix} \boldsymbol{s}_{1,m}^{\mathscr{A}} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{s}_{2,m}^{\mathscr{A}} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \boldsymbol{s}_{k,m}^{\mathscr{A}} & \mathbf{0} \\ \boldsymbol{\rho}_1^H & \boldsymbol{\rho}_2^H & \ldots & \boldsymbol{\rho}_k^H & \widetilde{\boldsymbol{\alpha}}_{k+1} \end{bmatrix} \in \mathbb{K}_n^{(k+1)\times(k+1)}.$$

The orthogonality of $\widetilde{\mathscr{P}}_{k+1}$ and $\widetilde{\mathscr{Q}}_{k+1}$ now follows from the orthogonality of the sequences $\left\{\vec{\mathscr{V}}_{i,m}^{\mathscr{A}}\right\}_{i=1:k}$ and $\left\{\vec{\mathscr{U}}_{i,m}^{\mathscr{A}}\right\}_{i=1:k}$, respectively, given by (15). □

In the preceding theorem we assumed $\boldsymbol{\beta}_m$ to be nonvanishing. If, instead, $\boldsymbol{\beta}_m$ vanishes, then the singular tubes of $\mathscr{B}_m$ are singular tubes of $\mathscr{A}$, and the left and right singular lateral slices of $\mathscr{A}$ can be determined from those of $\mathscr{B}_m$. Similarly, if $\widetilde{\boldsymbol{\beta}}_{k+1}$ in (27) vanishes, then the singular tubes of $\widetilde{\mathscr{B}}_{k+1}$ are singular tubes of $\mathscr{A}$, and the singular lateral slices of $\mathscr{A}$ can be determined from $\widetilde{\mathscr{P}}_{k+1}$ and $\widetilde{\mathscr{Q}}_{k+1}$.

If $\widetilde{\boldsymbol{\beta}}_{k+1}$ is nonvanishing, then we append new lateral slices to $\widetilde{\mathscr{P}}_{k+1}$ and $\widetilde{\mathscr{Q}}_{k+1}$ repeatedly until iteration $m-k$. This is the subject of the following theorem.

Theorem 6. Assume that m steps of Algorithm 5 have been applied to $\mathscr{A}$ and that eqs. (25) and (27) hold. If the $\widetilde{\boldsymbol{\beta}}_{k+1}$ are nonvanishing for $1 \le k < m$, then we have the following relations

$$\begin{aligned}\mathscr{A} \star \widetilde{\mathscr{P}}_m &= \widetilde{\mathscr{Q}}_m \star \widetilde{\mathscr{B}}_m, \\ \mathscr{A}^H \star \widetilde{\mathscr{Q}}_m &= \widetilde{\mathscr{P}}_m \star \widetilde{\mathscr{B}}_m^H + \widetilde{\boldsymbol{\beta}}_m \star \vec{\widetilde{\mathscr{P}}}_{m+1} \star \vec{\mathscr{E}}_m^H, \end{aligned}$$

where $\widetilde{\mathscr{P}}_m \in \mathbb{K}_n^{p\times m}$ and $\widetilde{\mathscr{Q}}_m \in \mathbb{K}_n^{\ell\times m}$ have orthonormal lateral slices, $\widetilde{\mathscr{B}}_m \in \mathbb{K}_n^{m\times m}$ is f-upper triangular, $\widetilde{\boldsymbol{\beta}}_m \in \mathbb{K}_n$, $\vec{\widetilde{\mathscr{P}}}_{m+1} \in \mathbb{K}_n^p$ is orthogonal to $\widetilde{\mathscr{P}}_m$, and $\vec{\mathscr{E}}_m \in \mathbb{K}_n^m$ is the canonical element under the t-product. The first k lateral slices of $\widetilde{\mathscr{P}}_m$ and $\widetilde{\mathscr{Q}}_m$ are the same as those of the tensors $\widetilde{\mathscr{P}}_{k+1}$ and $\widetilde{\mathscr{Q}}_{k+1}$, respectively, given in Theorem 5.

Proof. Let the tensors $\widetilde{\mathscr{P}}_{k+1}$ and $\widetilde{\mathscr{Q}}_{k+1}$ defined in (25) and (27), respectively, be represented by

$$\widetilde{\mathscr{P}}_{k+1} = \left[\vec{\widetilde{\mathscr{P}}}_1, \vec{\widetilde{\mathscr{P}}}_2, \ldots, \vec{\widetilde{\mathscr{P}}}_{k+1}\right] \in \mathbb{K}_n^{p\times(k+1)}$$

and

$$\widetilde{\mathscr{Q}}_{k+1} = \left[\vec{\widetilde{\mathscr{Q}}}_1, \vec{\widetilde{\mathscr{Q}}}_2, \ldots, \vec{\widetilde{\mathscr{Q}}}_{k+1}\right] \in \mathbb{K}_n^{\ell\times(k+1)},$$

and the tensor $\widetilde{\mathscr{P}}_{k+2}$ be given by

$$\widetilde{\mathscr{P}}_{k+2} = \left[\widetilde{\mathscr{P}}_{k+1}, \vec{\widetilde{\mathscr{P}}}_{k+2}\right] \in \mathbb{K}_n^{p\times(k+2)}.$$

By normalizing the quantity $\left(\mathscr{I}_\ell - \widetilde{\mathscr{Q}}_{k+1} \star \widetilde{\mathscr{Q}}_{k+1}^H\right) \star \mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+2}$, we obtain the lateral slice $\vec{\widetilde{\mathscr{Q}}}_{k+2}$ such that $\widetilde{\boldsymbol{\alpha}}_{k+2} \star \vec{\widetilde{\mathscr{Q}}}_{k+2} = \left(\mathscr{I}_\ell - \widetilde{\mathscr{Q}}_{k+1} \star \widetilde{\mathscr{Q}}_{k+1}^H\right) \star \mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+2}$. Application of (19) gives

$$\begin{aligned}\widetilde{\boldsymbol{\alpha}}_{k+2} \star \vec{\widetilde{\mathscr{Q}}}_{k+2} &= \left(\mathscr{I}_\ell - \widetilde{\mathscr{Q}}_{k+1} \star \widetilde{\mathscr{Q}}_{k+1}^H\right) \star \mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+2} \\ &= \mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+2} - \widetilde{\mathscr{Q}}_{k+1} \star \widetilde{\mathscr{Q}}_{k+1}^H \star \mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+2} \\ &= \mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+2} - \widetilde{\mathscr{Q}}_{k+1} \star \left(\widetilde{\mathscr{B}}_{k+1} \star \widetilde{\mathscr{P}}_{k+1}^H + \widetilde{\boldsymbol{\beta}}_{k+1} \star \vec{\mathscr{E}}_{k+1} \star \vec{\widetilde{\mathscr{P}}}_{k+2}^H\right) \star \vec{\widetilde{\mathscr{P}}}_{k+2} \\ &= \mathscr{A} \star \vec{\widetilde{\mathscr{P}}}_{k+2} - \widetilde{\boldsymbol{\beta}}_{k+1} \star \vec{\widetilde{\mathscr{Q}}}_{k+1}. \end{aligned} \tag{28}$$

Consider the tensors

$$\widetilde{\mathcal{Q}}_{k+2} = \left[\widetilde{\mathcal{Q}}_{k+1}, \vec{\widetilde{\mathcal{Q}}}_{k+2}\right] \in \mathbb{K}_n^{\ell\times(k+2)}$$

and

$$\widetilde{\mathcal{B}}_{k+2} = \begin{bmatrix} \boldsymbol{s}_{1,m}^{\mathcal{A}} & \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\rho}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{s}_{2,m}^{\mathcal{A}} & \mathbf{0} & \dots & \boldsymbol{\rho}_2 & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{s}_{k,m}^{\mathcal{A}} & \boldsymbol{\rho}_k & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \widetilde{\boldsymbol{\alpha}}_{k+1} & \widetilde{\boldsymbol{\beta}}_{k+1} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} & \widetilde{\boldsymbol{\alpha}}_{k+2} \end{bmatrix} \in \mathbb{K}_n^{(k+2)\times(k+2)}.$$

Using (18) and (28), we get

$$\mathcal{A} \star \widetilde{\mathcal{P}}_{k+2} = \widetilde{\mathcal{Q}}_{k+2} \star \widetilde{\mathcal{B}}_{k+2}.$$

To determine the lateral slice $\vec{\widetilde{\mathcal{P}}}_{k+3}$, we normalize $\left(\mathcal{I} - \widetilde{\mathcal{P}}_{k+2} \star \widetilde{\mathcal{P}}_{k+2}^H\right) \star \mathcal{A}^H \star \vec{\widetilde{\mathcal{Q}}}_{k+2}$ so that

$$\widetilde{\boldsymbol{\beta}}_{k+2} \star \vec{\widetilde{\mathcal{P}}}_{k+3} = \left(\mathcal{I} - \widetilde{\mathcal{P}}_{k+2} \star \widetilde{\mathcal{P}}_{k+2}^H\right) \star \mathcal{A}^H \star \vec{\widetilde{\mathcal{Q}}}_{k+2}$$

and

$$\widetilde{\boldsymbol{\beta}}_{k+2} \star \vec{\widetilde{\mathcal{P}}}_{k+3} = \mathcal{A}^H \star \vec{\widetilde{\mathcal{Q}}}_{k+2} - \widetilde{\boldsymbol{\alpha}}_{k+2} \star \vec{\widetilde{\mathcal{P}}}_{k+2}. \tag{29}$$

It now follows from (18) and (29) that

$$\mathcal{A}^H \star \widetilde{\mathcal{Q}}_{k+2} = \widetilde{\mathcal{P}}_{k+2} \star \widetilde{\mathcal{B}}_{k+2}^H + \widetilde{\boldsymbol{\beta}}_{k+2} \star \vec{\widetilde{\mathcal{P}}}_{k+3} \star \vec{\mathcal{E}}_{k+2}^H.$$

We can continue this procedure until iteration $m-k$ and then obtain

$$\mathcal{A} \star \widetilde{\mathcal{P}}_m = \widetilde{\mathcal{Q}}_m \star \widetilde{\mathcal{B}}_m, \quad \mathcal{A}^H \star \widetilde{\mathcal{Q}}_m = \widetilde{\mathcal{P}}_m \star \widetilde{\mathcal{B}}_m^H + \widetilde{\boldsymbol{\beta}}_m \star \vec{\widetilde{\mathcal{P}}}_{m+1} \star \vec{\mathcal{E}}_m^H,$$

where $\widetilde{\mathcal{P}}_m$ and $\widetilde{\mathcal{Q}}_m$ have orthonormal lateral slices and

$$\widetilde{\mathcal{B}}_m = \begin{bmatrix} \boldsymbol{s}_{1,m}^{\mathcal{A}} & \mathbf{0} & \dots & \boldsymbol{\rho}_1 & \mathbf{0} & \dots & \mathbf{0} \\ & \ddots & & \vdots & & & \\ & & \boldsymbol{s}_{k,m}^{\mathcal{A}} & \boldsymbol{\rho}_k & & & \\ & & & \widetilde{\boldsymbol{\alpha}}_{k+1} & \widetilde{\boldsymbol{\beta}}_{k+1} & & \\ & & & & \ddots & \ddots & \\ & & & & & \widetilde{\boldsymbol{\alpha}}_{m-1} & \widetilde{\boldsymbol{\beta}}_{m-1} \\ & & & & & & \widetilde{\boldsymbol{\alpha}}_m \end{bmatrix} \in \mathbb{K}_n^{m\times m}.$$

This gives the desired result. □

If we would like to compute the smallest singular triplets of $\mathcal{A}$, then we can use the same theorems; Theorems 5 and 7, but instead of working with the k first right singular lateral slices $\vec{\mathcal{V}}_{i,m}^{\mathcal{A}}$, $1 \le i \le k$, to construct the $\widetilde{\mathcal{P}}_{k+1}$, $\widetilde{\mathcal{Q}}_{k+1}$, $\widetilde{\mathcal{B}}_{k+1}$, $\widetilde{\boldsymbol{\beta}}_{k+1}$, and $\vec{\widetilde{\mathcal{P}}}_{k+2}$ in Equations (18) and (19), we use the k last right singular lateral slices. The computations are analogous to those described above.

