Noname manuscript No.
(will be inserted by the editor)

The global Golub-Kahan method and Gauss quadrature for tensor function approximation

A. H. Bentbib · M. El Ghomari · K. Jbilou · L. Reichel

Dedicated to our friend Claude Brezinski on the occasion of his 80th birthday.

Received: date / Accepted: date

Abstract This paper is concerned with Krylov subspace methods based on the tensor t-product for computing certain quantities associated with generalized third-order tensor functions. We use the tensor t-product and define the tensor global Golub-Kahan bidiagonalization process for approximating tensor functions. Pairs of Gauss and Gauss-Radau quadrature rules are applied to determine the desired quantities with error bounds. An applications to the computation of the tensor nuclear norm is presented and illustrates the effectiveness of the proposed methods.

Keywords Generalized tensor function · tensor t-product · tensor nuclear norm · Golub-Kahan bidiagonalization · Gauss quadrature

1 Introduction

Matrix functions are required in many scientific fields. They appear, e.g., in exponential integrators in differential equations [19] and as centrality and communicability measures [9] in network analysis. For square matrices, matrix functions can be defined in terms of the spectral decomposition or the Jordan canonical form

A. H. Bentbib
Faculté des Sciences et Techniques-Gueliz, Laboratoire de Mathématiques Appliquées et Informatique, Morocco. E-mail: a.bentbib@uca.ac.ma

M. El Ghomari
Department of Mathematics, École Normale Supérieure, Mohammed V University in Rabat, Av. Mohammed Belhassan El Ouazzani, Takaddoum, B.P. 5118, Rabat, Morocco. E-mail: m.elghomari10@gmail.com

K. Jbilou
Université du Littoral, Côte d'Opale, batiment H. Poincarré,
50 rue F. Buisson, F-62280 Calais Cedex, France. University UM6P, Benguerir Morocco.
E-mail: jbilou@univ-littoral.fr

L. Reichel
Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. E-mail: reichel@math.kent.edu

[18]. Generalized matrix functions of rectangular matrices were first proposed by Hawkins and Ben-Israel [17], who based these matrix functions on the singular value decomposition (SVD) or the compact singular value decomposition (CSVD) of the matrix. These generalized matrix functions find applications in Hamiltonian dynamical systems [6] and the analysis of directed networks [1].

Tensors are multidimensional arrays of data and generalize matrices. They have essential roles in, e.g., network analysis [4] and multidifferential equations [26]. Kilmer, Martin, and their collaborators [27,31] introduced the tensor t-product. It generalizes matrix-matrix and matrix-vector products to tensors and has many nice properties. The t-product has found applications in image processing [7,8,25, 31,34,35], signal processing [28,36], and data completion and denoising [2,10,16, 22,30,38]. Lund [29] defined tensor functions based on the t-product of third-order f-square tensors. The frontal slices of these tensors are square matrices and can be expressed in terms of their spectral factorization [29] or their Jordan canonical form [33]. Generalized functions of tensors, whose frontal slices are rectangular matrices, have been described by Miao et al. [32] in terms of the tensor singular value decomposition (t-SVD) and the tensor compact singular value decomposition (t-CSVD). These decompositions are based on the tensor t-product; the t-SVD has been proposed by Kilmer et al. [27].

Global Krylov subspace techniques for reducing large matrices to smaller matrices were first described in [23,24] and were there applied to the solution of linear systems of equations with multiple right-hand sides. These subspace methods are block Krylov methods that use a non-standard inner product, referred to as a Frobenius inner product. The tensor t-global Arnoldi method for color image processing has been described in [7,34]. This paper introduces the tensor t-global Lanczos method, as well as the tensor t-global Golub-Kahan method for approximating generalized tensor functions. These methods are based on the tensor t-product. We describe how pairs of Gauss-type quadrature rules can be used in conjunction with the tensor t-global Lanczos and tensor t-global Golub-Kahan methods to compute upper and lower bounds (or estimates of such bounds) for expressions of the form

$$\mathrm{trace}_{(1)}(g(\mathcal{A}^T * \mathcal{A})), \tag{1}$$

where $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$ is a third-order tensor, the superscript ^{T} stands for transposition (see Definition 2 below), $g(t) := (\sqrt{t})^{-1} f(\sqrt{t})$, where the function f is such that g is well defined for $t \geq 0$, and $\mathrm{trace}_{(1)}(\mathcal{K}) := \sum_{i=1}^{s} K_{ii}^{(1)}$. Here $K_{ii}^{(1)}$ denotes the diagonal entries of the first frontal slice of the third-order tensor $\mathcal{K} \in \mathbb{R}^{s\times s\times p}$. The need to evaluate matrix functions of the form (1) arises, for instance, when computing tensor nuclear norms; see below.

For small tensors $\mathcal{A}$, we can evaluate (1) in a straightforward manner. When the tensor $\mathcal{A}$ is too large to make straightforward computation of (1) feasible or attractive, we determine upper and lower bound, or estimates of such bounds, by first applying a few steps of the t-global Golub-Kahan method to $\mathcal{A}$. This yields a small bidiagonal matrix, and multiplication by its transpose gives a small symmetric tridiagonal matrix, which can be associated with Gauss quadrature rules. These rules can be applied to determine upper and lower bounds for (1), or estimates of such bounds. The main computational effort when applying these rules

is the evaluation of a few tensor-matrix products that are required by the t-global Golub-Kahan algorithm. When the tensor $\mathcal{A}$ is symmetric, the t-global Golub-Kahan method can be replaced by the t-global Lanczos method. This reduces the computational work. The latter method also will be described.

This paper is organized a follows. Section 2 reviews definitions of the tensor t-product and some algebraic properties of third-order tensors. We also recall the definition of t-functions given by Lund [29] and the definition of generalized tensor functions introduced by Miao et al. [32] using the t-CSVD, and describe some of their properties. Section 3 introduces the tensor t-global Lanczos and the tensor t-Golub-Kahan methods. This section also discusses the computation of Gauss-type quadrature rules that can be used to bound or determine estimates of bounds of the expression (1). Numerical examples with applications to tensor nuclear norm computations are presented in Section 4, and Section 5 contains concluding remarks.

2 The tensor t-product

This section reviews the tensor t-product for third order tensors and some of its properties. This product was proposed by Kilmer and Martin [27]. It is generally evaluated with the aid of the Discrete Fourier Transformation (DFT). The beginning of this section reviews results in [27].

Let $v \in \mathbb{R}^p$. The DFT of v, denoted by $\bar{v}$, is given by

$$\bar{v} = F_p v \in \mathbb{C}^p, \tag{2}$$

where $F_p = [f_{ij}]_{i,j=0}^{p-1}$ is the DFT matrix with entries

$$f_{ij} = [\omega_p^{ij}], \quad i, j = 0, 1, \ldots, p-1,$$

and $\omega_p = \exp(-2\pi \mathbf{i}/p)$ is the primitive pth root of unity with $\mathbf{i} = \sqrt{-1}$. The Fast Fourier Transform (FFT) allows the evaluation of the matrix-vector product (2) in only $O(p \log_2(p))$ arithmetic floating point operations (flops). Since $F_p^{-1} = \frac{1}{p} F^H$, where the superscript ^{H} denotes transposition and complex conjugation, the matrix-vector products $F_p^{-1} v$ also can be evaluated in $O(p \log_2(p))$ flops. The FFT and its inverse are implemented in Matlab by the functions `fft` and `ifft`, respectively, i.e., $\bar{v} = \texttt{fft}(v)$ and $v = \texttt{ifft}(\bar{v})$.

Let $v \in \mathbb{R}^p$. It is well known that the entries of the vector

$$\bar{v} = [\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_p] = \texttt{fft}(v)$$

satisfy

$$\bar{v}_1 \in \mathbb{R}, \quad \text{and} \quad \texttt{conj}(\bar{v}_i) = \bar{v}_{p-i+2}, \quad i = 2, \ldots, \left\lfloor \frac{p+1}{2} \right\rfloor,$$

where the operator `conj` complex conjugates its argument.

Let the third order tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$ have frontal slices $\mathcal{A}^{(k)} \in \mathbb{R}^{m\times n}$, $k = 1, 2, \ldots, p$. The operations `bcirc`, `unfold`, and `fold` are defined as

$$\texttt{bcirc}(\mathcal{A}) := \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(p)} & \mathcal{A}^{(p-1)} & \ldots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \mathcal{A}^{(p)} & \ldots & \mathcal{A}^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathcal{A}^{(p)} & \mathcal{A}^{(p-1)} & \ldots & \mathcal{A}^{(2)} & \mathcal{A}^{(1)} \end{bmatrix}, \qquad \texttt{unfold}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} \\ \mathcal{A}^{(2)} \\ \vdots \\ \vdots \\ \mathcal{A}^{(p)} \end{bmatrix},$$

and $\texttt{fold}(\texttt{unfold}(\mathcal{A})) := \mathcal{A}$. Block circular matrices can be block diagonalized by using the DFT [25]. In detail, for any third-order tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$, there is a block diagonal matrix such that

$$\texttt{bcirc}(\mathcal{A}) = (F_p^H \otimes I_m) \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_p \end{bmatrix} (F_p \otimes I_n),$$

where $A_1, \ldots, A_p \in \mathbb{C}^{m\times n}$. The diagonal blocks A_i satisfy

$$\begin{cases} A_i \in \mathbb{C}^{m\times n}, \\ \texttt{conj}(A_i) = A_{p-i+2}, \end{cases}$$

where $\texttt{conj}(A_i)$ denotes the complex conjugate of the matrix A_i. Introduce the tensor $\bar{\mathcal{A}} \in \mathbb{C}^{m\times n\times p}$, whose frontal slices are the diagonal blocks $A_1, \ldots, A_p$, i.e.,

$$\bar{\mathcal{A}} = \texttt{fold} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_p \end{bmatrix}.$$

The tensor $\bar{\mathcal{A}}$ can be computed by applying the FFT to tubes of the tensor $\mathcal{A}$. This can be done with the Matlab command

$$\bar{\mathcal{A}} = \texttt{fft}(\mathcal{A}, [\,], 3).$$

Moreover, $\texttt{ifft}(\bar{\mathcal{A}}, [\,], 3) = \mathcal{A}$.

Definition 1 ([27]) Let $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$ and $\mathcal{B} \in \mathbb{R}^{n\times s\times p}$ be third-order tensors. Then the t-product $\mathcal{A} * \mathcal{B} \in \mathbb{R}^{m\times s\times p}$ is given by

$$\mathcal{A} * \mathcal{B} := \texttt{fold}(\texttt{bcirc}(A)\,\texttt{unfold}(B)).$$

We note that

$$\texttt{unfold}(\mathcal{A} * \mathcal{B}) = \texttt{bcirc}(A)\,\texttt{unfold}(B).$$

Definition 2 ([25,27,29])

– The transpose of the tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$ is the tensor $\mathcal{A}^T \in \mathbb{R}^{n\times m\times p}$ obtained by transposing each of the frontal slices of $\mathcal{A}$ and then reversing the order of the transposed frontal slices 2 through p.

Algorithm 1 The t-product [21]

Input: $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$ and $\mathcal{B} \in \mathbb{R}^{n\times \ell\times p}$.

1. Compute $\bar{\mathcal{A}} = \texttt{fft}(\mathcal{A}, [\,], 3)$ and $\bar{\mathcal{B}} = \texttt{fft}(\mathcal{B}, [\,], 3)$.
2. Compute each frontal slice of $\bar{\mathcal{C}}$ by

$$\bar{C}^{(i)} = \begin{cases} \bar{A}^{(i)}\bar{B}^{(i)}, & i = 1, \cdots, \left\lceil \frac{p+1}{2} \right\rceil \\ \texttt{conj}\left(\bar{C}^{(p-i+2)}\right), & i = \left\lceil \frac{p+1}{2} \right\rceil + 1, \cdots, p \end{cases}$$

3. Compute $\mathcal{C} = \texttt{ifft}(\bar{\mathcal{C}}, [\,], 3)$.

Output: $\mathcal{C} = \mathcal{A} * \mathcal{B} \in \mathbb{R}^{m\times \ell\times p}$.