3.4 | Augmentation by harmonic Ritz lateral slices

When the smallest singular values of a matrix A are clustered, their computation by the restarted Lanczos bidiagonalization method as described above may require many iterations. In this situation it may be beneficial to instead compute approximations of the smallest singular values of A by seeking to determine approximations of the largest singular values of the matrix $\left(A^TA\right)^{-1}$ without explicitly computing the matrix $\left(A^TA\right)^{-1}$. This was done for the matrix case by computing harmonic Ritz vectors; see[2,32]. Harmonic Ritz vectors furnish approximations of eigenvectors of A^TA associated with the corresponding harmonic Ritz values.

In the case of tensors, harmonic Ritz lateral slices furnish approximations of eigenvectors of $\mathscr{A}^H \star \mathscr{A}$ associated with harmonic Ritz tubes of $\mathscr{A}^H \star \mathscr{A}$. The harmonic Ritz tubes $\breve{\boldsymbol{\theta}}_j$ of $\mathscr{A}^H \star \mathscr{A}$ associated with the partial tensor tridiagonalization defined in (14) are the eigentubes of the generalized eigenvalue problem

$$\left(\left(\mathscr{B}_m^H \star \mathscr{B}_m \right)^2 + \boldsymbol{\alpha}_m^2 \star \boldsymbol{\beta}_m^2 \star \vec{\mathscr{E}}_m \star \vec{\mathscr{E}}_m^H \right) \star \vec{\breve{\omega}}_j = \breve{\boldsymbol{\theta}}_j \star \mathscr{B}_m^H \star \mathscr{B}_m \star \vec{\breve{\omega}}_j, \quad 1 \leq j \leq m. \tag{30}$$

The eigenpair $\{\breve{\boldsymbol{\theta}}_j, \vec{\breve{\omega}}_j\}$ can be computed without forming the tensor $\mathscr{B}_m^H \star \mathscr{B}_m$. Let

$$\vec{\omega}_j = \mathscr{B}_m \star \vec{\breve{\omega}}_j. \tag{31}$$

Using the relations

$$\boldsymbol{\alpha}_m \star \vec{\mathscr{E}}_m^H = \vec{\mathscr{E}}_m^H \star \mathscr{B}_m \text{ and } \boldsymbol{\alpha}_m \star \vec{\mathscr{E}}_m = \mathscr{B}_m^H \star \vec{\mathscr{E}}_m,$$

we can write

$$\boldsymbol{\alpha}_m^2 \star \boldsymbol{\beta}_m^2 \star \vec{\mathscr{E}}_m \star \vec{\mathscr{E}}_m^H = \boldsymbol{\beta}_m^2 \star \mathscr{B}_m^H \star \vec{\mathscr{E}}_m \star \vec{\mathscr{E}}_m^H \star \mathscr{B}_m.$$

Therefore, using (31), the relation (30) can be written as

$$\mathscr{B}_m^H \star \left(\mathscr{B}_m \star \mathscr{B}_m^H \star \mathscr{B}_m + \boldsymbol{\beta}_m^2 \star \vec{\mathscr{E}}_m \star \vec{\mathscr{E}}_m^H \star \mathscr{B}_m \right) \star \mathscr{B}_m^{-1} \star \vec{\omega}_j = \breve{\boldsymbol{\theta}}_j \star \mathscr{B}_m^H \star \mathscr{B}_m \star \mathscr{B}_m^{-1} \star \vec{\omega}_j.$$

It follows that

$$\left(\mathscr{B}_m \star \mathscr{B}_m^H + \boldsymbol{\beta}_m^2 \star \vec{\mathscr{E}}_m \star \vec{\mathscr{E}}_m^H \right) \star \vec{\omega}_j = \breve{\boldsymbol{\theta}}_j \star \vec{\omega}_j \tag{32}$$

and

$$\left(\mathscr{B}_m \star \mathscr{B}_m^H + \boldsymbol{\beta}_m^2 \star \vec{\mathscr{E}}_m \star \vec{\mathscr{E}}_m^H \right) = \mathscr{B}_{m,m+1} \star \mathscr{B}_{m,m+1}^H.$$

In this subsection, we denote the singular triplets of $\mathscr{B}_{m,m+1}$ by $\{\boldsymbol{s}_i', \vec{\mathscr{U}}_i', \vec{\mathscr{V}}_i'\}$ for $1 \leq i \leq m$, with the first k of them being the smallest singular triplets. Recall that we are interested in determining approximations of the smallest singular triplets of $\mathscr{A}$. The k smallest singular triplets of $\mathscr{B}_{m,m+1}$ form the tensors

$$\begin{aligned} \mathscr{U}_k' &= \left[\vec{\mathscr{U}}_1', \vec{\mathscr{U}}_2', \ldots, \vec{\mathscr{U}}_k' \right] \in \mathbb{K}_n^{m \times k}, \quad \mathscr{V}_k' = \left[\vec{\mathscr{V}}_1', \vec{\mathscr{V}}_2', \ldots, \vec{\mathscr{V}}_k' \right] \in \mathbb{K}_n^{(m+1) \times k}, \\ \mathscr{S}_k' &= \left[\boldsymbol{s}_1' \star \vec{\mathscr{E}}_1, \boldsymbol{s}_2' \star \vec{\mathscr{E}}_2, \ldots, \boldsymbol{s}_k' \star \vec{\mathscr{E}}_k \right] \in \mathbb{K}_n^{k \times k}, \end{aligned}$$

where

$$\mathscr{B}_{m,m+1} \star \mathscr{V}_k' = \mathscr{U}_k' \star \mathscr{S}_k' \text{ and } \mathscr{B}_{m,m+1}^H \star \mathscr{U}_k' = \mathscr{V}_k' \star \mathscr{S}_k'.$$

We obtain from the above equations that

$$\mathscr{B}_{m,m+1} \star \mathscr{B}_{m,m+1}^H \star \mathscr{U}_k' = \mathscr{U}_k' \star \left(\mathscr{S}_k' \right)^2,$$

where

$$\left(\mathscr{S}_k' \right)^2 = \left[\left(\boldsymbol{s}_1' \right)^2 \star \vec{\mathscr{E}}_1, \ldots, \left(\boldsymbol{s}_k' \right)^2 \star \vec{\mathscr{E}}_k \right].$$

Consequently, the eigenpair $\left\{ \left(\boldsymbol{s}_i' \right)^2, \mathscr{U}_i' \right\}$ satisfies (32), and $\left\{ \left(\boldsymbol{s}_i' \right)^2, \mathscr{B}_m^{-1} \star \mathscr{U}_i' \right\}$ is an eigenpair of (30). It follows that the harmonic Ritz lateral slice associated with $\breve{\boldsymbol{\theta}}_j$ is given by

$$\vec{\breve{\mathscr{V}}}_j = \mathscr{P}_m \star \vec{\breve{\omega}}_j = \mathscr{P}_m \star \mathscr{B}_m^{-1} \star \vec{\mathscr{U}}_j'. \tag{33}$$

We turn to the computation of the residual of harmonic Ritz lateral slices. Using eqs. (14) and (32), we obtain the relations

$$\begin{aligned} \mathscr{A}^H \star \mathscr{A} \star \vec{\breve{\mathscr{V}}}_j - \breve{\boldsymbol{\theta}}_j \star \vec{\breve{\mathscr{V}}}_j &= \mathscr{A}^H \star \mathscr{A} \star \mathscr{P}_m \star \vec{\breve{\omega}}_j - \breve{\boldsymbol{\theta}}_j \star \mathscr{P}_m \star \vec{\breve{\omega}}_j \\ &= \left(\mathscr{P}_m \star \mathscr{B}_m^H \star \mathscr{B}_m + \boldsymbol{\beta}_m \star \vec{\mathscr{E}}_m^H * \mathscr{B}_m \right) \star \vec{\breve{\omega}}_j - \breve{\boldsymbol{\theta}}_j \star \mathscr{P}_m \star \vec{\breve{\omega}}_j \\ &= \mathscr{P}_m \star \mathscr{B}_m^{-1} \star \left(\mathscr{B}_m \star \mathscr{B}_m^H - \boldsymbol{\theta}_j * \mathscr{I}_m \right) \star \vec{\omega}_j + \boldsymbol{\beta}_m \star \vec{\mathscr{P}}_{m+1} \star \vec{\mathscr{E}}_m^H \star \vec{\omega}_j \\ &= -\boldsymbol{\beta}_m^2 \star \mathscr{P}_m \star \mathscr{B}_m^{-1} * \vec{\mathscr{E}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\omega}_j + \boldsymbol{\beta}_m * \vec{\mathscr{P}}_{m+1} \star \vec{\mathscr{E}}_m^H \star \vec{\omega}_j \\ &= \vec{\mathscr{E}}_m^H \star \vec{\omega}_j \star \boldsymbol{\beta}_m \left(\vec{\mathscr{P}}_{m+1} - \boldsymbol{\beta}_m \star \mathscr{P}_m \star \mathscr{B}_m^{-1} \star \vec{\mathscr{E}}_m \right). \end{aligned}$$

It follows that the residual can be expressed as

$$\vec{\breve{\mathscr{R}}}_m = \vec{\mathscr{P}}_{m+1} - \boldsymbol{\beta}_m \star \mathscr{P}_m \star \mathscr{B}_m^{-1} \star \vec{\mathscr{E}}_m. \tag{34}$$

We now proceed analogously as in the previous subsection, i.e., we use the smallest harmonic Ritz eigentubes of $\mathscr{B}_{m+1,m}^H \star \mathscr{B}_{m+1,m}$ and associated eigenslices to approximate the k smallest singular triplets of $\mathscr{A}$. This yields relations that are analogous to (11) and (12). The following theorem provides the details.

Theorem 7. Apply m steps of Algorithm 5 to the third-order tensor $\mathscr{A}$ and assume that the tensor $\mathscr{B}_m$ in (11) and (12) is invertible. Then, for $k = 1, \ldots, m-1$, we have the relations

$$\mathscr{A} \star \breve{\mathscr{P}}_{k+1} = \breve{\mathscr{Q}}_{k+1} \star \breve{\mathscr{B}}_{k+1}, \tag{35}$$

$$\mathscr{A}^H \star \breve{\mathscr{Q}}_{k+1} = \breve{\mathscr{P}}_{k+1} \star \breve{\mathscr{B}}_{k+1}^H + \breve{\boldsymbol{\beta}}_{k+1} \star \vec{\breve{\mathscr{P}}}_{k+2} \star \vec{\breve{\mathscr{E}}}_{k+1}^H, \tag{36}$$

where $\breve{\mathscr{P}}_{k+1} \in \mathbb{K}_n^{p\times(k+1)}$ and $\breve{\mathscr{Q}}_{k+1} \in \mathbb{K}_n^{\ell\times(k+1)}$ have orthonormal lateral slices and $\breve{\mathscr{B}}_{k+1} \in \mathbb{K}_n^{(k+1)\times(k+1)}$ is an upper triangular tensor, where the k first lateral slices of $\breve{\mathscr{P}}_{k+1}$ are a t-linear combination of the k first harmonic Ritz lateral slices of $\mathscr{A}$ with $\vec{\breve{\mathscr{P}}}_{k+2} \in \mathbb{K}_n^p$ is orthogonal to $\breve{\mathscr{P}}_{k+1}$. Moreover, $\vec{\breve{\mathscr{E}}}_{k+1} \in \mathbb{K}_n^m$ is the canonical lateral slice under the t-product.

Proof. Let $\{\vec{\breve{\mathscr{V}}}_i\}_{i=1:k}$ be the first k harmonic Ritz lateral slices of $\mathscr{A}$. Using (33) and (34), we get

$$\begin{aligned}\left[\boldsymbol{s}_1' \star \vec{\breve{\mathscr{V}}}_1, \boldsymbol{s}_2' \star \vec{\breve{\mathscr{V}}}_2, \ldots, \boldsymbol{s}_k' \star \vec{\breve{\mathscr{V}}}_k, \vec{\breve{\mathscr{R}}}_m\right] &= \left[\mathscr{P}_m, \vec{\mathscr{P}}_{m+1}\right] \star \begin{bmatrix} \mathscr{B}_m^{-1} \star \mathscr{U}_k' \star \mathscr{S}_k' & -\boldsymbol{\beta}_m \star \mathscr{B}_m^{-1} \star \vec{\mathscr{E}}_m \\ \mathbf{0} & \boldsymbol{e} \end{bmatrix} \\ &= \mathscr{P}_{m+1} \star \begin{bmatrix} \mathscr{B}_m^{-1} \star \mathscr{U}_k' \star \mathscr{S}_k' & -\boldsymbol{\beta}_m \star \mathscr{B}_m^{-1} \star \vec{\mathscr{E}}_m \\ \mathbf{0} & \boldsymbol{e} \end{bmatrix},\end{aligned}$$

where $\boldsymbol{e}$ has been defined in Equation (7).

Introduce the tensor

$$\mathscr{J}_{k+1} = \begin{bmatrix} \mathscr{B}_m^{-1} \star \mathscr{U}_k' \star \mathscr{S}_k' & -\boldsymbol{\beta}_m \star \mathscr{B}_m^{-1} \star \vec{\mathscr{E}}_m \\ \mathbf{0} & \boldsymbol{e} \end{bmatrix}. \tag{37}$$

Using the reduced t-QR factorization of $\mathscr{J}_{k+1}$, we get

$$\mathscr{J}_{k+1} = \mathscr{Q}_{k+1}' \star \mathscr{R}_{k+1}',$$

where $\mathscr{Q}_{k+1}' \in \mathbb{K}_n^{(m+1)\times(k+1)}$ has orthonormal lateral slices and $\mathscr{R}_{k+1}' \in \mathbb{K}_n^{(k+1)\times(k+1)}$ is an f-upper triangular tensor; see [19]. This factorization can be computed by a simple modification of Algorithm 3.

Let

$$\breve{\mathscr{P}}_{k+1} = \left[\vec{\breve{\mathscr{P}}}_1, \vec{\breve{\mathscr{P}}}_2, \ldots, \vec{\breve{\mathscr{P}}}_{k+1}\right] = \mathscr{P}_{m+1} \star \mathscr{Q}_{k+1}' \in \mathbb{K}_n^{\ell\times(k+1)}. \tag{38}$$

Then

$$\begin{aligned}\mathscr{A} \star \breve{\mathscr{P}}_{k+1} &= \mathscr{A} \star \mathscr{P}_{m+1} \star \mathscr{Q}_{k+1}' \\ &= \left[\mathscr{A} \star \mathscr{P}_m, \mathscr{A} \star \vec{\mathscr{P}}_{m+1}\right] \star \mathscr{Q}_{k+1}' \\ &= \left[\mathscr{A} \star \mathscr{P}_m, \mathscr{A} \star \vec{\mathscr{P}}_{m+1}\right] \star \mathscr{J}_{k+1} \star \left(\mathscr{R}_{k+1}'\right)^{-1} \\ &= \left[\mathscr{A} \star \mathscr{P}_m \star \mathscr{B}_m^{-1} \star \mathscr{U}_k' \star \mathscr{S}_k', \mathscr{A} \star \vec{\mathscr{P}}_{m+1} - \mathscr{A} \star \mathscr{P}_m \star \boldsymbol{\beta}_m \star \mathscr{B}_m^{-1} \star \vec{\mathscr{E}}_m\right] \star \left(\mathscr{R}_{k+1}'\right)^{-1} \\ &= \left[\mathscr{Q}_m \star \mathscr{U}_k' \star \mathscr{S}_k', \mathscr{A} \star \vec{\mathscr{P}}_{m+1} - \vec{\mathscr{Q}}_m \star \boldsymbol{\beta}_m\right] \star \left(\mathscr{R}_{k+1}'\right)^{-1}.\end{aligned}$$

Define

$$\breve{\mathscr{Q}}_k = \mathscr{Q}_m \star \mathscr{U}_k' \in \mathbb{K}_n^{p\times k}. \tag{39}$$

Using the orthogonality of $\mathscr{A} \star \vec{\mathscr{P}}_{m+1} - \boldsymbol{\beta}_m \star \vec{\mathscr{Q}}_m$ against the lateral slices of $\breve{\mathscr{Q}}_k$ gives

$$\breve{\boldsymbol{\alpha}}_{k+1} \star \vec{\breve{\mathscr{Q}}}_{k+1} = -\boldsymbol{\beta}_m \star \vec{\mathscr{Q}}_m + \mathscr{A} \star \vec{\mathscr{P}}_{m+1} - \breve{\mathscr{Q}}_k \star \begin{bmatrix} \breve{\boldsymbol{\gamma}}_1 \\ \breve{\boldsymbol{\gamma}}_2 \\ \vdots \\ \breve{\boldsymbol{\gamma}}_k \end{bmatrix}, \tag{40}$$

where $\left\|\vec{\breve{\mathcal{Q}}}_{k+1}\right\| = 1$ and $\breve{\boldsymbol{\alpha}}_{k+1}$ is the tube obtained from the normalization of the tensor

$$-\boldsymbol{\beta}_m \star \vec{\mathcal{Q}}_m + \mathcal{A} \star \vec{\mathcal{P}}_{m+1} - \breve{\mathcal{Q}}_k \star \begin{bmatrix} \breve{\boldsymbol{\gamma}}_1 \\ \breve{\boldsymbol{\gamma}}_2 \\ \vdots \\ \breve{\boldsymbol{\gamma}}_k \end{bmatrix}$$

with

$$\breve{\mathcal{Q}}_k^H \star \left(-\boldsymbol{\beta}_m \star \vec{\mathcal{Q}}_m + \mathcal{A} \star \vec{\mathcal{P}}_{m+1}\right) = \begin{bmatrix} \breve{\boldsymbol{\gamma}}_1 \\ \breve{\boldsymbol{\gamma}}_2 \\ \vdots \\ \breve{\boldsymbol{\gamma}}_k \end{bmatrix}.$$

It follows from (39) and (40) that

$$\begin{aligned} \mathcal{A} \star \breve{\mathcal{P}}_{k+1} &= \left[\mathcal{Q}_m \star \mathcal{U}_k' \star \mathcal{S}_k', \breve{\boldsymbol{\alpha}}_{k+1} \star \vec{\breve{\mathcal{Q}}}_{k+1} + \breve{\mathcal{Q}}_k \star \begin{bmatrix} \breve{\boldsymbol{\gamma}}_1 \\ \vdots \\ \breve{\boldsymbol{\gamma}}_k \end{bmatrix}\right] \star \left(\mathcal{R}_{k+1}'\right)^{-1} \\ &= \left[\mathcal{Q}_m \star \mathcal{U}_k', \vec{\breve{\mathcal{Q}}}_{k+1}\right] \star \begin{bmatrix} \boldsymbol{s}_1' & & & \breve{\boldsymbol{\gamma}}_1 \\ & \ddots & & \vdots \\ & & \boldsymbol{s}_k' & \breve{\boldsymbol{\gamma}}_k \\ & & & \breve{\boldsymbol{\alpha}}_{k+1} \end{bmatrix} \star \left(\mathcal{R}_{k+1}'\right)^{-1}. \end{aligned}$$

Hence,

$$\mathcal{A} \star \breve{\mathcal{P}}_{k+1} = \breve{\mathcal{Q}}_{k+1} \star \breve{\mathcal{B}}_{k+1}, \tag{41}$$

with

$$\breve{\mathcal{B}}_{k+1} = \begin{bmatrix} \boldsymbol{s}_1' & & & \breve{\boldsymbol{\gamma}}_1 \\ & \ddots & & \vdots \\ & & \boldsymbol{s}_k' & \breve{\boldsymbol{\gamma}}_k \\ & & & \breve{\boldsymbol{\alpha}}_{k+1} \end{bmatrix} \star \left(\mathcal{R}_{k+1}'\right)^{-1} \in \mathbb{K}_n^{(k+1)\times(k+1)}, \tag{42}$$

where $\breve{\mathcal{B}}_{k+1}$ is an upper triangular tensor as it is the t-product of two upper triangular tensors.