- A tensor is said to be "f-diagonal" if each frontal slice is diagonal.
- A tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times p}$ is said to be t-symmetric if $\mathcal{A}^T = \mathcal{A}$.
- The identity tensor $\mathcal{I}_{nnp} \in \mathbb{R}^{n\times n\times p}$ is the tensor, whose first frontal slice is the $n \times n$ identity matrix, and the other frontal slices are $n \times n$ zero matrices.
- A tensor $\mathcal{P} \in \mathbb{R}^{n\times n\times p}$ is said to be orthogonal if $\mathcal{P}^T * \mathcal{P} = \mathcal{P} * \mathcal{P}^T = \mathcal{I}_{nnp}$.
- A tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times p}$ is said to be invertible, if there is a tensor $\mathcal{B} \in \mathbb{R}^{n\times n\times p}$ such that $\mathcal{A} * \mathcal{B} = I_{nnp}$ and $\mathcal{B} * \mathcal{A} = I_{nnp}$. In this case, we denote the inverse $\mathcal{B}$ by $\mathcal{A}^{-1}$. It is clear that $\mathcal{A}$ is invertible if and only if $\texttt{bcirc}(\mathcal{A})$ is invertible.
- The inner product of the tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n\times s\times p}$ is defined by

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1=1}^{n}\sum_{i_2=1}^{s}\sum_{i_3=1}^{p} \mathcal{A}_{i_1 i_2 i_3}\mathcal{B}_{i_1 i_2 i_3} = \text{trace}_{(1)}(\mathcal{A}^T * \mathcal{B}),$$

  where for $\mathcal{K} \in \mathbb{R}^{s\times s\times p}$,

$$\text{trace}_{(1)}(\mathcal{K}) := \sum_{i=1}^{s} K^{(1)}_{ii},$$

  and $K^{(1)}$ denotes the first frontal slice of $\mathcal{K}$.
- The Frobenius norm of $\mathcal{A} \in \mathbb{R}^{n\times s\times p}$ is given by

$$\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}.$$

- The trace of $\mathcal{A} \in \mathbb{R}^{n\times s\times p}$ is defined by

$$\text{trace}(\mathcal{A}) := \text{trace}(\texttt{bcirc}(\mathcal{A})).$$

- The trace of $\mathcal{A} \in \mathbb{R}^{n\times s\times p}$ can be computed in terms of $\text{trace}_{(1)}(\mathcal{A})$ as

$$\text{trace}(\mathcal{A}) := p\,\text{trace}_{(1)}(\mathcal{A}). \tag{3}$$

- The tensors $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m\}$ with $\mathcal{V}_i \in \mathbb{R}^{n\times s\times p}$ are said to be F-orthonormal if

$$\langle \mathcal{V}_j, \mathcal{V}_k \rangle = \begin{cases} 1 & j = k, \\ 0 & j \neq k. \end{cases}$$

Definition 3 ([7]) Let the tensors

$$\mathcal{M} = [\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_s] \in \mathbb{R}^{n\times sk\times p} \quad \text{and} \quad \mathcal{N} = [\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_\ell] \in \mathbb{R}^{n\times \ell k\times p}$$

be partitioned into lateral slices $\mathcal{M}_i$ and $\mathcal{N}_i$ of size $n \times k \times p$, respectively. Then the $\diamond$-product of the tensors $\mathcal{M}$ and $\mathcal{N}$ is given by

$$\mathcal{M}^T \diamond \mathcal{N} = [\langle \mathcal{N}_j, \mathcal{M}_i \rangle]_{i=1,2,\ldots,s}^{j=1,2,\ldots,\ell} \in \mathbb{R}^{s\times \ell}.$$

Let $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n\times n\times p}$, and let the function f be such that the matrix $f(\texttt{bcirc}(\mathcal{A}))$ is well defined. Then Lund [29] introduced the tensor t-function

$$f(\mathcal{A}) * \mathcal{B} := \text{fold}\left[f(\texttt{bcirc}(\mathcal{A}))\,\texttt{unfold}(\mathcal{B})\right].$$

2.1 Generalized tensor functions by the tSVD

Generalized matrix functions of a matrix $A \in \mathbb{R}^{m\times n}$ with $m \neq n$ are described, e.g., by Ben-Israel and Greville [3]. They are based on the compact singular value decomposition (CSVD),

$$A = U_r \Sigma_r V_r^T, \tag{4}$$

where $r \leq \min\{m, n\}$ is the rank of A. Here the matrices $U_r \in \mathbb{R}^{m\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ have orthonormal columns, and the nontrivial entries of the diagonal matrix $\Sigma_r = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_r] \in \mathbb{R}^{r\times r}$ are the nonvanishing singular values of A. Typically, the singular values are ordered in nonincreasing order, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$; see, e.g., [14] for details on the CSVD.

Let the function $f : \mathbb{R} \to \mathbb{R}$ be defined on the positive real axis. The generalized matrix function determined by f and (4) is given by

$$f^{gen}(A) = U_r f(\Sigma_r) V_r^T, \quad f(\Sigma_r) = \text{diag}[f(\sigma_1), f(\sigma_2), \ldots, f(\sigma_r)]. \tag{5}$$

Lemma 1 *Let $A \in \mathbb{R}^{m\times n}$, and let $f^{gen} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be the generalized matrix function* (5)*. Then*

1. $[f^{gen}(A)]^T = f^{gen}(A^T)$,
2. *Let the matrices $X \in \mathbb{R}^{m\times m}$ and $Y \in \mathbb{R}^{n\times n}$ be orthogonal. Then $f^{gen}(XAY) = Xf^{gen}(A)Y$.*
3. $f^{gen}(A) = f(\sqrt{AA^T})(\sqrt{AA^T})^\dagger A = A(\sqrt{A^TA})^\dagger f(\sqrt{A^TA})$,

where $M^\dagger$ is the Moore-Penrose pseudoinverse of M.

Proof The first two properties are shown by Arrigo et al. [1], the last one by Ben-Israel and Greville [3].

The analogue of formula (5) for tensors recently was introduced by Miao et al. [32]. To define these functions, we require the t-CSVD decomposition of a rectangular tensor.

Lemma 2 *([21]) The tensor $A \in \mathbb{R}^{m\times n\times p}$ can be factored as*

$$\mathcal{A} = \mathcal{U}_r * \mathcal{S}_r * \mathcal{Q}_r^T, \tag{6}$$

where $\mathcal{U}_r \in \mathbb{R}^{m\times r\times p}$ and $\mathcal{Q}_r \in \mathbb{R}^{n\times r\times p}$ are orthogonal tensors, and $\mathcal{S}_r \in \mathbb{R}^{r\times r\times p}$ is an f-diagonal tensor. The parameter $r = \max\{r_1, r_2, \ldots r_p\}$, where $r_j = \text{rank}(\Sigma_j)$ denotes the tubal-rank of $\mathcal{A}$ based on the t-product.

The block circulant matrices $\texttt{bcirc}(\mathcal{Q}_r)$, $\texttt{bcirc}(\mathcal{S}_r)$, and $\texttt{bcirc}(\mathcal{U}_r)$ can be block diagonalized by the FFT as follows:

$$\texttt{bcirc}(\mathcal{Q}_r) = (F_p^H \otimes I_n) \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_p \end{bmatrix} (F_p \otimes I_r), \text{ where } \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_p \end{bmatrix} = \texttt{bdiag}(\text{fft}(\mathcal{Q}_r, [\,], 3)), \tag{7}$$

$$\texttt{bcirc}(\mathcal{S}_r) = (F_p^H \otimes I_r) \begin{bmatrix} \Sigma_1 & & & \\ & \Sigma_2 & & \\ & & \ddots & \\ & & & \Sigma_p \end{bmatrix} (F_p \otimes I_r), \text{ where } \begin{bmatrix} \Sigma_1 & & & \\ & \Sigma_2 & & \\ & & \ddots & \\ & & & \Sigma_p \end{bmatrix} = \texttt{bdiag}(\text{fft}(\mathcal{S}_r, [\,], 3)), \tag{8}$$

and

$$\texttt{bcirc}(\mathcal{U}_r) = (F_p^H \otimes I_m) \begin{bmatrix} U_1 & & & \\ & U_2 & & \\ & & \ddots & \\ & & & U_p \end{bmatrix} (F_p \otimes I_r), \text{ where } \begin{bmatrix} U_1 & & & \\ & U_2 & & \\ & & \ddots & \\ & & & U_p \end{bmatrix} = \texttt{bdiag}(\text{fft}(\mathcal{U}_r, [\,], 3)),$$

with $\Sigma_j \in \mathbb{R}^{r\times r}$, $U_j \in \mathbb{R}^{m\times r}$, and $Q_j \in \mathbb{R}^{n\times r}$, for $j = 1, 2, \ldots, p$. Here the function `bdiag` applied to a tensor $\mathcal{D} \in \mathbb{R}^{m\times n\times p}$ gives a block diagonal matrix, whose diagonal blocks are the frontal slices $\mathcal{D}^{(i)}$, $i = 1, 2, \ldots, p$, of $\mathcal{D}$, i.e.,

$$\texttt{bdiag}(\mathcal{D}) = \begin{bmatrix} D^{(1)} & & & \\ & D^{(2)} & & \\ & & \ddots & \\ & & & D^{(p)} \end{bmatrix}.$$

Let the $\sigma_i^{(j)} \in \mathbb{R}$ be the diagonal entries of the matrix Σ_j, ordered so that $\sigma_1 \geq \sigma_2, \geq \ldots \geq \sigma_p \geq 0$, and let $q_i^{(j)}$ denote the ith column of the matrix Q_j, i.e.,

$$\begin{aligned} \Sigma_j &= \text{diag}[\sigma_1^{(j)}, \sigma_2^{(j)}, \ldots, \sigma_r^{(j)}], \\ Q_j &= [q_1^{(j)}, q_2^{(j)}, \ldots, q_r^{(j)}]. \end{aligned} \tag{9}$$

We may have $\sigma_i^{(j)} = 0$ for some i and j, since $r = \max\{r_1, r_2, \ldots, r_p\}$.

Consider the matrix $A \in \mathbb{R}^{m\times n}$. The square root of the positive eigenvalues of the matrix $A^T A \in \mathbb{R}^{n\times n}$ are the nonvanishing singular values of A. This result extends to tensors. According to (6), the tensor $\mathcal{A}^T * \mathcal{A}$ has the spectral decomposition

$$\mathcal{A}^T * \mathcal{A} = \mathcal{Q}_r * (\mathcal{S}_r^T * \mathcal{S}_r) * \mathcal{Q}_r^T. \tag{10}$$

Thus, the set of t-eigenvalues of $\mathcal{A}^T * \mathcal{A}$ can be defined as the set of eigenvalues of $\texttt{bcirc}(\mathcal{S}_r^T * \mathcal{S}_r)$, where

$$\text{spec}(\texttt{bcirc}(\mathcal{S}_r^T * \mathcal{S}_r)) = \{(\sigma_i^{(j)})^2, \quad 1 \leq j \leq p, \quad 1 \leq i \leq r\}.$$

Here the operator spec extracts the spectrum of the matrix. The quantities $\sigma_i^{(j)}$ are the t-singular values of the tensor $\mathcal{A}$. We are now in a position to describe the generalized tensor function defined by Miao et al. [32].

Proposition 1 *([32]) Let the third-order tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$ have the factorization* (6)*, and let the function $f : \mathbb{R} \to \mathbb{R}$ be defined on the nonnegative real axis. Introduce the function $f^{gen} : \mathbb{R}^{m\times n\times p} \to \mathbb{R}^{m\times n\times p}$ by*

$$f^{gen}(\mathcal{A}) = \mathcal{U}_r * \widetilde{f}(\mathcal{S}_r) * \mathcal{Q}_r^T, \tag{11}$$

where

$$\widetilde{f}(\mathcal{S}_r) = \texttt{bcirc}^{-1}\left((F_p^H \otimes I_r)\begin{bmatrix}\widetilde{f}(\Sigma_1) & & & \\ & \widetilde{f}(\Sigma_2) & & \\ & & \ddots & \\ & & & \widetilde{f}(\Sigma_p)\end{bmatrix}(F_p \otimes I_r)\right) \tag{12}$$

and

$$\widetilde{f}(\Sigma_i) = f\left(\sqrt{\Sigma_i\Sigma_i^T}\right)\left(\sqrt{\Sigma_i\Sigma_i^T}\right)^{\dagger}\Sigma_i = \Sigma_i\left(\sqrt{\Sigma_i^T\Sigma_i}\right)^{\dagger} f\left(\sqrt{\Sigma_i^T\Sigma_i}\right), \quad i = 1, 2, \ldots, p.$$

The matrices $\Sigma_i \in \mathbb{R}^{r\times r}$ are defined by (8)*, and $M^{\dagger}$ denotes the Moore-Penrose pseudoinverse of the matrix M.*

Lemma 3 *([32]) Consider the third-order tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$, let $f : \mathbb{R} \to \mathbb{R}$, and let $f^{gen} : \mathbb{R}^{m\times n\times p} \to \mathbb{R}^{m\times n\times p}$ be the corresponding generalized function for third-order tensors. Then*

1. $[f^{gen}(\mathcal{A})]^T = [f^{gen}(\mathcal{A}^T)]$.
2. *Let $\mathcal{P} \in \mathbb{R}^{m\times m\times p}$ and $\mathcal{Q} \in \mathbb{R}^{n\times n\times p}$ be orthogonal tensors. Then $f^{gen}(\mathcal{P}*\mathcal{A}*\mathcal{Q}) = \mathcal{P} * f^{gen}(\mathcal{A}) * \mathcal{Q}$.*
3. $f^{gen}(\mathcal{A}) = f(\sqrt{\mathcal{A}*\mathcal{A}^T}) * (\sqrt{\mathcal{A}*\mathcal{A}^T})^{\dagger} * \mathcal{A} = \mathcal{A} * (\sqrt{\mathcal{A}^T*\mathcal{A}})^{\dagger} * f(\sqrt{\mathcal{A}^T*\mathcal{A}})$.

3 Approximation of $\mathbf{trace_{(1)}}(\boldsymbol{\mathcal{W}^T * f^{gen}(\mathcal{A}) * \mathcal{V}})$

The evaluation of generalized tensor functions (11) requires the tensor singular value decomposition (6). However, when m, n, and p are large, the computation of this decomposition may be unfeasible or unattractive due to the large computational burden. Fortunately, many applications do not require the evaluation of the whole tensor function $f^{gen}(\mathcal{A})$. Instead, it may suffice to determine an estimate of quantities of the form

$$\mathbb{I}(f) := \text{trace}_{(1)}(\mathcal{W}^T * f^{gen}(\mathcal{A}) * \mathcal{V}), \tag{13}$$

where $\mathcal{W} \in \mathbb{R}^{m\times s\times p}$ and $\mathcal{V} \in \mathbb{R}^{n\times s\times p}$, with $s \ll \min\{m, n\}$.

According to the third equation of Lemma 3, the expression (13) can be written as bilinear forms involving functions of the tensors $\mathcal{A} * \mathcal{A}^T$ and $\mathcal{A}^T * \mathcal{A}$. We have

$$\begin{aligned}\text{trace}_{(1)}(\mathcal{W}^T * f^{gen}(\mathcal{A}) * \mathcal{V}) &= \text{trace}_{(1)}(\widetilde{\mathcal{W}}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V}) &(14)\\ &= \text{trace}_{(1)}(\mathcal{W} * g(\mathcal{A} * \mathcal{A}^T) * \widetilde{\mathcal{V}}), &(15)\end{aligned}$$

where $\widetilde{\mathcal{V}} = \mathcal{A} * \mathcal{V}$, $\widetilde{\mathcal{W}} = \mathcal{A}^T * \mathcal{W}$, and $g(t) = (\sqrt{t})^{-1} f(\sqrt{t})$.

We will focus on the expression in the right-hand side of (14); the discussion for the expression (15) proceeds similarly. First, note that if $\widetilde{\mathcal{W}} \neq \mathcal{V}$, then we can use the polarization identity, see, e.g., [11,15],

$$\begin{aligned}\text{trace}_{(1)}(\widetilde{\mathcal{W}}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V}) = \frac{1}{4}[&\text{trace}_{(1)}((\widetilde{\mathcal{W}} + \mathcal{V})^T * (g(\mathcal{A}^T * \mathcal{A}) * (\widetilde{\mathcal{W}} + \mathcal{V})))\\ &- \text{trace}_{(1)}((\widetilde{\mathcal{W}} - \mathcal{V})^T * (g(\mathcal{A}^T * \mathcal{A}) * (\widetilde{\mathcal{W}} - \mathcal{V})))],\end{aligned}$$

and therefore limit ourselves to compute approximations in the case when $\widetilde{\mathcal{W}} = \mathcal{V}$. Thus, it suffices to compute approximations of expressions of the form

$$I_s(g) := \mathrm{trace}_{(1)}(\mathcal{V}^T * (g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V})), \tag{16}$$

where $\mathcal{A} \in \mathbb{R}^{n\times n\times p}$, $\mathcal{V} \in \mathbb{R}^{n\times s\times p}$, and $g(\mathcal{A}^T * \mathcal{A}) = (\sqrt{\mathcal{A}^T * \mathcal{A}})^\dagger * f(\sqrt{\mathcal{A}^T * \mathcal{A}})$. To this end, we express (16) as a Riemann-Stieltjes integral by using the spectral factorization of $\mathcal{A}^T * \mathcal{A}$.

Proposition 2 *Let the $\sigma_i^{(j)}$ be t-singular values of $\mathcal{A}$ and let the vectors $q_i^{(j)} \in \mathbb{R}^n$ be defined by* (9)*. The matrix V_j denotes the jth frontal slice of the discrete Fourier transformation of $\mathcal{V}$. Let the function $f : \mathbb{R} \to \mathbb{R}$ satisfy $f(0) = 0$. Then the expression* (16) *can be written as the Riemann-Stieltjes integral*

$$I_s(g) = \sum_{j=1}^{p}\sum_{i=1}^{r} \frac{f(\sigma_i^{(j)})}{\sigma_i^{(j)}} \mathrm{trace}_{(1)}(\widetilde{V}_i^{(j)}) = \int_{\sigma_{min}^2}^{\sigma_{max}^2} g(t)d\alpha(t), \tag{17}$$

where

$$r = \max(\mathrm{rank}(\Sigma_j)), \quad \sigma_{max} = \max_{1\le i\le r}\{\sigma_i^{(j)}\}, \quad \sigma_{min} = \min_{1\le i\le r}\{\sigma_i^{(j)}\}, \quad j = 1, 2, \dots p,$$

and

$$\mathtt{bcirc}(\widetilde{V}_i^{(j)}) = (F_p^H \otimes I_s)(E_j V_j^T q_i^{(j)} q_i^{(j),T} V_j E_j^T)(F_p \otimes I_s).$$

The matrix $E_j \in \mathbb{R}^{ps\times s}$ is made up of the columns $(j-1)p+1, (j-1)p+2, \dots, jp$ of the identity matrix $I_{ps} \in \mathbb{R}^{ps\times ps}$ and $\alpha(\lambda)$ is a nondecreasing piece-wise constant function with possible discontinuities at the positive eigenvalues $(\sigma_i^{(j)})^2$ of $\mathtt{bcirc}(\mathcal{A}^T * \mathcal{A})$; $d\alpha(\lambda)$ is the associated measure.

Proof It follows from (10) that

$$\mathcal{A}^T * \mathcal{A} = \mathcal{Q}_r * (\mathcal{S}_r^T * \mathcal{S}_r) * \mathcal{Q}_r^T.$$

By the definition (11) of generalized tensor functions, we obtain

$$\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V} = \mathcal{V}^T * \mathcal{Q}_r * g(\mathcal{S}_r^T * \mathcal{S}_r) * \mathcal{Q}_r^T * \mathcal{V}.$$

Moreover,

$$\mathtt{bcirc}(\mathcal{V}) = (F_p^H \otimes I_n) \begin{bmatrix} V_1 & & & \\ & V_2 & & \\ & & \ddots & \\ & & & V_p \end{bmatrix} (F_p \otimes I_s). \tag{18}$$

Using (7), (18), and the orthogonality of the matrix F_p, we get

$$\mathtt{bcirc}(\mathcal{V}^T * \mathcal{Q}_r) = \mathtt{bcirc}(\mathcal{V})^T\, \mathtt{bcirc}(\mathcal{Q}_r) = (F_p^H \otimes I_s) \begin{bmatrix} V_1^T Q_1 & & & \\ & V_2^T Q_2 & & \\ & & \ddots & \\ & & & V_p^T Q_p \end{bmatrix} (F_p \otimes I_s).$$

It follows from (12) that the matrix $\texttt{bcirc}(g(\mathcal{S}_r^T * \mathcal{S}_r)) \in \mathbb{C}^{rp\times rp}$ can be written as

$$\texttt{bcirc}(g(\mathcal{S}_r^T * \mathcal{S}_r)) = (F_p^H \otimes I_r) \begin{bmatrix} g(\Sigma_1^T\Sigma_1) & & & \\ & g(\Sigma_2^T\Sigma_2) & & \\ & & \ddots & \\ & & & g(\Sigma_p^T\Sigma_p) \end{bmatrix} (F_p \otimes I_r).$$

Using the above relations, we obtain

$$\begin{aligned}\texttt{bcirc}(\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V}) &= \texttt{bcirc}(\mathcal{V}^T * \mathcal{Q}_r * g(\mathcal{S}_r^T * \mathcal{S}_r) * \mathcal{Q}_r^T * \mathcal{V}) \\ &= \texttt{bcirc}(\mathcal{V}^T * \mathcal{Q}_r)\,\texttt{bcirc}(g(\mathcal{S}_r^T * \mathcal{S}_r))\,\texttt{bcirc}(\mathcal{Q}_r^T * \mathcal{V}) \\ &= (F_p^H \otimes I_s) \begin{bmatrix} V_1^TQ_1g(\Sigma_1^T\Sigma_1)Q_1^TV_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & V_p^TQ_pg(\Sigma_p^T\Sigma_p)Q_p^TV_p \end{bmatrix} (F_p \otimes I_s).\end{aligned}$$

Since $f(0) = 0$, we have

$$V_j^TQ_jg(\Sigma_j^T\Sigma_j)Q_j^TV_j = \sum_{i=1}^{r} \frac{f(\sigma_i^{(j)})}{\sigma_i^{(j)}} V_j^T q_i^{(j)} q_i^{(j),T} V_j, \quad \text{for } j = 1, 2, \ldots, p.$$

It follows that

$$\texttt{bcirc}(\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V}) = (F_p^H \otimes I_s) \begin{bmatrix} \sum_{i=1}^r \frac{f(\sigma_i^{(1)})}{\sigma_i^{(1)}} V_1^T q_i^{(1)} q_i^{(1),T} V_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \sum_{i=1}^r \frac{f(\sigma_i^{(p)})}{\sigma_i^{(p)}} V_p^T q_i^{(p)} q_i^{(p),T} V_p \end{bmatrix} (F_p \otimes I_s).$$

The above equation can be written as

$$\begin{aligned}\texttt{bcirc}(\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V}) &= \sum_{j=1}^{p}\sum_{i=1}^{r} \frac{f(\sigma_i^{(j)})}{\sigma_i^{(j)}} (F_p^H \otimes I_s) E_j V_j^T q_i^{(j)} q_i^{(j),T} V_j E_j^T (F_p \otimes I_s) \\ &= \sum_{j=1}^{p}\sum_{i=1}^{r} \frac{f(\sigma_i^{(j)})}{\sigma_i^{(j)}}\, \texttt{bcirc}(\widetilde{V}_i^{(j)}).\end{aligned}$$

Using the linearity of the operator $\texttt{bcirc}^{-1}$, we obtain

$$\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V} = \sum_{j=1}^{p}\sum_{i=1}^{r} \frac{f(\sigma_i^{(j)})}{\sigma_i^{(j)}} \widetilde{V}_i^{(j)}.$$

Finally, we get

$$\text{trace}_{(1)}(\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V}) = \sum_{j=1}^{p}\sum_{i=1}^{r} \frac{f(\sigma_i^{(j)})}{\sigma_i^{(j)}} \text{trace}_{(1)}(\widetilde{V}_i^{(j)}) = \int_{\sigma_{min}^2}^{\sigma_{max}^2} g(t)d\alpha(t).$$

We remark that the condition $f(0) = 0$ in Proposition 2 is required only when some singular values $\sigma_i^{(j)}$ vanish. The following subsections describe two approaches to approximate generalized tensor functions.