To show (36), we first notice that

$$\mathcal{A}^H \star \breve{\mathcal{Q}}_k = \mathcal{A}^H \star \mathcal{Q}_m \star \mathcal{U}_k' = \mathcal{P}_{m+1} \star \mathcal{B}_{m,m+1}^H \star \mathcal{U}_k' = \mathcal{P}_{m+1} \star \mathcal{V}_k' \star \mathcal{S}_k'.$$

Using the fact that

$$\mathcal{B}_{m,m+1} = \left[\mathcal{B}_m, \boldsymbol{\beta}_m \star \vec{\mathcal{E}}_m\right] = \mathcal{B}_m \star \left[\mathcal{I}_m, \boldsymbol{\beta}_m \star \mathcal{B}_m^{-1} \star \vec{\mathcal{E}}_m\right],$$

we get

$$\mathcal{B}_{m,m+1} \star \mathcal{V}_k' = \mathcal{U}_k' \star \mathcal{S}_k' \Leftrightarrow \left[\mathcal{I}_m, \boldsymbol{\beta}_m \star \mathcal{B}_m^{-1} \star \vec{\mathcal{E}}_m\right] \star \mathcal{V}_k' = \mathcal{B}_m^{-1} \star \mathcal{U}_k' \star \mathcal{S}_k'.$$

It follows from the above result that

$$\mathcal{V}_k' = \begin{bmatrix} \mathcal{B}_m^{-1} \star \mathcal{U}_k' \star \mathcal{S}_k & -\boldsymbol{\beta}_m \star \mathcal{B}_m^{-1} \star \vec{\mathcal{E}}_m \\ \mathbf{0} & \boldsymbol{e} \end{bmatrix} \star \begin{bmatrix} \mathcal{I}_k \\ \vec{\mathcal{E}}_{m+1}^H \star \mathcal{V}_k' \end{bmatrix} = \mathcal{J}_{k+1} \star \begin{bmatrix} \mathcal{I}_k \\ \vec{\mathcal{E}}_{m+1}^H \star \mathcal{V}_k' \end{bmatrix}.$$

We obtain

$$\begin{aligned}
\mathscr{A}^H \star \breve{\mathscr{Q}}_k &= \mathscr{A}^H \star \mathscr{Q}_m \star \mathscr{U}_k' \\
&= \mathscr{P}_{m+1} \star \mathscr{B}_{m,m+1} \star \mathscr{U}_k' \\
&= \mathscr{P}_{m+1} \star \mathscr{V}_k' \star \mathscr{S}_k' \\
&= \mathscr{P}_{m+1} \star \mathscr{J}_{k+1} \star \begin{bmatrix} \mathscr{I}_k \\ \vec{\mathscr{E}}_{m+1}^H \star \mathscr{V}_k' \end{bmatrix} \star \mathscr{S}_k' \\
&= \mathscr{P}_{m+1} \star \mathscr{Q}_{k+1}' \star \mathscr{R}_{k+1}' \star \begin{bmatrix} \mathscr{I}_k \\ \vec{\mathscr{E}}_{m+1}^H \star \mathscr{V}_k' \end{bmatrix} \star \mathscr{S}_k' \\
&= \breve{\mathscr{P}}_{k+1} \star \mathscr{R}_{k+1}' \star \begin{bmatrix} \mathscr{I}_k \\ \vec{\mathscr{E}}_{m+1}^H \star \mathscr{V}_k' \end{bmatrix} \star \mathscr{S}_k'.
\end{aligned}$$

The relation (41) now yields

$$\breve{\mathscr{Q}}_k^H \star \mathscr{A} \star \breve{\mathscr{P}}_{k+1} = \breve{\mathscr{B}}_{k,k+1} \Leftrightarrow \breve{\mathscr{P}}_{k+1}^H \star \mathscr{A}^H \star \breve{\mathscr{Q}}_k = \breve{\mathscr{B}}_{k,k+1}^H,$$

where $\breve{\mathscr{B}}_{k,k+1} \in \mathbb{K}_n^{(k+1)\times k}$ is the subtensor of $\breve{\mathscr{B}}_{k+1}$, which is obtained by removing the last horizontal slice of $\breve{\mathscr{B}}_{k+1}$. Then

$$\breve{\mathscr{P}}_{k+1}^H \star \mathscr{A}^H \star \breve{\mathscr{Q}}_k = \mathscr{R}_{k+1}' \star \begin{bmatrix} \mathscr{I}_k \\ \vec{\mathscr{E}}_{m+1}^H \star \mathscr{V}_k' \end{bmatrix} \star \mathscr{S}_k' = \breve{\mathscr{B}}_{k,k+1}^H$$

and

$$\breve{\mathscr{P}}_{k+1}^H \star \mathscr{A}^H \star \vec{\breve{\mathscr{Q}}}_{k+1} = \breve{\mathscr{B}}_{k+1}^H \star \breve{\mathscr{Q}}_{k+1}^H \star \vec{\breve{\mathscr{Q}}}_{k+1} = \breve{\mathscr{B}}_{k+1}^H \star \vec{\mathscr{E}}_{k+1} = \breve{\boldsymbol{\alpha}}_{k+1} \star \vec{\mathscr{E}}_{k+1}.$$

Hence,

$$\mathscr{A}^H \star \vec{\breve{\mathscr{Q}}}_{k+1} = \breve{\boldsymbol{\alpha}}_{k+1} \star \vec{\breve{\mathscr{P}}}_{k+1} + \vec{\breve{\mathscr{R}}}_{k+1}' \tag{43}$$

with $\vec{\breve{\mathscr{R}}}_{k+1}' \perp \breve{\mathscr{P}}_{k+1}$. It follows that

$$\mathscr{A}^H \star \breve{\mathscr{Q}}_{k+1} = \breve{\mathscr{P}}_{k+1} \star \breve{\mathscr{B}}_{k+1}^H + \vec{\breve{\mathscr{R}}}_{k+1}' \star \vec{\mathscr{E}}_{k+1}^H.$$

Normalization of $\vec{\breve{\mathscr{R}}}_{k+1}'$ gives

$$\mathscr{A}^H \star \breve{\mathscr{Q}}_{k+1} = \breve{\mathscr{P}}_{k+1} \star \breve{\mathscr{B}}_{k+1}^H + \breve{\boldsymbol{\beta}}_{k+1} \star \vec{\breve{\mathscr{P}}}_{k+2} \star \vec{\mathscr{E}}_{k+1}^H.$$

The orthonormality of the lateral slices of $\breve{\mathscr{P}}_{k+1}$ and $\breve{\mathscr{Q}}_{k+1}$ holds by the construction of these tensors. Specifically, it follows from (38) that the lateral slices of $\breve{\mathscr{P}}_{k+1}$ are orthonormal. Due to (39), the first k lateral slices of $\breve{\mathscr{Q}}_{k+1}$ are orthonormal. □

Notice that if $\breve{\boldsymbol{\beta}}_{k+1}$ given in (36) vanishes, then we have determined k singular triplets, i.e., these singular triplets of $\mathscr{A}$ can be computed by using the singular triplets of $\breve{\mathscr{B}}_{k+1}$, as well as $\breve{\mathscr{P}}_{k+1}$ and $\breve{\mathscr{Q}}_{k+1}$ defined in (35) and (36). If $\breve{\boldsymbol{\beta}}_{k+1}$ does not vanish, then we append new lateral slices to $\breve{\mathscr{P}}_{k+1}$ and $\breve{\mathscr{Q}}_{k+1}$ in a similar way as we did in the previous subsection. The following result is analogous to Theorem 6.

Theorem 8. Carry out m steps of Algorithm 5 and assume that eqs (35) and (36) hold for $k = 1, 2, \ldots, m-1$. Further, let $\breve{\boldsymbol{\beta}}_{k+1}$ in (36) be nonvanishing. Then we have the following relations

$$\begin{aligned}
\mathscr{A} \star \breve{\mathscr{P}}_m &= \breve{\mathscr{Q}}_m \star \breve{\mathscr{B}}_m, \\
\mathscr{A}^H \star \breve{\mathscr{Q}}_m &= \breve{\mathscr{P}}_m \star \breve{\mathscr{B}}_m^H + \breve{\boldsymbol{\beta}}_m \star \vec{\breve{\mathscr{P}}}_{m+1} \star \vec{\mathscr{E}}^H,
\end{aligned}$$

where $\breve{\mathscr{P}}_m \in \mathbb{K}_n^{p\times m}$ and $\breve{\mathscr{Q}}_m \in \mathbb{K}_n^{\ell\times m}$ are orthonormal tensors, $\breve{\mathscr{B}}_m \in \mathbb{K}_n^{m\times m}$ is an upper triangular tensor, $\breve{\boldsymbol{\beta}}_m$ is a tube of n elements, $\vec{\breve{\mathscr{P}}}_{m+1} \in \mathbb{K}_n^p$ is orthogonal to all the lateral slices of $\breve{\mathscr{P}}_m$ and $\vec{\mathscr{E}}^H \in \mathbb{K}_n^\ell$ is the canonical lateral slice under the t-product, where the first k lateral slices of $\breve{\mathscr{P}}_m$ and $\breve{\mathscr{Q}}_m$ are the same as the lateral slices of $\breve{\mathscr{P}}_{k+1}$ and $\breve{\mathscr{Q}}_{k+1}$, respectively, given in Theorem 7.

Proof. These results can be shown similarly as Theorem 6. □

Theorem 7 requires the invertibility of $\mathscr{B}_m$. Notice that this tensor is well conditioned if all the frontal slices of $\widehat{\mathscr{B}}_m$ are well conditioned, i.e., if

$$\max_{1\leq i\leq n} \kappa\left(\widehat{\mathscr{B}}_m^{(i)}\right)$$

is small, where

$$\kappa(\widehat{\mathscr{B}}_m^{(i)}) = \frac{\left(\widehat{\boldsymbol{s}}_1^{\mathscr{B}_m}\right)^{(i)}}{\left(\widehat{\boldsymbol{s}}_m^{\mathscr{B}_m}\right)^{(i)}}.$$

Algorithm 6 describes computations required to compute approximations of either the k largest singular triplets or the k smallest singular triplets of a third-order tensor $\mathscr{A}$ using the methods we developed in the present and previous subsections.