3.1 The tensor t-global Lanczos method and Gauss quadrature

Let $\mathscr{A}$ be a large symmetric tensor. We describe the tensor t-global Lanczos method for generating an orthonormal basis for the tensor global Krylov subspace $\mathcal{K}_m^g(\mathscr{A}, \mathcal{V})$. In the computed examples reported in Section 4, we let $\mathscr{A} = \mathcal{A}^T * \mathcal{A} \in \mathbb{R}^{n\times n\times p}$, where $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$, but the method can be applied to reduce an arbitrary large symmetric tensor to a small symmetric tensor.

The tensor t-global Lanczos method is a special case of the t-global Arnoldi method, which was introduced in [7]. The latter method is designed to determine an F-orthonormal basis for a t-global Krylov subspace when $\mathcal{A} \in \mathbb{R}^{n\times n\times p}$ is a nonsymmetric tensor. The generation of a new element $\mathcal{V}_{j+1}$ of the F-orthonormal basis by the t-global Arnoldi method requires explicit orthogonalization against all already available basis elements, say $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_j$, by using a recurrence relation of the form

$$\begin{aligned} \mathcal{V}_1 &= \frac{\mathcal{V}}{\|\mathcal{V}\|_F}, \\ h_{j+1,j}\mathcal{V}_{j+1} &= \mathscr{A} * \mathcal{V}_j - \sum_{i=1}^{j} h_{i,j}\mathcal{V}_i, \quad j = 1, 2, \ldots, k. \end{aligned} \tag{19}$$

The coefficients $h_{i,j}$ are determined so that the tensors $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{j+1}$ are F-orthonormal. Specifically,

$$h_{i,j} = \langle \mathscr{A} * \mathcal{V}_j, \mathcal{V}_i \rangle, \quad 1 \le i \le j, \quad h_{j+1,j} = \|\mathscr{A} * \mathcal{V}_j - \sum_{i=1}^{j} h_{i,j}\mathcal{V}_i\|_F.$$

The following result is the foundation of the tensor t-global Lanczos method.

Proposition 3 *Apply k steps of the t-global Arnoldi method to a t-symmetric tensor $\mathscr{A}$ with initial tensor $\mathcal{V} \neq \mathcal{O}$. Then generically the coefficients $h_{i,j}$ generated by the algorithm satisfy*

$$h_{i,j} = 0 \quad \textit{for} \quad 1 \le i < j - 1, \tag{20}$$

$$h_{j,j+1} = h_{j+1,j}, \tag{21}$$

for $j = 1, 2, \ldots, k$.

Proof We first show (20). Since the tensor $\mathscr{A}$ is t-symmetric, we have

$$h_{i,j} = \langle \mathscr{A}\mathcal{V}_j, \mathcal{V}_i \rangle = \langle \mathscr{A}\mathcal{V}_i, \mathcal{V}_j \rangle.$$

It follows from (19) that

$$\mathscr{A} * \mathcal{V}_i \in \operatorname{span}\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{i+1}\}.$$

Due to the F-orthonormality of $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{j+1}$ and (19), we obtain $h_{i,j} = \langle \mathcal{A}\mathcal{V}_i, \mathcal{V}_j \rangle = 0$ for $1 \le i < j - 1$. Equality (21) is a consequence of

$$h_{j,j+1} = \langle \mathscr{A} * \mathcal{V}_{j+1}, \mathcal{V}_j \rangle = \langle \mathscr{A} * \mathcal{V}_j, \mathcal{V}_{j+1} \rangle = h_{j+1,j}.$$

We are in a position to describe the tensor t-global Lanczos algorithm. This algorithm computes an F-orthonormal basis $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{k+1}\}$ by a three-term recurrence relation. It can be written as follows: Let $\mathcal{V} \in \mathbb{R}^{n\times s\times p}\backslash\{\mathcal{O}\}$ and define

$$\mathcal{V}_1 := \mathcal{V}/\beta_0, \text{ with } \beta_0 = \|\mathcal{V}\|_F. \tag{22}$$

The t-global Lanczos algorithm is determined by the recursion relations

$$\beta_j \mathcal{V}_{j+1} := \mathscr{A} * \mathcal{V}_j - \alpha_j \mathcal{V}_j - \beta_{j-1}\mathcal{V}_{j-1}, \quad j = 1, 2, \ldots, k, \tag{23}$$

with

$$\alpha_j = \langle \mathscr{A} * \mathcal{V}_j, \mathcal{V}_j\rangle, \quad \beta_j = \|\mathscr{A} * \mathcal{V}_j - \alpha_j \mathcal{V}_j - \beta_{j-1}\mathcal{V}_{j-1}\|_F.$$

The β_j are normalization coefficients such that $\|\mathcal{V}_{j+1}\|_F = 1$. The tensor t-global Lanczos algorithm is summarized in Algorithm 2.

Algorithm 2 The tensor t-global Lanczos algorithm

Input: t-symmetric tensor $\mathscr{A} \in \mathbb{R}^{n\times n\times p}$, initial tensor $\mathcal{V} \in \mathbb{R}^{n\times s\times p}\backslash\{\mathcal{O}\}$, and the number of steps k.

1. $\beta_0 = \|\mathcal{V}\|_F$; $\mathcal{V}_1 = \mathcal{V}/\beta_0$; $\mathcal{V}_0 = \mathcal{O}_{n\times s\times p}$;
2. for $j = 1 : k$
   (a) $\widetilde{\mathcal{V}}_j = \mathscr{A} * \mathcal{V}_j - \beta_{j-1}\mathcal{V}_{j-1}$;
   (b) $\alpha_j = \langle \mathcal{V}_j, \widetilde{\mathcal{V}}_j\rangle$;
   (c) $\widetilde{\mathcal{V}}_j = \widetilde{\mathcal{V}}_j - \alpha_j \mathcal{V}_j$;
   (d) $\beta_j = \|\widetilde{\mathcal{V}}_j\|_F$;
   (e) $\mathcal{V}_{j+1} = \widetilde{\mathcal{V}}_j/\beta_j$;
3. end

Output: The t-global Lanczos decomposition (24).

Algorithm 2 computes an F-orthonormal tensor basis $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{k+1}\}$ with $\mathcal{V}_j \in \mathbb{R}^{n\times s\times p}$, and the scalars $\{\alpha_1, \alpha_2, \ldots, \alpha_k; \beta_1, \beta_2, \ldots, \beta_k\}$, assuming that all coefficients β_j generated are positive. This is the generic situation.

We next discuss some useful properties of Algorithm 2. It follows from the recursion formulas of the algorithm that generically the algorithm yields the decomposition

$$\mathscr{A} * \mathbb{V}_k = \mathbb{V}_k \circledast \mathbb{T}_k + \beta_k[\mathcal{O}_{n\times s\times p}, \ldots, \mathcal{O}_{n\times s\times p}, \mathcal{V}_{k+1}], \tag{24}$$

where $\mathbb{V}_k = [\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k] \in \mathbb{R}^{n\times ks\times p}$ is made up of orthonormal lateral tensor slices $\mathcal{V}_j$, with

$$\mathbb{V}_k^T \diamond \mathbb{V}_k = I_k, \quad \mathbb{T}_k = \text{tridiag}([\alpha_1, \alpha_2, \ldots, \alpha_k], [\beta_1, \beta_2, \ldots, \beta_{k-1}]),$$

and the product $\circledast$ is defined by

$$\mathbb{V}_k \circledast \mathbb{T}_k = [\mathbb{V}_k \circledast T_{\cdot,1}, \mathbb{V}_k \circledast T_{\cdot,2}, \ldots, \mathbb{V}_k \circledast T_{\cdot,k}] \in \mathbb{R}^{n\times ks\times p}.$$

Here $T_{\cdot,j}$ denotes the jth column of $\mathbb{T}_k$ and

$$\mathbb{V}_k \circledast y = \sum_{j=1}^{k} y_j \mathcal{V}_j, \quad y = [y_1, y_2, \ldots, y_k]^T \in \mathbb{R}^k.$$

Lemma 4 *Let $Z = [Z_1, Z_2, \ldots, Z_k] \in \mathbb{R}^{n\times ks\times p}$ be made up of tensors $Z_i \in \mathbb{R}^{n\times s\times p}$, and let $T, S \in \mathbb{R}^{k\times k}$. Then*

$$(Z \circledast T) \circledast S = Z \circledast (TS).$$

Proof Let $S_{\cdot,j}$ and $T_{\cdot,j}$ denote the jth columns of the matrices S and T, respectively, and let $s_{i,j}$ and $t_{i,j}$ be (i,j)th entries of S and T, respectively. Using the definition of the $\circledast$ product, we obtain

$$\begin{aligned}
[Z \circledast T] \circledast S &= [(Z \circledast T) \circledast S_{\cdot,1}, \ldots, (Z \circledast T) \circledast S_{\cdot,k}] \\
&= [\sum_{j=1}^{k} (Z \circledast T_{\cdot,j}) s_{j,1}, \ldots, \sum_{j=1}^{k} (Z \circledast T_{\cdot,j}) s_{j,k}] \\
&= [\sum_{j=1}^{k} (\sum_{l=1}^{k} t_{l,j} Z_l) s_{j,1}, \ldots, \sum_{j=1}^{k} (\sum_{l=1}^{k} t_{l,j} Z_l) s_{j,k}] \\
&= [\sum_{j=1}^{k} \sum_{l=1}^{k} t_{l,j} s_{j,1} Z_l, \ldots, \sum_{j=1}^{k} \sum_{l=1}^{k} t_{l,j} s_{j,k} Z_l] \\
&= [\sum_{l=1}^{k} \sum_{j=1}^{k} t_{l,j} s_{j,1} Z_l, \ldots, \sum_{l=1}^{k} \sum_{j=1}^{k} t_{l,j} s_{j,k} Z_l] \\
&= [\sum_{l=1}^{k} [TS]_{l,1} Z_l, \ldots, \sum_{l=1}^{k} [TS]_{l,k} Z_l] \\
&= [Z \circledast (TS)_{\cdot,1}, \ldots, Z \circledast (TS)_{\cdot,k}] \\
&= Z \circledast (TS).
\end{aligned}$$

The recursion formulas of Algorithm 2 show that

$$\mathcal{V}_j = p_{j-1}(\mathscr{A}) * \mathcal{V}, \quad j = 1, 2, \ldots, k+1, \tag{25}$$

for certain polynomials p_{j-1} of degree $j-1$. These polynomials satisfy the same recursion relation as the tensors $\mathcal{V}_j$. The following proposition provides the details.

Proposition 4 *Let $\mathscr{A}$ be a t-symmetric tensor and let the coefficients α_j and β_j be those in* (22) *and* (23)*. Then the sequence of polynomials $p_0, p_1, \ldots, p_k$ defined by* (25) *are orthonormal with respect to the bilinear form*

$$(p, q) = \langle p(\mathscr{A}) * \mathcal{V}, q(\mathscr{A}) * \mathcal{V} \rangle = \int_{\sigma_{min}^2}^{\sigma_{max}^2} p(\lambda) q(\lambda) d\alpha(\lambda),$$

where the limits of integration $\sigma_{\min}^2$ and $\sigma_{\max}^2$, and the measure $d\alpha$ are defined in (17)*. These polynomials satisfy the three-term recurrence relation*

$$\begin{aligned}
\beta_1 p_1(\lambda) &= (\lambda - \alpha_1) p_0(\lambda), \quad p_0(\lambda) = 1/\beta_0, \\
\beta_{j+1} p_{j+1} &= (\lambda - \alpha_{j+1}) p_j(\lambda) - \beta_j p_{j-1}(\lambda), \quad j = 1, 2, \ldots, k-1,
\end{aligned}$$

where

$$\alpha_j = (p_{j-1}, \lambda p_{j-1}),$$

and the $\beta_j > 0$ are determined by the requirements that $(p_j, p_j) = 1$.

We turn to Gauss-type quadrature rules associated with the measure $d\alpha$ in (17). Introduce the spectral factorization of the matrix $\mathbb{T}_k$ in (24),

$$\mathbb{T}_k = P_k S_k P_k^T,$$

where $S_k = \mathrm{diag}[\theta_1^{(k)}, \theta_2^{(k)}, \ldots, \theta_k^{(k)}]$ and the matrix $P_k \in \mathbb{R}^{k\times k}$ is orthogonal. It is well known that the weights of the k-node Gauss rule associated with the measure $d\alpha$ are given by $w_j = (e_1^T P_k e_j)^2$ for $j = 1, 2, \ldots, k$, and the nodes of this quadrature rule are the eigenvalues $\theta_j^{(k)}$, $j = 1, 2, \ldots, k$; see, e.g., [12,13,15] for details. The k-node Gauss quadrature rule for the approximation of the expression (16) can be written as

$$\begin{aligned}\mathcal{G}_k^{Lan}(g) &:= \|\mathcal{V}\|_F^2 e_1^T P_k g(S_k) P_k^T e_1 = \|\mathcal{V}\|_F^2 e_1^T g(\mathbb{T}_k) e_1 \\ &= \|\mathcal{V}\|_F^2 e_1^T (\sqrt{\mathbb{T}_k})^\dagger f(\sqrt{\mathbb{T}_k}) e_1, \end{aligned} \tag{26}$$

where $\theta_i^{(k)}$ denotes the ith eigenvalue of $\mathbb{T}_k$, and e_1 is the first column of the identity matrix $I_k \in \mathbb{R}^{k\times k}$; see [15].

A $(k+1)$-node Gauss-Radau quadrature rule with a fixed node ξ smaller than or equal to σ_{min}^2 or larger than or equal to σ_{max}^2 can be computed by using the spectral factorization of the matrix

$$\mathbb{T}_{k+1,\xi} := \begin{bmatrix} \mathbb{T}_k & \beta_k e_k \\ \beta_k e_k^T & \widetilde{\alpha}_\xi \end{bmatrix} \in \mathbb{R}^{(k+1)\times(k+1)},$$

where $e_k = [0, \ldots, 0, 1] \in \mathbb{R}^k$. The entry $\widetilde{\alpha}_\xi$ is determined so that the matrix $\mathbb{T}_{k+1,\xi}$ has an eigenvalue at ξ. It can be shown that $\widetilde{\alpha}_\xi$ satisfies

$$\widetilde{\alpha}_\xi = \xi + \beta_k^2 e_k^T (\mathbb{T}_k - \xi I_k)^{-1} e_k;$$

see, e.g., [12,13,15] for details.

Analogously to (26), the $(k+1)$-node Gauss-Radau quadrature rule can be expressed as

$$\mathcal{R}_{k+1}^{Lan,\xi}(g) := \|\mathcal{V}\|_F^2 \widetilde{e}_1^T g(\mathbb{T}_{k+1,\xi}) \widetilde{e}_1 = \|\mathcal{V}\|_F^2 \widetilde{e}_1^T (\sqrt{\mathbb{T}_{k+1,\xi}})^\dagger f(\sqrt{\mathbb{T}_{k+1,\xi}}) \widetilde{e}_1, \tag{27}$$

where $\widetilde{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{k+1}$; see [15].

Let $\theta_1^{(k)}, \theta_2^{(k)}, \ldots, \theta_k^{(k)}$ denote the nodes of the Gauss rule (26), and assume that the integrand g is $2k$ times continuously differentiable in the interval of integration $[\sigma_{\min}^2, \sigma_{\max}^2]$. Then the remainder term for this rule is given by

$$\mathcal{I}_s(g) - \mathcal{G}_k^{Lan}(g) = \frac{g^{(2k)}(\widetilde{\theta})}{(2k)!} \int_{\sigma_{min}^2}^{\sigma_{max}^2} s^2(\lambda) d\alpha(\lambda), \tag{28}$$

where $\widetilde{\theta} \in [\sigma_{\min}^2, \sigma_{\max}^2]$ and $s(\lambda) = (\lambda - \theta_1^{(k)}) \cdots (\lambda - \theta_k^{(k)})$; see, e.g., [13,15]. Similarly, let $\theta_{1,\xi}^{(k)}, \theta_{2,\xi}^{(k)}, \ldots, \theta_{k,\xi}^{(k)}$ denote the k "free" nodes of the Gauss-Radau quadrature rule (27). The remainder formula for this rule is given by

$$\mathcal{I}_s(g) - \mathcal{R}_{m+1}^{Lan,\xi}(g) = \frac{g^{(2k+1)}(\widetilde{\theta}_\xi)}{(2k+1)!} \int_{\sigma_{min}^2}^{\sigma_{max}^2} (\lambda - \xi) \widetilde{s}^2(\lambda) d\alpha(\lambda), \tag{29}$$

Table 1 Upper and lower bounds for $\mathcal{I}_s(g)$

$sign(g^{(2k)})$	$sign(g^{(2k+1)})$	Choice of ξ	Bounds
$g^{(2k)} > 0$	$g^{(2k+1)} > 0$	$\xi \geq \sigma_{\max}^2$	$\mathcal{G}_k^{Lan}(g) \leq \mathcal{I}_s(g) \leq \mathcal{R}_{k+1}^{Lan,\xi}(g)$
$g^{(2k)} > 0$	$g^{(2k+1)} < 0$	$\xi \leq \sigma_{\min}^2$	$\mathcal{G}_k^{Lan}(g) \leq \mathcal{I}_s(g) \leq \mathcal{R}_{k+1}^{Lan,\xi}(g)$
$g^{(2k)} < 0$	$g^{(2k+1)} > 0$	$\xi \leq \sigma_{\min}^2$	$\mathcal{R}_{k+1}^{Lan,\xi}(g) \leq \mathcal{I}_s(g) \leq \mathcal{G}_k^{Lan}(g)$
$g^{(2k)} < 0$	$g^{(2k+1)} < 0$	$\xi \geq \sigma_{\max}^2$	$\mathcal{R}_{k+1}^{Lan,\xi}(g) \leq \mathcal{I}_s(g) \leq \mathcal{G}_k^{Lan}(g)$

where the scalar $\widetilde{\theta}_\xi$ lives in the smallest open interval that contains the point set $\{\xi, \sigma_{\min}^2, \sigma_{\max}^2\}$, and $\widetilde{s}(\lambda) = (\lambda - \theta_{1,\xi}^{(k)}) \cdots (\lambda - \theta_{k,\xi}^{(k)})$; see [13,15]. When the derivatives $g^{(2k)}$ and $g^{(2k+1)}$ of the integrand g do not change sign in the interval of integration $[\sigma_{\min}^2, \sigma_{\max}^2]$, the Radau node ξ can be chosen $\xi \leq \sigma_{\min}^2$ or $\xi \geq \sigma_{\max}^2$ so that the remainder term (29) is of opposite sign as the remainder term (28) for standard Gauss quadrature. Then the Gauss and Gauss-Radau rules give quadrature errors of opposite sign and their values, therefore, bracket the exact value of the integral (16); see [15] for further details.

Table 1 displays the bounds obtained depending on the signs of the derivatives $g^{(2k)}$ and $g^{(2k+1)}$, and the allocation of ξ.

Algorithm 3 describes how an approximation of (16) can be computed by a pair of Gauss and Gauss-Radau quadrature rules using the t-tensor global Lanczos method. When the algorithm is applied to $\mathcal{A}^T * \mathcal{A}$, the evaluation of $\mathcal{A}^T * \mathcal{A} * \mathcal{V}_j$ in line [3.(a)] is carried out by computing the two products $u = \mathcal{A} * \mathcal{V}_j$ and $v = \mathcal{A}^T * u$.

We approximate the integral (16) by $U_{\rm app}(g) = [\mathcal{G}_k^{Lan}(g) + \mathcal{R}_{k+1}^{Lan,\xi}(g)]/2$ and estimate the error in $U_{\rm app}(g)$ by the difference

$$|\mathcal{G}_k^{Lan}(g) - \mathcal{R}_{k+1}^{Lan,\xi}(g)]|/|\mathcal{G}_k^{Lan}(g) + \mathcal{R}_{k+1}^{Lan,\xi}(g)]|.$$

Algorithm 3 Approximation of $\text{trace}_{(1)}(\mathcal{V}^T * g(\mathscr{A}) * \mathcal{V})$ by pairs of Gauss and Gauss-Radau quadrature rules using the t-tensor global Lanczos method.

Input: t-symmetric tensor $\mathscr{A} \in \mathbb{R}^{n \times n \times p}$, initial tensor $\mathcal{V} \in \mathbb{R}^{n \times s \times p}$, and function f.

1. Choose tolerance $\epsilon > 0$ and the maximum number of iterations I_{max}.
2. $\beta_0 = \|\mathcal{V}\|_F$; $\mathcal{V}_1 = \mathcal{V}/\beta_0$; $\mathcal{V}_0 = \mathcal{O}_{n \times s \times p}$;
3. for $j = 1 : I_{max}$
   (a) $\widetilde{\mathcal{V}}_j = \mathscr{A} * \mathcal{V}_j - \beta_{j-1}\mathcal{V}_{j-1}$;
   (b) $\alpha_j = \langle \mathcal{V}_j, \widetilde{\mathcal{V}}_j \rangle$; $\widetilde{\mathcal{V}}_j = \widetilde{\mathcal{V}}_j - \alpha_j \mathcal{V}_j$;
   (c) $\mathbb{H}_j = \text{tridiag}([\alpha_1, \ldots, \alpha_j], [\beta_1, \ldots, \beta_{j-1}])$;
   (d) $G_j(g) = e_1^T (\sqrt{\mathbb{T}_j})^\dagger f(\sqrt{\mathbb{T}_j}) e_1$;
   (e) $\beta_j = \|\widetilde{\mathcal{V}}_j\|_F$; $\mathcal{V}_{j+1} = \widetilde{\mathcal{V}}_{j+1}/\beta_j$;
   (f) $\tau_j = [0, \ldots, 0, 0, \beta_j]^T$;
   (g) $\widetilde{\alpha}_\xi = \xi + \tau_j^T (\mathbb{T}_j - \xi I_j)^{-1} \tau_j$;
   (h) Compute $\mathbb{T}_{j+1,\xi} = \text{tridiag}([\alpha_1, \ldots, \alpha_j, \widetilde{\alpha}_\xi], [\beta_1, \ldots, \beta_{j-1}, \beta_j])$;
   (i) $R_{j+1,\xi}(g) = \widetilde{e}_1^T (\sqrt{\mathbb{T}_{j+1,\xi}})^\dagger f(\sqrt{\mathbb{T}_{j+1,\xi}}) \widetilde{e}_1$;
   (j) if $|G_j(g) - R_{j+1,\xi}(g)|/|G_j(g) + R_{j+1,\xi}(g)| < \epsilon$
   $U_{\rm app}(g) = \beta_0^2 [G_j(g) + R_{j+1,\xi}(g)]/2$; Break;
   endif
4. endfor

Output: Approximation $U_{\rm app}(g)$ of $\text{trace}_{(1)}(\mathcal{V}^T * g(\mathscr{A}) * \mathcal{V})$ given by (16).