Algorithm 6 Tensor Lanczos Bidiagonalization Ritz (t-LBR) algorithm for computing the largest and the smallest singular triplets.

Input: $\mathscr{A} \in \mathbb{K}_n^{\ell\times p}$.
m: the number of tensor Lanczos bidiagonalization steps.
$\vec{\mathscr{P}}_1 \in \mathbb{K}_n^p$ with unit norm.
k: the number of the desired singular triplets.
δ: The tolerance to accept the singular triplets approximated.
ϵ: machine epsilon.
type: A Boolean variable for the kind of augmentation which is either 'Ritz' for Ritz augmentation or 'Harm' for harmonic Ritz augmentation.
Output: The k desired singular triplets of $\mathscr{A}$, $\{\sigma_i, \vec{\mathscr{U}}_i, \vec{\mathscr{V}}_i\}_{i=1:k}$.

1: Compute the Partial Lanczos bidiagonalization of $\mathscr{A}$ by Algorithm 5.
2: Compute the t-SVD of $\mathscr{B}_m$ using Algorithm 2.
3: Check the convergence stated in Equation (17). If all the k desired singular triplets are well approximated, then exist.
4: Compute the augmented vectors:
5: **if** type='Ritz' or $\boldsymbol{k}(\mathscr{B}_m) > \epsilon^{\frac{1}{2}}$ **then**
6: Compute the tensors $\mathscr{P} := \widetilde{\mathscr{P}}_{k+1}$, $\mathscr{Q} := \widetilde{\mathscr{Q}}_{k+1}$, $\mathscr{B} := \widetilde{\mathscr{B}}_{k+1}$ and the residual $\vec{\mathscr{F}}_k$ from (20), (23), (24) and (26).
7: **end if**
8: **if** type='Harm' and $\boldsymbol{k}(\mathscr{B}_m) \leq \epsilon^{\frac{1}{2}}$ **then**
9: Compute the t-SVD of $\mathscr{B}_{m,m+1}$.
10: Compute the t-QR factorization of $\mathscr{J}_{k+1}$ in (37).
11: Compute the tensors $\mathscr{P} := \breve{\mathscr{P}}_{k+1}$, $\mathscr{Q} := \breve{\mathscr{Q}}_{k+1}$, $\mathscr{B} := \breve{\mathscr{B}}_{k+1}$ and the residual $\vec{\breve{\mathscr{R}}}_m$ from (38), (39), (42) and (43).
12: **end if**
13: Append $m-k$ lateral slices to $\mathscr{P}$ and $\mathscr{Q}$, and $m-k$ horizontal and lateral slices to $\mathscr{B}$ to obtain $\mathscr{P}_m$, $\mathscr{Q}_m$ and $\mathscr{B}_m$, and determine a new residual $\vec{\mathscr{R}}_m$.
14: Go to 2.

4 | MULTIDIMENSIONAL PRINCIPAL COMPONENT ANALYSIS FOR FACIAL RECOGNITION

Principal component analysis (PCA) is used in numerous areas of science and engineering, such as in data denoising, image classification, and facial recognition. Some approaches to color image classification involve conversion of color images to gray scale images to reduce the computational burden, because color images are represented by tensors, while gray scale images can be represented by matrices; see[33,34]. However, this conversion entails loss of information. A color image in RGB format can be represented by a third-order tensor. This section discusses the application of PCA to third-order tensors.

PCA when applied to gray scale face recognition computes a set of characteristics (eigenfaces) corresponding to the main components of the initial set of training images. Recognition is done by projecting the training images into the eigenface subspace, in which an image of a person is classified by comparing it with other available images in the eigenface subspace. The main advantages of this procedure are its simplicity, speed, and insensitivity to small changes of the faces.

When applying PCA to third-order tensors, the t-product, tubes, lateral slices, and third-order tensors are analogues of scalars, vectors, and matrices in the eigenface technique for classifying grayscale images. Using this identification, PCA for third-order tensors that represent color images is structurally very similar to PCA for matrices that represent grayscale images. The latter is described in [14].

Let N training color images $I_1, I_2, \dots, I_N$ of size $\ell \times p \times n$ be available. They are represented by the third-order tensors $\mathscr{I}_1, \mathscr{I}_2, \dots, \mathscr{I}_N$ in $\mathbb{K}_n^{\ell \times p}$. The procedure of recognizing color facial images using third-order tensors is as follows:

1. For each image I_i for $i = 1, 2, \dots, N$, we determine a lateral slice $\vec{\mathscr{X}}_i \in \mathbb{K}_n^{\ell p}$ by vectorizing each frontal slice, i.e., $\vec{\mathscr{X}}_i^{(s)} = \texttt{vec}(\mathscr{I}_i^{(s)})$ for $s = 1, 2, \dots, n$. We then construct a tensor, whose frontal slices are given by $\vec{\mathscr{X}}_i$, i.e.,
$$\mathscr{X} = \left[\vec{\mathscr{X}}_1, \vec{\mathscr{X}}_2, \dots, \vec{\mathscr{X}}_N\right] \in \mathbb{K}_n^{\ell p \times N}.$$
2. Compute the mean of the frontal slices of $\mathscr{X}$, i.e.,
$$\vec{\mathscr{M}} = \sum_{i=1}^{N} \frac{\vec{\mathscr{X}}_i}{N},$$
and let
$$\overline{\mathscr{X}} = [\overrightarrow{\overline{\mathscr{X}}}_1, \overrightarrow{\overline{\mathscr{X}}}_2, \dots, \overrightarrow{\overline{\mathscr{X}}}_N], \qquad \overrightarrow{\overline{\mathscr{X}}}_i = \vec{\mathscr{X}}_i - \vec{\mathscr{M}}.$$
3. Determine the first k left singular vectors of $\overline{\mathscr{X}}$. We denote them by $\vec{\mathscr{U}}_1, \dots, \vec{\mathscr{U}}_k$. Construct the projection subspace
$$\mathbb{U}_k = \texttt{span}\left\{\vec{\mathscr{U}}_1, \vec{\mathscr{U}}_2, \dots, \vec{\mathscr{U}}_k\right\} \tag{44}$$
and let
$$\mathscr{U}_k = \left[\vec{\mathscr{U}}_1, \vec{\mathscr{U}}_2, \dots, \vec{\mathscr{U}}_k\right] \in \mathbb{K}_n^{\ell p \times k}.$$
4. Project each face I_i onto the subspace (44) to obtain $\mathscr{U}_k^H \star \overrightarrow{\overline{\mathscr{X}}}_i$. A test image I_0 also is projected onto the same space to get $\mathscr{U}_k^H \star \left(\vec{\mathscr{X}}_0 - \vec{\mathscr{M}}\right)$. Finally, determine the closest image to the test image by computing the minimal distance between the projected test image and all the projected training images.

The main difference between methods that use PCA for facial recognition is the way that the first (dominant) left singular vectors of $\overline{\mathscr{X}}$ are computed. In the present paper, we use our proposed method to compute the dominant singular triplets that are used in PCA. The following algorithm summarises the different steps in our approach.

Algorithm 7 Facial recognition using tensor Lanczos bidiagonalization with Ritz augmentation.

1: **Input:** Training set of images $\mathscr{X}$ (N images), mean image $\overline{\mathscr{X}}$, test image $\mathscr{I}_0$ with its associate lateral slice $\vec{\mathscr{X}}_0 = \texttt{vec}(\mathscr{I}_0)$; m the number of tensor Lanczos bidiagonalization algorithm; k the number of the desired left singular slices.
2: **Output:** Closest image in the database.
3: $[\mathscr{U}_k, \mathscr{S}_k, \mathscr{V}_k]$ = t-LBR($\overline{\mathscr{X}}, m, k$) using Algorithm 6.
4: Project $\overline{\mathscr{X}}$ onto $\mathbb{U}_k$ to get $\mathscr{P} = \mathscr{U}_k^H \star \overline{\mathscr{X}}$.
5: Project the mean of the test image I_0 onto $\mathbb{U}_k$, $\vec{\mathscr{P}}_0 = \mathscr{U}_k^H \star \left(\vec{\mathscr{X}}_0 - \vec{\mathscr{M}}\right) = \mathscr{U}_k^H \overrightarrow{\overline{\mathscr{X}}}_0$.
6: Find $i = \underset{i=1,2,\dots,N}{\arg\min} \left\|\vec{\mathscr{P}}_0 - \vec{\mathscr{P}}_i\right\|_F$.

i	Methods	$100\times100\times3$	$500\times500\times3$	$1000\times1000\times3$	$100\times100\times5$	$500\times500\times5$
1	Ritz	7.13e-14	1.60e-13	2.27e-13	2.85e-14	1.63e-13
	GK	8.16e-10	0.09	0.01	3.18e-08	0.01
2	Ritz	9.29e-14	1.98e-13	1.56e-13	5.62e-14	1.48e-13
	GK	1.27e-05	0.07	0.44	3.12e-04	0.15
3	Ritz	5.01e-14	2.70e-13	8.93e-14	5.41e-14	2.66e-13
	GK	0.02	0.95	1.78	6.05e-04	0.51
4	Ritz	3.39e-13	4.92e-11	9.01e-13	3.39e-14	6.74e-13
	GK	0.01	1.60	3.37	0.08	2.03

Table 1 The Frobenius norm $\|\mathscr{S}(i,i,:)-\mathbf{\Sigma}(i,i,:)\|_F$, where $\mathscr{S}(i,i,:)$ denotes the singular tubes computed by either augmentation by Ritz lateral slices (Ritz) or by partial Lanczos bidiagonalization also known a partial Golub-Kahan bidiagonalization (GK), and $\mathbf{\Sigma}(i,i,:)$ stands for the singular tubes determined by the t-SVD method with $m=20$ for $i=1,2,3,4$.

Ritz	$100\times100\times3$		$500\times500\times3$		$1000\times1000\times3$		$100\times100\times5$		$500\times500\times5$	
augmentation	iter	time	iter	time	iter	time	iter	time	iter	time
$m=10$	15	0.40	29	2.84	41	18.20	13	0.41	29	4.09
$m=20$	3	0.15	5	2.14	7	12.88	3	0.18	5	2.91

Table 2 Number of iterations (iter) needed by the Ritz augmentation method to determine the four largest singular tubes for third-order tensors of different sizes with $m=10,\ 20$. The columns with header "time" shows the CPU time in seconds.

5 | NUMERICAL EXPERIMENTS

This section illustrates the performance of Algorithm 6 for detecting the largest or smallest singular triplets when applied to synthetic data, tensor compression, and facial recognition. All computations are carried out on a laptop computer with 2.3 GHz Intel Core i5 processors and 8 GB of memory using MATLAB 2018a.