3.2 The tensor t-global Golub-Kahan bidiagonalization method and Gauss quadrature

This subsection describes the application of the tensor t-global Golub-Kahan algorithm to determining upper and lower bounds for the bilinear form (16). The tensor t-global Golub-Kahan algorithm has been described in [7]. Application of k steps of this algorithm to the tensor $\mathcal{A}$ with initial tensor $\mathcal{V}$ yields the decompositions

$$\begin{aligned} \mathcal{A} * \mathbb{V}_k &= \mathbb{W}_k \circledast B_k, \\ \mathcal{A}^T * \mathbb{W}_k &= \mathbb{V}_k \circledast B_k^T + \beta_k [\mathcal{O}_{n\times s\times p}, \ldots, \mathcal{O}_{n\times s\times p}, \mathcal{V}_{k+1}], \end{aligned} \tag{30}$$

where $B_k \in \mathbb{R}^{k\times k}$ is an upper bidiagonal matrix

$$B_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ 0 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & 0 & \alpha_k \end{bmatrix}. \tag{31}$$

The tensors

$$\mathbb{V}_k = [\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k], \quad \mathbb{W}_k = [\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_k], \tag{32}$$

are made up of F-orthonormal tensors $\mathcal{V}_j \in \mathbb{R}^{m\times s\times p}$ and $\mathcal{W}_j \in \mathbb{R}^{n\times s\times p}$, and

$$\mathcal{A} * \mathbb{V}_k = [\mathcal{A} * \mathcal{V}_1, \mathcal{A} * \mathcal{V}_2, \ldots, \mathcal{A} * \mathcal{V}_k], \quad \mathcal{A}^T * \mathbb{W}_k = [\mathcal{A}^T * \mathcal{W}_1, \mathcal{A}^T * \mathcal{W}_2, \ldots, \mathcal{A}^T * \mathcal{W}_k].$$

The $\circledast$ product in (30) is defined in (24). The tensor t-global Golub-Kahan bidiagonalization algorithm is described by Algorithm 4.

Algorithm 4 The tensor t-global Golub-Kahan algorithm

Input: The tensors $\mathcal{A} \in \mathbb{R}^{m\times n\times p}, \mathcal{V} \in \mathbb{R}^{n\times s\times p}$, and an integer k.

1. Set $\beta_1 = \|\mathcal{V}\|_F$, $\mathcal{V}_1 = \mathcal{V}/\beta_1$, and $\beta_0 = 0$.
2. for $j = 1 : k$
   (a) $\widetilde{\mathcal{W}}_j = \mathcal{A} * \mathcal{V}_j - \beta_{j-1}\mathcal{W}_{j-1}$
   (b) $\alpha_j = \|\widetilde{\mathcal{W}}_j\|_F$
   (c) $\mathcal{W}_j = \widetilde{\mathcal{W}}_j/\alpha_j$
   (d) $\widetilde{\mathcal{V}}_{j+1} = \mathcal{A}^T * \mathcal{W}_j - \alpha_j\mathcal{V}_j$
   (e) $\beta_j = \|\widetilde{\mathcal{V}}_{j+1}\|_F$
   (f) $\mathcal{V}_{j+1} = \widetilde{\mathcal{V}}_{j+1}/\beta_j$
3. endfor

Output: The t-global Golub-Kahan decompositions (30).

Proposition 5 *Let $\mathbb{V}_k = [\mathcal{V}_1, \ldots, \mathcal{V}_k]$ be the tensor in* (32)*, and let the matrix B_k be defined by* (31)*. Then*

$$\mathcal{A}^T * \mathcal{A} * \mathbb{V}_k = \mathbb{V}_k \circledast (B_k^T B_k) + \beta_k\alpha_k [\mathcal{O}_{n\times s\times p}, \ldots, \mathcal{O}_{n\times s\times p}, \mathcal{V}_{k+1}].$$

Proof We obtain from relation (30) that

$$\mathcal{A}^T * \mathcal{A} * \mathbb{V}_k = \mathcal{A}^T * (\mathbb{W}_k \circledast B_k).$$

Set $B_k = [B_{\cdot,1}, \ldots, B_{\cdot,k}]$, where $B_{\cdot,j}$ denotes the jth column of B_k, and $b_{i,j}$ is the (i,j)th entry of B_k. Then

$$\begin{aligned}
\mathcal{A}^T * (\mathbb{W}_k \circledast B_k) &= [\mathcal{A}^T * (\mathbb{W}_k \circledast B_{\cdot,1}), \ldots, \mathcal{A}^T * (\mathbb{W}_k \circledast B_{\cdot,k})] \\
&= [\mathcal{A}^T * \sum_{j=1}^{k} B_{j,1}\mathcal{W}_j, \ldots, \mathcal{A}^T * \sum_{j=1}^{k} b_{j,k}\mathcal{W}_j] \\
&= [\sum_{j=1}^{k} b_{j,1}\mathcal{A}^T * \mathcal{W}_j, \ldots, \sum_{j=1}^{k} b_{j,k}\mathcal{A}^T * \mathcal{W}_j] \\
&= [(\mathcal{A}^T * \mathbb{W}_k) \circledast B_{\cdot,1}, \ldots, (\mathcal{A}^T * \mathbb{W}_k) \circledast B_{\cdot,k}] \\
&= (\mathcal{A}^T * \mathbb{W}_k) \circledast B_k.
\end{aligned}$$

It follows from (30) that

$$\begin{aligned}
\mathcal{A}^T * \mathcal{A} * \mathbb{V}_k &= [(\mathbb{V}_k \circledast B_k^T) + \beta_k[\mathcal{O}_{n\times s\times p}, \ldots, \mathcal{O}_{n\times s\times p}, \mathcal{V}_{k+1}]] \circledast B_k \\
&= (\mathbb{V}_k \circledast B_k^T) \circledast B_k + \beta_k[\mathcal{O}_{n\times s\times p}, \ldots, \mathcal{O}_{n\times s\times p}, \mathcal{V}_{k+1}] \circledast B_k,
\end{aligned}$$

and Lemma 4 yields

$$(\mathbb{V}_k \circledast B_k^T) \circledast B_k = \mathbb{V}_k \circledast B_k^T B_k.$$

Moreover,

$$[\mathcal{O}_{n\times s\times p}, \ldots, \mathcal{O}_{n\times s\times p}, \mathcal{V}_{k+1}] \circledast B_k = [b_{k,1}\mathcal{V}_{k+1}, \ldots, b_{k,k-1}\mathcal{V}_{k+1}, b_{k,k}\mathcal{V}_{k+1}].$$

Using the structure of B_k, i.e., the facts that $b_{k,1} = \cdots = b_{k,k-1} = 0$ and $b_{k,k} = \alpha_k$, we obtain

$$[\mathcal{O}_{n\times s\times p}, \ldots, \mathcal{O}_{n\times s\times p}, V_{k+1}] \circledast B_k = [O_{n\times s\times p}, \ldots, O_{n\times s\times p}, \alpha_k\mathcal{V}_{k+1}].$$

The matrix $B_k^T B_k$ is symmetric and tridiagonal, and coincides (in exact arithmetic) with the matrix $\mathbb{T}_k$ obtained when the Lanczos algorithm is applied to $\mathscr{A} = \mathcal{A}^T * \mathcal{A}$. Therefore, the quadratic form in (16) can be approximated by using a k-node Gauss quadrature rule

$$\mathcal{G}_k^{GK}(g) := \|\mathcal{V}\|_F^2 e_1^T g(B_k^T B_k) e_1 = \|\mathcal{V}\|_F^2 e_1^T (\sqrt{B_k^T B_k})^\dagger f(\sqrt{B_k^T B_k}) e_1. \tag{33}$$

Similarly as in our discussion above of the Gauss rule $\mathcal{G}_k^{Lan}(g)$, we have that if the derivative $g^{(2k)}$ is of constant sign in the interval of integration, $[\sigma_{\min}^2, \sigma_{\max}^2]$, then the Gauss rule (33) provides an upper or lower bound for (16). To obtain the other bound, we use a $(k+1)$-point Gauss-Radau quadrature rule with a fixed node ξ. This rule is defined by

$$\mathcal{R}_{k+1}^{GK,\xi}(g) := \|\mathcal{V}\|_F^2 \widetilde{e}_1^T g(\widehat{\mathbb{H}}_{k+1,\xi}) \widetilde{e}_1 = \|\mathcal{V}\|_F^2 \widetilde{e}_1^T (\sqrt{\widehat{\mathbb{H}}_{k+1,\xi}})^\dagger f(\sqrt{\widehat{\mathbb{H}}_{k+1,\xi}}) \widetilde{e}_1, \tag{34}$$

where

$$\widehat{\mathbb{H}}_{k+1,\xi} := \begin{bmatrix} B_k^T B_k & \beta_k\alpha_k e_k \\ \beta_k\alpha_k e_k^T & \widetilde{\omega}_\xi \end{bmatrix} \in \mathbb{R}^{(k+1)\times(k+1)}$$

Table 2 Upper and lower bounds for $\mathcal{I}_s(g)$.

$sign(g^{(2k)})$	$sign(g^{(2k+1)})$	Choice of ξ	Bounds
$g^{(2k)} > 0$	$g^{(2k+1)} > 0$	$\xi \geq \sigma_{\max}^2$	$\mathcal{G}_k^{GK}(g) \leq \mathcal{I}_s(g) \leq \mathcal{R}_{k+1}^{GK,\xi}(g)$
$g^{(2k)} > 0$	$g^{(2k+1)} < 0$	$\xi \leq \sigma_{\min}^2$	$\mathcal{G}_k^{GK}(g) \leq \mathcal{I}_s(g) \leq \mathcal{R}_{k+1}^{GK,\xi}(g)$
$g^{(2k)} < 0$	$g^{(2k+1)} > 0$	$\xi \leq \sigma_{\min}^2$	$\mathcal{R}_{k+1}^{GK,\xi}(g) \leq \mathcal{I}_s(g) \leq \mathcal{G}_k^{GK}(g)$
$g^{(2k)} < 0$	$g^{(2k+1)} < 0$	$\xi \geq \sigma_{\max}^2$	$\mathcal{R}_{k+1}^{GK,\xi}(g) \leq \mathcal{I}_s(g) \leq \mathcal{G}_k^{GK}(g)$

and

$$\widetilde{\omega}_\xi = \xi + (\beta_k \alpha_k)^2 e_k^T (B_k^T B_k - \xi I_k)^{-1} e_k.$$

The situations when Gauss and Gauss-Radau quadrature rules give upper or lower bounds of the integral $\mathcal{I}_s(g)$ are summarized in Table 2.

Algorithm 5 describes how an approximation of (16) can be computed by a pair of Gauss and Gauss-Radau quadrature rules using the tensor t-global Golub-Kahan bidiagonalization algorithm. Similarly as in Subsection 3.1, we approximate (16) by $U_{\text{app}}(g) = [\mathcal{G}_k^{GK}(g) + \mathcal{R}_{k+1}^{GK,\xi}(g)]/2$ and estimate the error in $U_{\text{app}}(g)$ by the difference

$$|\mathcal{G}_k^{GK}(g) - \mathcal{R}_{k+1}^{GK,\xi}(g)]|/|\mathcal{G}_k^{GK}(g) + \mathcal{R}_{k+1}^{GK,\xi}(g)]|.$$

Algorithm 5 Approximation of $\text{trace}_{(1)}(\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V})$ by a pair of Gauss and Gauss-Radau quadrature rules by using the tensor t-global Golub-Kahan method.