5.1 | Examples with synthetic data

We use synthetic data generated by the MATLAB command `randn`(ℓ,p,n), which generates a tensor $\mathscr{A}\in\mathbb{K}_n^{\ell\times p}$, whose entries are normally distributed pseudorandom numbers with mean zero and variance one.

5.1.1 | Largest singular values

Table 1 displays the error in the four largest approximate singular tubes computed by augmentation by Ritz lateral slices (referred to as Ritz in the table) and by the partial Lanczos bidiagonalization/Golub-Kahan algorithm (referred to as GK in the table) as described in [14], but using the t-product. These errors are given by $\|\mathscr{S}(i,i,:)-\mathbf{\Sigma}(i,i,:)\|_F$ for $i=1,2,3,4$ with $m=20$, where $\mathscr{S}(i,i,:)$, and $\mathbf{\Sigma}(i,i,:)$ denote, respectively, the i^{th} approximated singular tube, and the i^{th} exact singular tube given from the t-SVD. Table 2 shows the number of iterations required when using augmentation by Ritz lateral slices to approximate the four largest singular triplets for tensors of different sizes and the number of Lanczos bidiagonalization steps m.

Table 1 shows the Ritz augmentation method to yield much higher accuracy than the GK method. Figures 1 and 2 display the values of some frames of the first 10 singular tubes of third-order tensors of sizes $100\times100\times3$ and $1000\times1000\times5$, respectively, computed by Ritz augmentation using Algorithm 6, the t-SVD, and partial Lanczos bidiagonalization (GK). Each tube is denoted by $\mathscr{S}(k,k,:)\in\mathbb{K}_n$, where n is equal to 3 or 5, and $k=1,2,\ldots,10$. In other word, for a fixed i with $1\le i\le n$, we plot $\mathscr{S}(k,k,i)\in\mathbb{K}_n$ for $k=1,2,\ldots,10$. As mentioned above, the ith computed singular triplet is accepted as an approximate singular triplet if $\vec{\mathscr{R}}_m\star\vec{\mathscr{E}}_m^H\star\vec{\mathscr{U}}_i$ is small enough for $1\le i\le k$, where k is the number of desired singular triplets and the $\vec{\mathscr{U}}_i$ are left singular lateral slice of the current tensor $\mathscr{B}_m$; see eq. (17). Figure 3 shows the evolution of the error computed by (17) for the first three singular triplets determined by Algorithm 6 when applied to a third-order tensor of size $1000\times1000\times3$ for $m=20$.

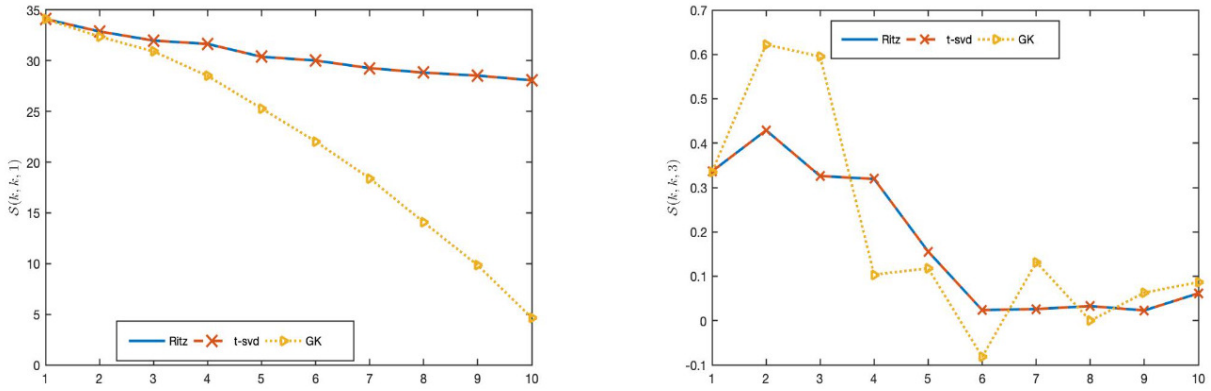

Figure 1 On the left, we display the values of the first frontal slices (frames) of the first 10 singular tubes detected by t-SVD, Ritz augmentation and Partial Lanczos bidiagonalization (GK) for a synthetic data of size $100 \times 100 \times 3$ with $m = 20$, and on the right we plotted the third frontal slices of these tubes, i.e., $\mathscr{S}(k,k,i)$ with $k = 1, 2, \ldots, 10$ and $i = 1, 3$.

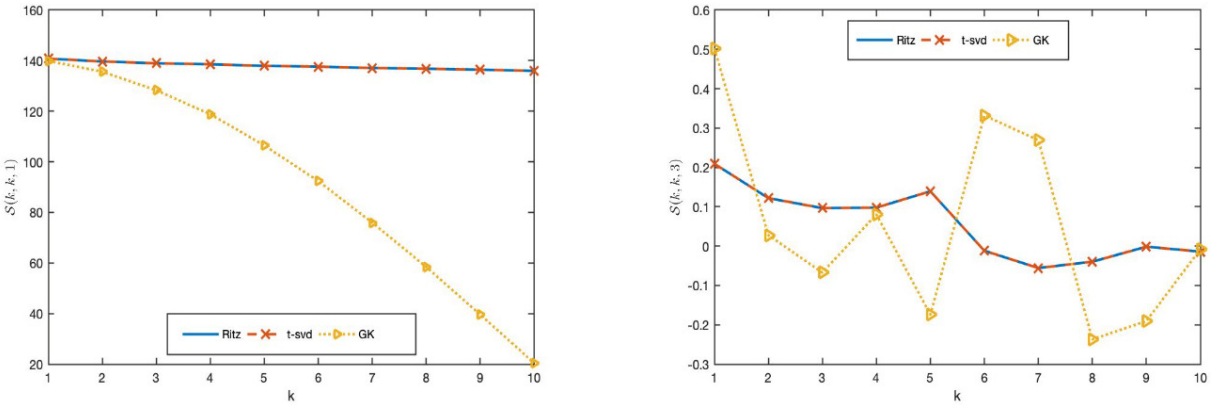

Figure 2 The left-hand side pane shows the values of the first frontal slices (frames) of the first 10 singular tubes computed by t-SVD, Ritz augmentation, and the partial Lanczos bidiagonalization (GK) method for a synthetic data of size $1000 \times 1000 \times 5$ with $m = 20$. The right-hand side pane displays the third frontal slices of these tubes, i.e., $\mathscr{S}(k,k,i)$ for $k = 1, 2, \ldots, 10$ and $i = 1, 3$.

Figures 1 and 2 illustrate that using Algorithm 6 with Ritz augmented method gives more accurate approximations than the GK method. In particular, the frontal slices of each tube computed with Algorithm 6 are very close to the corresponding frontal slices of the tubes determined by the t-SVD, independently of the size of the third-order tensor.

5.1.2 | Smallest singular values

This subsection illustrates the performance of Algorithm 6 with Ritz augmentation (referred to as Ritz) and with harmonic Ritz augmentation (referred to as Harm) for computing the smallest singular triplets of synthetic third-order tensors of different sizes. Table 3 displays the error in the fourth smallest singular tubes computed by Ritz augmentation and harmonic Ritz augmentation for $m = 20$, and compares with results determined by the t-SVD method. In Table 4 we show the number of iterations and the required CPU time (in seconds) for these methods when $m = 20$.

Tables 3 and 4 show that harmonic Ritz augmentation gives higher accuracy than Ritz augmentation when computing the smallest singular triplets. Figures 4 and 5 depict the Frobenius norm of the remainder term $\vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\mathscr{U}}_i$ for each iteration with Algorithm 6 with Ritz augmentation and harmonic Ritz augmentation when approximating the last two singular triplets for $m = 20$.

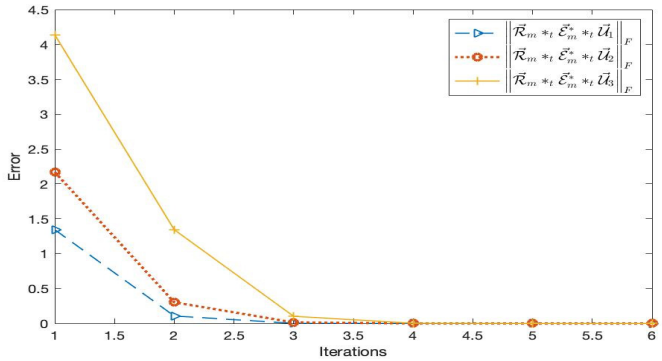

Figure 3 Evolution of the remainder term for a third-order tensor of size $1000 \times 1000 \times 3$ when computing the first three singular triplets by Algorithm 6 with Ritz augmentation.

i	Method	$100 \times 100 \times 3$	$100 \times 100 \times 5$	$500 \times 500 \times 3$	$500 \times 500 \times 5$
$n-3$	Ritz	3.82e-11	5.22e-12	1.34e-10	2.50e-10
	Harm	1.03e-13	4.64e-13	4.66e-13	1.07e-13
$n-2$	Ritz	1.99e-14	4.34e-13	1.20e-14	1.68e-11
	Harm	4.94e-15	3.10e-13	2.46e-14	3.77e-14
$n-1$	Ritz	8.36e-14	4.56e-14	1.77e-14	6.86e-12
	Harm	1.64e-15	6.05e-15	2.88e-14	1.39e-13
n	Ritz	1.38e-15	7.71e-16	6.49e-15	2.00e-12
	Harm	8.59e-16	7.90e-16	3.01e-15	1.41e-14

Table 3 The Frobenius norm $\|\mathscr{S}(i,i,:) - \mathbf{\Sigma}(i,i,:)\|_F$, where $\mathscr{S}(i,i,:)$ denotes the singular tubes determined by Ritz augmentation or harmonic Ritz augmentation for $m = 20$, and $\mathbf{\Sigma}(i,i,:)$ are tubes computed by the t-SVD method for the four smallest tubes, i.e., for $i = n-3, n-2, n-1, n$.

Method	$100 \times 100 \times 3$		$500 \times 500 \times 3$		$100 \times 100 \times 5$		$500 \times 500 \times 5$	
	CPU time	iter	CPU time	iter	CPU time	iter	CPU time	iter
Ritz	0.99	31	231.81	615	1.11	30	425.83	831
Harm	0.85	29	227.49	606	1.03	30	355.35	723

Table 4 CPU time in seconds, and number of iterations required by Algorithm 6 with Ritz augmentation and harmonic Ritz augmentation for $m = 20$ to compute the four smallest singular triplets of synthetic third-order tensors of different sizes.

Figures 4 and 5 show the error $\|\vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\mathscr{U}}_i\|_F$ associated with Ritz augmentation in Algorithm 6 to converge in a smoother way than the corresponding error for harmonic Ritz augmentation. Both errors converge to zero as the number of iterations increases.

5.2 | Application to data compression

Figure 6 displays examples of image compression using two color images: "house" of size $256 \times 256 \times 3$ and "Hawaii" of size $1200 \times 1200 \times 3$. For each image, we compute the k largest singular triplets using Ritz augmentation in Algorithm 6, which will be referred to as "Ritz," for different numbers k of desired singular triplets. Figure 7 displays the relative error of the compressed

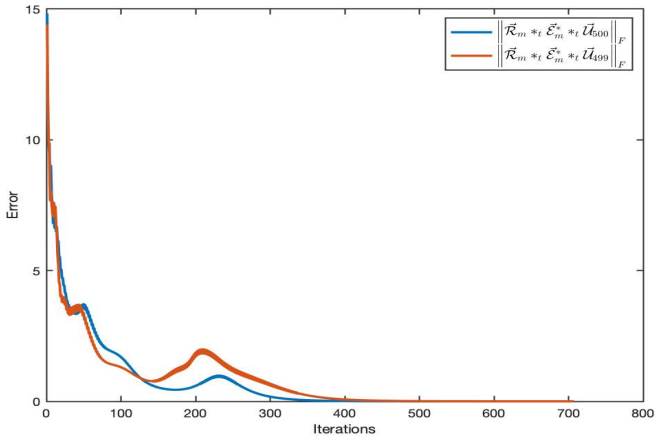

Figure 4 The Frobenius norm of $\vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\mathscr{U}}_i$ obtained by Algorithm 6 with Ritz augmentation when approximating the two smallest singular triplets of a synthetic tensor of size $500 \times 500 \times 5$ with $m = 20$ at each iteration for $i = 499, 500$.

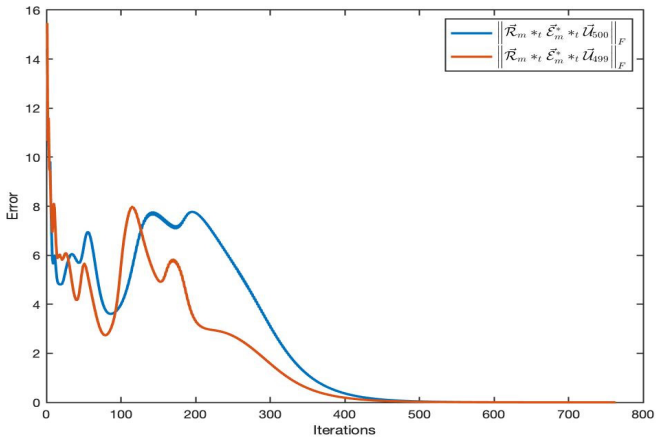

Figure 5 The Frobenius norm of $\vec{\mathscr{R}}_m \star \vec{\mathscr{E}}_m^H \star \vec{\mathscr{U}}_i$ obtained by harmonic Ritz augmentation when approximating the last two singular triplets of a synthetic tensor data of size $500 \times 500 \times 5$ with $m = 20$, at each iteration for $i = 499, 500$.

images for $k = 5, 10, 15, 25$, by using Ritz augmentation (Ritz) and the t-SVD method. This error is measured by

$$\frac{\|\mathscr{A}_k - \mathscr{A}\|_F}{\|\mathscr{A}\|_F}, \tag{45}$$

where $\mathscr{A}$ denotes the tensor that represents the original image and $\mathscr{A}_k = \sum_{i=1}^{k} \vec{\mathscr{U}}_i \star \boldsymbol{s}_i \star \vec{\mathscr{V}}_i^H$.

Figure 7 shows the relative errors obtained with Algorithm 6 with Ritz augmentation and the t-SVD are almost the same. This means that the approximate singular tubes and the right and left singular lateral slices determined by Algorithm 6 with Ritz augmentation are very accurate.

5.3 | Facial recognition

We illustrate the application of Algorithm 7 to facial recognition using color images that are represented by third-order tensors. The images in our test are from the Georgia Tech database GTDB_crop[35], which contains 750 images of 50 persons, with

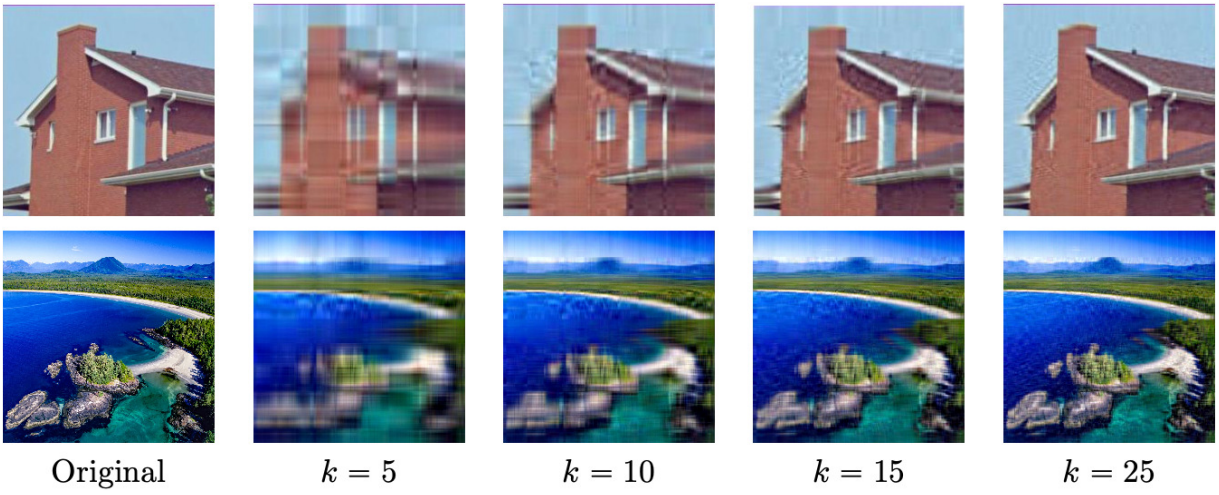

Figure 6 Examples of image compression applied to the "house" and "Hawaii" images for $k = 5, 10, 15, 25$ slices using Algorithm 6 with Ritz augmentation.

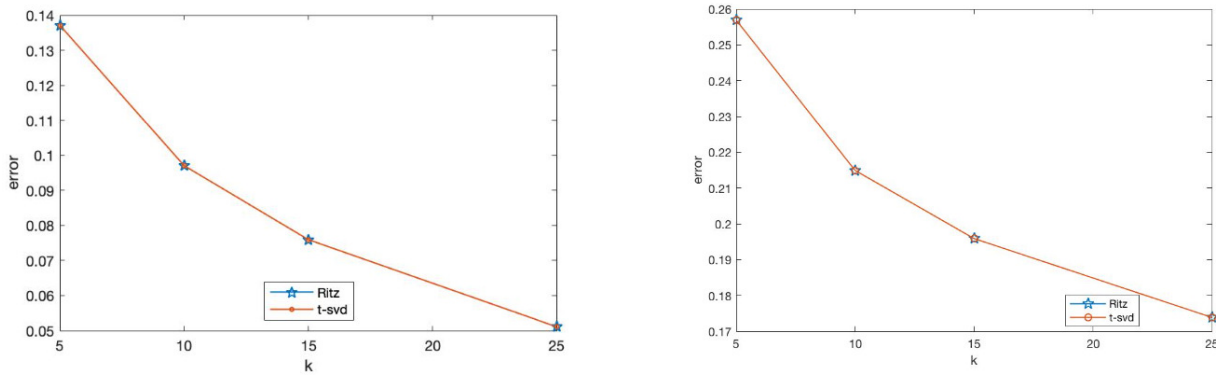

Figure 7 Relative compression error (45) for the images "house" and "Hawaii" obtained with Algorithm 6 with Ritz augmentation (Ritz) and the t-SVD method.

each person represented by 15 images that show various facial expressions and facial orientations, and different illumination conditions. Figure 8 shows an example of images of one person in the data set.

Each image in the data set is of size $100 \times 100 \times 3$ pixels, and we use 3 randomly chosen images of each person as test images. The remaining 600 images form our training set and define the tensor $\mathscr{X} \in \mathbb{K}_3^{10000\times 600}$. We applied Algorithm 7 and compared the results with those obtained by the t-SVD and also with results obtained by the Golub-Kahan (GK) algorithm using the t-product. The performance of these methods is measured by the identification rate given by

$$\text{Identification rate} = \frac{\text{number of correctly matched images}}{\text{number of test images}} \times 100(\%). \tag{46}$$

Figures 9 and 10 show results obtained for $k = 1$ and $k = 5$ for two different persons. The mean image is defined as in Algorithm 7.

Figures 9 and 10 show that Algorithm 7 performs well for some values of the truncation index k. In Figure 11, we plotted the identification rate (46) obtained with Algorithm 7 (Ritz augmentation), GK for $m = k$, and with the exact t-SVD method for the 150 test images.

Figure 8 An example of a person with different facial expressions and orientations.

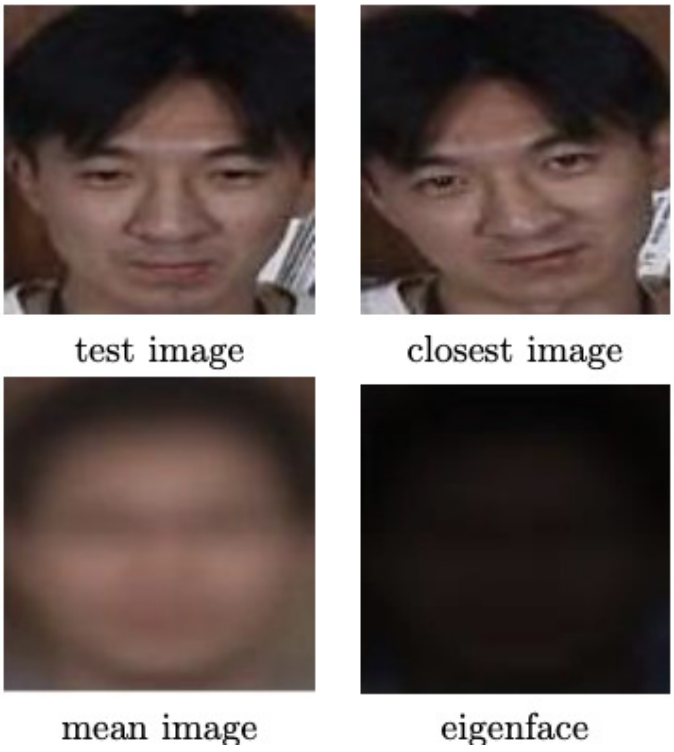

Figure 9 A test for $k = 1$.

k	2		3		4	
Method	Ritz	t-SVD	Ritz	t-SVD	Ritz	t-SVD
CPU time (s)	**10.60**	52.82	**13.11**	63.63	**13.88**	64.77

Table 5 CPU time (in seconds) for Algorithm 7 (Ritz) and for the t-SVD method for $m = 10$ and different values of the truncation index k.