Input: T-symmetric tensor $\mathcal{A} \in \mathbb{R}^{m \times n \times p}$, initial tensor $\mathcal{V} \in \mathbb{R}^{n \times s \times p}$ and function f.

1. Choose tolerance $\epsilon > 0$ and the maximum number of iterations I_{max}.
2. Set $\beta_1 = \|\mathcal{V}\|_F$, $\mathcal{V}_1 = \mathcal{V}/\beta_1$, and $\beta_0 = 0$.
3. for $j = 1 : I_{max}$
   (a) $\widetilde{\mathcal{W}}_j = \mathcal{A} * \mathcal{V}_j - \beta_{j-1}\mathcal{W}_{j-1}$;
   (b) $\alpha_j = \|\widetilde{\mathcal{W}}_j\|_F$; $\widetilde{\mathcal{W}}_j = \widetilde{\mathcal{W}}_j/\alpha_j$;
   (c) $B_j = \text{bidiag}([\alpha_1, \ldots, \alpha_j], [\beta_1, \ldots, \beta_{j-1}])$;
   (d) $G_j(g) = e_1^T(\sqrt{B_j^T B_j})^\dagger f(\sqrt{B_j^T B_j})e_1$;
   (e) $\widetilde{V}_{j+1} = \mathcal{A}^T * \mathcal{W}_j - \alpha_j \mathcal{V}_j$;
   (f) $\beta_j = \|\widetilde{\mathcal{V}}_{j+1}\|_F$; $\mathcal{V}_{j+1} = \widetilde{\mathcal{V}}_{j+1}/\beta_j$;
   (g) $\tau_j = [0, \ldots, 0, 0, \alpha_j \beta_j]^T$;
   (h) $\widetilde{w}_\xi = \xi + \tau_j^T (B_j^T B_j - \xi I_j)^{-1} \tau_j$;
   (i) Compute
   $$\mathbb{H}_{j+1,\xi} = \begin{bmatrix} B_j^T B_j & \tau_j \\ \tau^T & \widetilde{w}_\xi. \end{bmatrix}$$
   (j) $R_{j+1,\xi}(g) = \widetilde{e}_1^T(\sqrt{\mathbb{H}_{j+1,\xi}})^\dagger f(\sqrt{\mathbb{H}_{j+1,\xi}})\widetilde{e}_1$;
   (k) if $|G_j(g) - R_{j+1,\xi}(g)|/|G_j(g) + R_{j+1,\xi}(g)| < \epsilon$
   $U_{\text{app}}(g) = \beta_0^2[G_j(g) + R_{j+1,\xi}(g)]/2$; Break;
   endif
4. endfor

Output: Approximation $U_{\text{app}}(g)$ of $\text{trace}_{(1)}(\mathcal{V}^T * g(\mathcal{A}^T * \mathcal{A}) * \mathcal{V})$.

4 Numerical experiments

This section describes some numerical examples that illustrate the effectiveness of the proposed methods. All experiments were carried out using MATLAB R2015a on a computer with an Intel Core i-3 processor and 3.89 GBytes of RAM. The computations were done with about 15 significant decimal digits. We show the performance of the tensor t-global Lanczos method (TTGLM) described by Algorithm 3 as well as of the tensor t-global Golub-Kahan method (TTGGKM) described by Algorithm 5 when applied to compute tensor nuclear norms.

The tensor nuclear norm of the tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times p}$ is given by

$$\|\mathcal{A}\|_\star = \sum_{i=1}^{r} \mathcal{S}_r(i,i,1), \tag{35}$$

where $\mathcal{S}_r$ is defined by the t-CSVD of $\mathcal{A}$ in (6).

Proposition 6 *The tensor nuclear norm can be computed by using tensor functions as follows*

$$\|\mathcal{A}\|_\star = \mathrm{trace}_{(1)}\left(\sqrt{\mathcal{A}^T * \mathcal{A}}\right). \tag{36}$$

Proof By using (10) and the definition of generalized tensor functions (11), we have

$$\sqrt{\mathcal{A}^T * \mathcal{A}} = \mathcal{Q}_r * \sqrt{\mathcal{S}_r^T * \mathcal{S}_r} * \mathcal{Q}_r^T.$$

Application of (3) now gives

$$\begin{aligned}
\mathrm{trace}_{(1)}(\sqrt{\mathcal{A}^T * \mathcal{A}}) &= \frac{1}{p}\mathrm{trace}(\sqrt{\mathcal{A}^T * \mathcal{A}}) \\
&= \frac{1}{p}\mathrm{trace}(\mathcal{Q}_r * \sqrt{\mathcal{S}_r^T * \mathcal{S}_r} * \mathcal{Q}_r^T) \\
&= \frac{1}{p}\mathrm{trace}(\sqrt{\mathcal{S}_r^T * \mathcal{S}_r} * \mathcal{Q}_r^T * \mathcal{Q}_r) \\
&= \frac{1}{p}\mathrm{trace}(\sqrt{\mathcal{S}_r^T * \mathcal{S}_r}) \\
&= \mathrm{trace}_{(1)}(\sqrt{\mathcal{S}_r^T * \mathcal{S}_r}) \\
&= \sum_{i=1}^{r} \mathcal{S}_r(i,i,1) = \|\mathcal{A}\|_\star.
\end{aligned}$$

Using (36), we can express the tensor nuclear norm as

$$\begin{aligned}
\|\mathcal{A}\|_\star &= \sum_{i=1}^{n} [\sqrt{\mathcal{A}^T * \mathcal{A}}]_{ii}^{(1)} \\
&= \sum_{i=1}^{n} (E_i^T * \sqrt{\mathcal{A}^T * \mathcal{A}} * E_i)^{(1)} \\
&= \sum_{i=1}^{n} \mathtt{unfold}(E_i^T)\sqrt{\mathtt{bcirc}(\mathcal{A}^T * \mathcal{A})}\,\mathtt{unfold}(E_i) \\
&= \sum_{i=1}^{n} \widetilde{e}_i^T \sqrt{\mathtt{bcirc}(\mathcal{A}^T * \mathcal{A})}\widetilde{e}_i,
\end{aligned} \tag{37}$$

where $\widetilde{e}_i = \texttt{unfold}(E_i) = [e_i, 0, \ldots, 0]^T \in \mathbb{R}^{np\times 1}$, $E_i = \texttt{fold}(e_i \otimes I_p) \in \mathbb{R}^{n\times 1\times p}$, and e_i denotes the ith canonical basis vector of $\mathbb{R}^n$.

Straightforward evaluation of (37) or (36) requires considerable computational effort when n is large. A popular approach to reduce the computational burden is to use the stochastic trace estimator by Hutchinson [20]. Let Z be a discrete random variable with values $\{-1, 1\}$ with probability $1/2$ to achieve each value. Ubaru et al. [37] applied this stochastic trace estimator to approximate expressions of the form

$$\text{trace}(f(A)) = \sum_{i=1}^{n_A} e_i^T f(A) e_i, \text{ where } A \in \mathbb{R}^{n_A \times n_A}.$$

They considered a sequence of vectors, $\{v_1, v_2, \ldots, v_s\}$, where $v_i \in \mathbb{R}^{n_A}$. The entries of v_i correspond to independent samples of Z. Then

$$\text{trace}(f(A)) \approx \frac{1}{s} \sum_{j=1}^{s} v_i^T f(A) v_i.$$

We compute an approximation of (37) by the same technique and exploit the fact that the vectors $\{\widetilde{e}_1, \widetilde{e}_2, \ldots, \widetilde{e}_n\}$ have zero entries after the first n entries. Define s vectors $z_1, z_2, \ldots z_s$, $z_i \in \mathbb{R}^{np}$, such that the first n entries of each z_i contain n independent samples of Z, and zeros elsewhere, i.e.,

$$z_i = \begin{bmatrix} \mathbf{z}_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad i = 1, 2, \ldots, s,$$

where the subvectors $\mathbf{z}_i \in \mathbb{R}^n$ are generated by n independent samples of Z. Then (37) can be approximated by

$$\sum_{i=1}^{n} \widetilde{e}_i^T \sqrt{\texttt{bcirc}(\mathcal{A}^T * \mathcal{A})} \widetilde{e}_i \approx \frac{1}{s} \sum_{i=1}^{s} z_i^T \sqrt{\texttt{bcirc}(\mathcal{A}^T * \mathcal{A})} z_i.$$

Letting $Z_i = \texttt{fold}(z_i) \in \mathbb{R}^{n\times 1\times p}$, $i = 1, 2, \ldots, s$, we obtain

$$\begin{aligned} \|A\|_\star &\approx \frac{1}{s} \sum_{i=1}^{s} z_i^T \sqrt{\texttt{bcirc}(\mathcal{A}^T * \mathcal{A})} z_i = \frac{1}{s} \sum_{i=1}^{s} (Z_i^T * \sqrt{\mathcal{A}^T * \mathcal{A}} * Z_i)^{(1)} \\ &\approx \frac{1}{s} \text{trace}_{(1)}(\mathcal{V}^T * \sqrt{\mathcal{A}^T * \mathcal{A}} * \mathcal{V}), \end{aligned} \tag{38}$$

where $\mathcal{V} = [Z_1, Z_2, \ldots, Z_s] \in \mathbb{R}^{n\times s\times p}$.

The right-hand side of (38) is a particular case of the generalized tensor function (16) when $f(t) = t^2$, i.e., $g(t) = \sqrt{t}$. Therefore, the proposed methods can be applied to the approximation of the tensor nuclear norm. For both TTGLM and TTGGKM, the fixed node ξ associated with the Gauss-Radau quadrature rules is set to $\xi = 0$. Then we have the bounds

$$\mathcal{R}_{k+1}^{Lan,\xi}(g) \leq \mathcal{I}_s(g) \leq \mathcal{G}_k^{Lan}(g)$$

and

$$\mathcal{R}_{k+1}^{GK,\xi}(g) \leq \mathcal{I}_s(g) \leq \mathcal{G}_k^{GK}(g)$$

Table 3 Example 1: Approximation of the tensor nuclear norm

Tensors	Methods	Time	Iterations	TNN Comp.	Rel. Err.
$\mathcal{A}_1$	TTGLM	2.49	12	$3.8403 \cdot 10^3$	0.0102
$n = 1133$	TTGGKM	3.12	12	$3.8403 \cdot 10^3$	0.0102
$p = 3$, $s = 20$	t-SVD	7.05	$--$	$3.8799 \cdot 10^3$	$--$
$\mathcal{A}_2$	TTGLM	13.26	15	$5.7763 \cdot 10^3$	0.0015
$n = 2642$	TTGGKM	20.32	15	$5.7763 \cdot 10^3$	0.0015
$p = 3$, $s = 20$	t-SVD	65.44	$--$	$5.7847 \cdot 10^3$	$--$
$\mathcal{A}_3$	TTGLM	43.57	11	$9.6701 \cdot 10^3$	0.0023
$n = 5488$	TTGGKM	72.78	11	$9.6701 \cdot 10^3$	0.0023
$p = 3$, $s = 20$	t-SVD	206.35	$--$	$9.6484 \cdot 10^3$	$--$
$\mathcal{A}_4$	TTGLM	190.52	17	$1.0189 \cdot 10^4$	0.0208
$n = 6927$	TTGGKM	222.43	17	$1.0189 \cdot 10^4$	0.0208
$p = 3$, $s = 20$	t-SVD	719.76	$--$	$9.9809 \cdot 10^3$	$--$

for every k, where $\mathcal{R}_{k+1}^{Lan,\xi}(g)$, $\mathcal{G}_k^{Lan}(g)$, $\mathcal{R}_{k+1}^{GK,\xi}(g)$, and $\mathcal{G}_k^{GK}(g)$ are defined by (27), (26), (34), and (33), respectively; see Tables 1 and 2 for details.