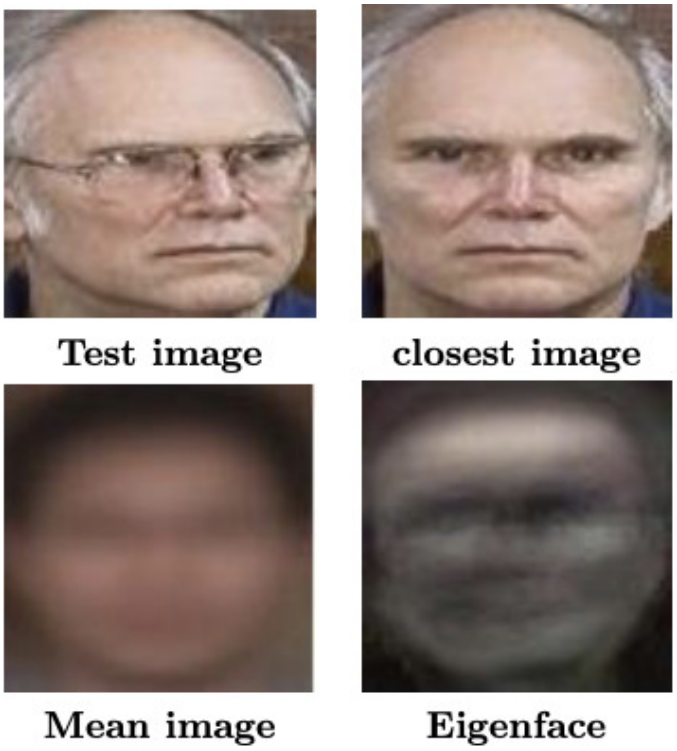

Figure 10 A test for $k = 5$.

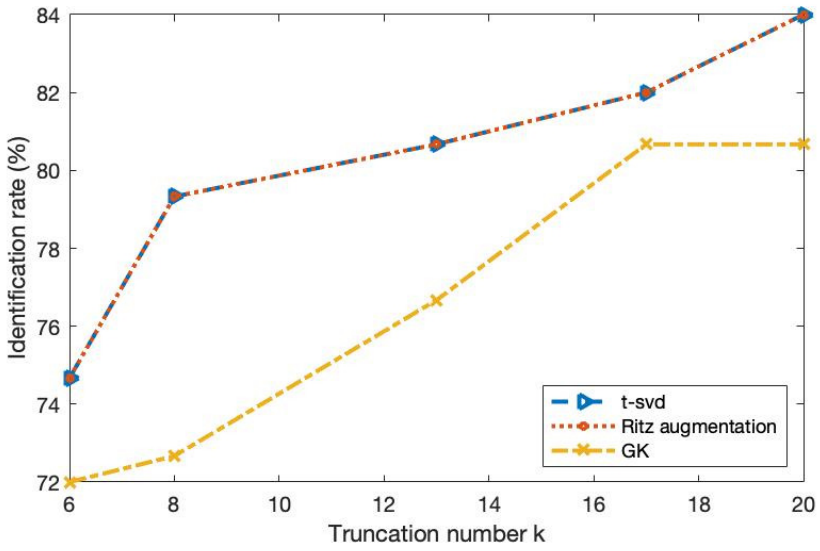

Figure 11 Identification rates for different truncation indices k by Ritz augmentation, t-SVD and Golub-kahan methods.

Table 5 reports CPU times for Algorithm 7 for $m = 10$ (Ritz) and for the t-SVD method for different values of the truncation index k. The results show Algorithm 7 to be very effective both in terms of accuracy and CPU time compared to the t-SVD and the classical Golub-Kahan methods.

6 | CONCLUSION

This paper presents two new methods for approximating the largest or smallest singular triplets of a large third-order tensor using the t-product. We use restarted Lanczos bidiagonalization for third-order tensors to develop the Ritz augmentation method to determine the largest or smallest singular triplets. Moreover, we propose the harmonic Ritz augmentation method to compute the smallest singular triplets. These methods are applied to data compression and face recognition.

ACKNOWLEDGEMENT

The authors would like to thank the referees for helpful comments and valuable suggestions. The authors have no conflict of interest.

References

1. Arnold T, Kane M, and Lewis BW. A Computational Approach To Statistical Learning. CRC Press; 2019.
2. Baglama J, and Reichel L. Augmented implicitly restarted Lanczos bidiagonalization methods. SIAM Journal on Scientific Computing. 2005;**27**(1):19–42.
3. Baglama J, Reichel L, and Lewis BW. irbla: Fast truncated singular value component analysis,https://cran.r-project.org/web/packages/irlba/index.html;.
4. Beik FPA, Najafi-Kalyani M, and Reichel L. Iterative Tikhonov regularization of tensor equations based on the Arnoldi process and some of its generalizations. Applied Numerical Mathematics. 2020;**151**:425–447.
5. Kolda TG, and Bader BW. Tensor decompositions and applications. SIAM review. 2009;**51**(3):455–500.
6. Kilmer ME, Braman K, Hao N, and Hoover RC. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. SIAM Journal on Matrix Analysis and Applications. 2013;**34**(1):148–172.
7. Reichel L, and Ugwu UO. Tensor Arnoldi–Tikhonov and GMRES-Type Methods for Ill-Posed Problems with a t-Product Structure. Journal of Scientific Computing. 2022;**290**:1–39.
8. Kernfeld E, Kilmer M, and Aeron S. Tensor–tensor products with invertible linear transforms. Linear Algebra and its Applications. 2015;**485**:545–570.
9. Reichel L, and Ugwu UO. Tensor Krylov subspace methods with an invertible linear transform product applied to image processing. Applied Numerical Mathematics. 2021;**166**:186–207.
10. Bentbib AH, Hachimi AE, Jbilou K, and Ratnani A. A tensor regularized nuclear norm method for image and video completion. Journal of Optimization Theory and Applications. 2022;**192**(2):401–425.
11. Lu C, Feng J, Chen Y, Liu W, Lin Z, and Yan S. Tensor robust principal component analysis with a new tensor nuclear norm. IEEE transactions on pattern analysis and machine intelligence. 2019;**42**(4):925–938.
12. Kilmer ME, Horesh L, Avron H, and Newman E. Tensor-tensor algebra for optimal representation and compression of multiway data. Proceedings of the National Academy of Sciences. 2021;**118**(28):e2015851118.
13. Bentbib AH, El Hachimi A, Jbilou K, and Ratnani A. Fast multidimensional completion and principal component analysis methods via the cosine product. Calcolo. 2022;**59**(3):26.
14. Hached M, Jbilou K, Koukouvinos C, and Mitrouli M. A multidimensional principal component analysis via the C-Product Golub–Kahan–SVD for classification and face recognition. Mathematics. 2021;**9**(11):1249.

15. Hao N, Kilmer ME, Braman K, and Hoover RC. Facial recognition using tensor-tensor decompositions. SIAM Journal on Imaging Sciences. 2013, volume=;.

16. El Guide M, El Ichi A, Jbilou K, and Sadaka R. On tensor GMRES and Golub-Kahan methods via the T-product for color image processing. Electron J Linear Algebra. 2021;**37**:524–543.

17. El Ichi A, Jbilou K, and Sadaka R. On tensor tubal-Krylov subspace methods. Linear and Multilinear Algebra. 2021;p. 1–24.

18. Reichel L, and Ugwu UO. The tensor Golub–Kahan–Tikhonov method applied to the solution of ill-posed problems with at-product structure. Numerical Linear Algebra with Applications. 2022;**29**(1):e2412.

19. Kilmer ME, and Martin CD. Factorization strategies for third-order tensors. Linear Algebra and its Applications. 2011;**435**(3):641–658.

20. Rojo O, and Rojo H. Some results on symmetric circulant matrices and on symmetric centrosymmetric matrices. Linear algebra and its applications. 2004;**392**:211–233.

21. Golub G, and Kahan W. Calculating the singular values and pseudo-inverse of a matrix. Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis. 1965;**2**(2):205–224.

22. Paige CC, and Saunders MA. LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Transactions on Mathematical Software (TOMS). 1982;**8**(1):43–71.

23. Hochstenbach ME. A Jacobi–Davidson type SVD method. SIAM Journal on Scientific Computing. 2001;**23**(2):606–628.

24. Kokiopoulou E, Bekas C, and Gallopoulos E. Computing smallest singular triplets with implicitly restarted Lanczos bidiagonalization. Applied Numerical Mathematics. 2004;**49**(1):39–61.

25. Björck Å. A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. BIT Numerical Mathematics. 1988;**28**:659–670.

26. Calvetti D, and Reichel L. Tikhonov regularization of large linear problems. BIT Numerical Mathematics. 2003;**43**:263–283.

27. Barlow JL. Reorthogonalization for the Golub–Kahan–Lanczos bidiagonal reduction. Numerische Mathematik. 2013;**124**(2):237–278.

28. Simon HD, and Zha H. Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. SIAM Journal on Scientific Computing. 2000;**21**(6):2257–2274.

29. Jia Z, and Niu D. A refined harmonic Lanczos bidiagonalization method and an implicitly restarted algorithm for computing the smallest singular triplets of large matrices. SIAM Journal on Scientific Computing. 2010;**32**(2):714–744.

30. Sorensen DC. Implicit application of polynomial filters in ak-step Arnoldi method. Siam journal on matrix analysis and applications. 1992;**13**(1):357–385.

31. Stathopoulos A, and Saad Y. Restarting techniques for the (Jacobi-) Davidson symmetric eigenvalue methods. Electron Trans Numer Anal. 1998;**7**:163–181.

32. Paige CC, Parlett BN, and Van der Vorst HA. Approximate solutions and eigenvalue bounds from Krylov subspaces. Numerical Linear Algebra with Applications. 1995;**2**(2):115–133.

33. Bajwa IS, Naweed M, Asif MN, and Hyder SI. Feature based image classification by using principal component analysis. ICGST Int J Graph Vis Image Process GVIP. 2009;**9**:11–17.

34. Pandey PK, Singh Y, and Tripathi S. Image processing using principle component analysis. International Journal of Computer Applications. 2011;**15**(4):37–40.

35. Nefian AV. Georgia Tech Face Database, Available online: http://www.anefian.com/research/face_reco.htm;.