Example 1: We consider four real-world networks that can be found in the Suite Sparse Matrix Collection [5]. The tensors in this example have three frontal slices, each of which corresponds to the adjacency matrix for a graph. When the adjacency matrices A_1, A_2, and A_3 do not have the same size, we let $n = \max\{n_1, n_2, n_3\}$, where n_i is the size of A_i and zero-pad the matrices A_i as necessary. We will use four $n \times n \times 3$ tensors $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, and $\mathcal{A}_4$ defined as follows

$$\mathcal{A}_1 = \texttt{fold}\left(\begin{bmatrix}\textbf{Roget}\\ \textbf{delaunay\_n10}\\ \textbf{email}\end{bmatrix}\right), \mathcal{A}_2 = \texttt{fold}\left(\begin{bmatrix}\textbf{CSphd}\\ \textbf{yeast}\\ \textbf{minnesota}\end{bmatrix}\right),$$

$$\mathcal{A}_3 = \texttt{fold}\left(\begin{bmatrix}\textbf{EPA}\\ \textbf{power}\\ \textbf{Erdos972}\end{bmatrix}\right), \qquad \mathcal{A}_4 = \texttt{fold}\left(\begin{bmatrix}\textbf{Erdos992}\\ \textbf{p2p-Gnutella08}\\ \textbf{Erdos02}\end{bmatrix}\right).$$

We apply the TTGLM and TTGGKM methods to obtain approximations of tensor nuclear norms using the right-hand side of (38), and set $\epsilon < 2 \cdot 10^{-2}$ and $I_{max} = 50$ for both Algorithms 3 and 5. The number of sample fibers s is set to be 20. We also compute the exact tensor nuclear norm (35) by using the t-SVD [27]. Table 3 reports the required CPU time (Time) in seconds, the number of iterations (Iterations), the computed approximation of the tensor nuclear norm (TNN Comp.), and the relative error (Rel. Err.) in the computed approximate solutions; we here consider the nuclear norm determined by the t-SVD to be exact. Table 3 also reports the required CPU time (Time) in seconds to find the exact tensor nuclear norm (TNN Comp.) using the t-SVD. As can be seen from this table, the computational cost for the t-SVD is much higher than for the TTGLM and TTGGKM methods. The table shows both the TTGLM and TTGGKM methods to determine approximations of about the same quality for the chosen tensors. However, the TTGGKM method demands more CPU time than the TTGLM method to approximate the tensor nuclear norm.

To illustrate the quality of the computed bounds of the left-hand side of (38) determined by pairs of Gauss and Gauss-Radau quadrature rules associated with the TTGLM and TTGGKM methods, we consider the tensors $\mathcal{A}_2$ and $\mathcal{A}_3$. Figures 1 and 2 show the upper and lower bounds determined by Algorithms 3 and 5 versus

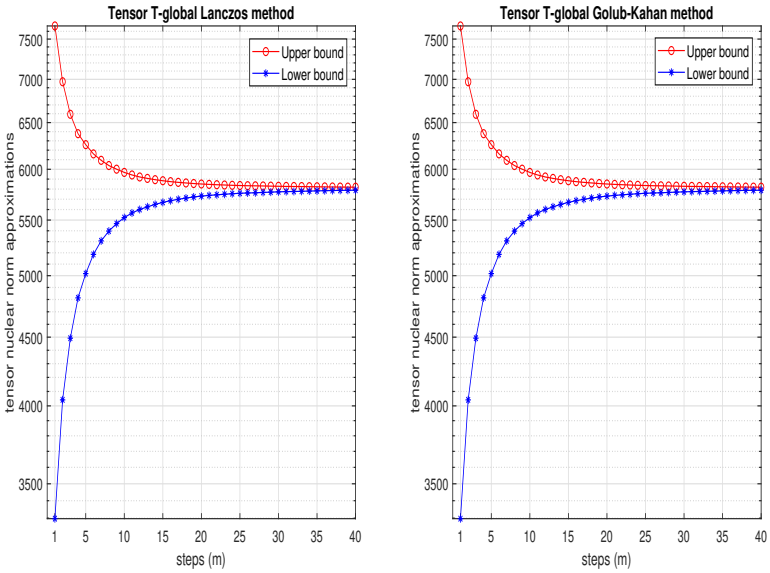

Fig. 1 Upper and lower bounds for $\frac{1}{s}\text{trace}_{(1)}\left(\mathcal{V}^T * \sqrt{\mathcal{A}_2^T * \mathcal{A}_2} * \mathcal{V}\right) \approx \|\mathcal{A}_2\|_\star$.

the number of iterations. The figures illustrate the effectiveness of the quadrature rules to bracket the tensor nuclear norm.

Example 2 (Application to image completion): The need to evaluate a tensor nuclear norm also arises in color image completion. The minimization of the tensor nuclear norm then often is combined with regularization. Image completion enhances the quality of a given incomplete image by solving the optimization problem

$$\arg\min_{\mathcal{A}} \|\mathcal{A}\|_* + \lambda \mathcal{R}(\mathcal{A}), \tag{39}$$

where λ is a regularization parameter and $\lambda\mathcal{R}(\mathcal{A})$ denotes a low-rank regularization term; see [2, 16, 10, 38] for discussions on different regularization procedures. In this example, we are interested in tensor completion using the tensor nuclear norm and the second order total variation technique (TNN-TV_2) discussed and studied in [2]. The low-rank regularization term in this approach is given by

$$R(\mathcal{A}) = \boldsymbol{TV}_2(\mathcal{A}) = \|(TV_2(\mathcal{A}^{(1)}), TV_2(\mathcal{A}^{(2)}), \ldots, TV_2(\mathcal{A}^{(p)})\|_2,$$

where

$$TV_2(\mathcal{A}^{(k)}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \sqrt{(D_2^1\mathcal{A}^{(k)})_{i,j}^2 + (D_2^2\mathcal{A}^{(k)})_{i,j}^2}, \quad k = 1, 2, \ldots, p.$$

The matrices D_2^1 and D_2^2 are defined as follows:

$$D_2^1\mathcal{A}^{(k)} = \mathcal{A}^{(k)}C_n \quad \text{and} \quad D_2^2\mathcal{A}^{(k)} = C_m\mathcal{A}^{(k)},$$

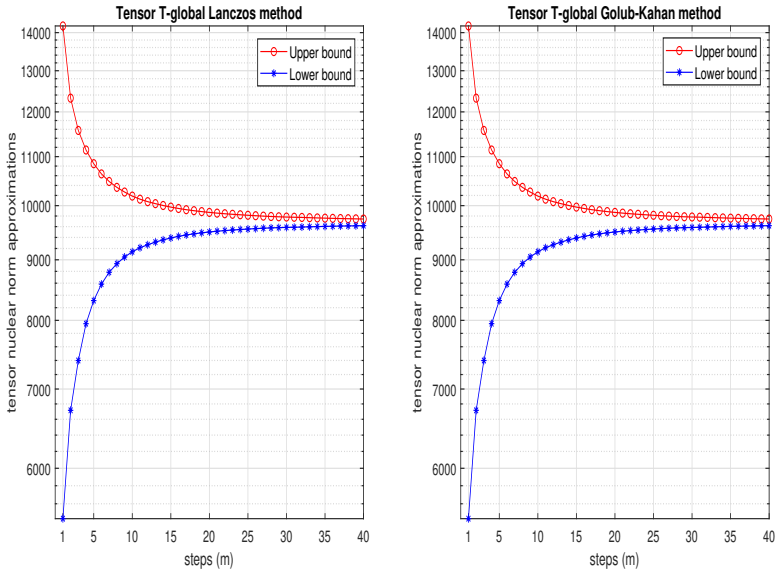

Fig. 2 Upper and lower bounds for $\frac{1}{s}\text{trace}_{(1)}\left(\mathcal{V}^T * \sqrt{\mathcal{A}_3^T * \mathcal{A}_3} * \mathcal{V}\right) \approx \|\mathcal{A}_3\|_\star$.

with

$$C_i = \frac{1}{2}\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 1 & -2 & 1 \\ 1 & \dots & \dots & 0 & 1 & -2 \end{bmatrix} \in \mathbb{R}^{i\times i}, \quad i \in \{m, n\}.$$

Moreover, the authors of [2] apply the Alternating Direction Method of Multipliers (ADMM) to solve the optimization problem (39) based on the TNN-TV$_2$ technique where the approximation of tensor nuclear norms is required. The implementation of TNN-TV$_2$ is described in algorithm [2, Algorithm 5]. The convergence analysis of this algorithm is also detailed in [2].

We consider three color images: Airplane, Brezinski, and fruits. They are represented by tensors of size $512 \times 512 \times 3$. Each frontal slice is a the matrix of that gives the color saturation in red, green, or blue. We illustrate the performance of TTGLM and TTGGKM when applied to approximate the tensor nuclear norm in (39). The recovered images are determined with the TNN-TV_2 algorithm in [2]. Only 10% of the pixels of the available given images are assumed to be known. We would like to determine approximations of the remaining 90% of the pixels. We use the same configuration parameters as in [2]. Figure 3 displays the "original" uncorrupted images, the available "observed" images, and the restored images obtained with the TNN-TV_2 algorithm. The unknown pixel values for each color in the available images are set to zero. The pixel value zero corresponds to black. The TTGLM and TTGGKM methods are applied to obtain approximations of

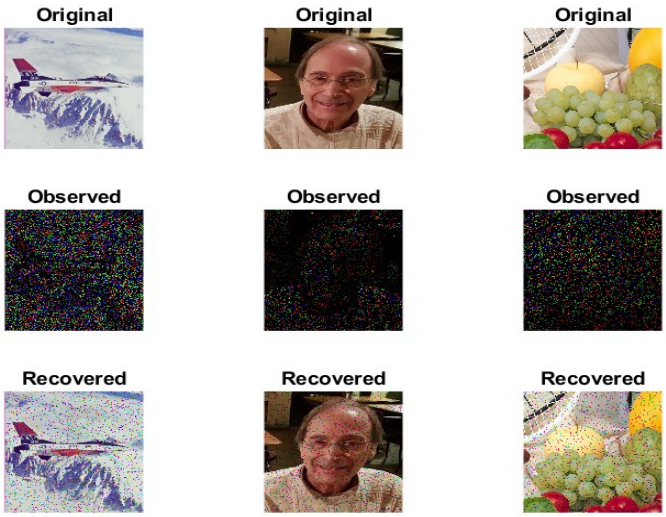

Fig. 3 Original, observed, and restored images.

the tensor nuclear norm using the right-hand sides of (38), where the number of sample fibers s is set to 10 ($\mathcal{V} \in \mathbb{R}^{n\times s\times p}$). We let $\epsilon < 2 \cdot 10^{-2}$ and $I_{max} = 70$ for both Algorithms 3 and 5. Table 4 displays the results obtained with the TTGLM, TTGGKM, and the t-SVD method. The tables shows the proposed methods to require less CPU time than the t-SVD and to require fewer iterations to satisfy the stopping criterion. Figures 4 and 5 show the upper and lower bounds for the approximation $\frac{1}{s}\text{trace}_{(1)}(\mathcal{V}^T * \sqrt{\mathcal{A}^T * \mathcal{A}} * \mathcal{V})$ obtained for the Brezinski image.

5 Conclusion

This paper proposes the tensor global Lanczos method and the tensor global Golub-Kahan bidiagonalization method based on the tensor t-product. Gauss and Gauss-Radau quadrature rules are applied to compute upper and lower bounds for quantities of the form (16) associated with generalized tensor functions. Applications to the computation of the tensor nuclear norm are described. The computed examples illustrate the effectiveness of the proposed methods.

Competing interests

The authors declare that they have no conflict of interest.

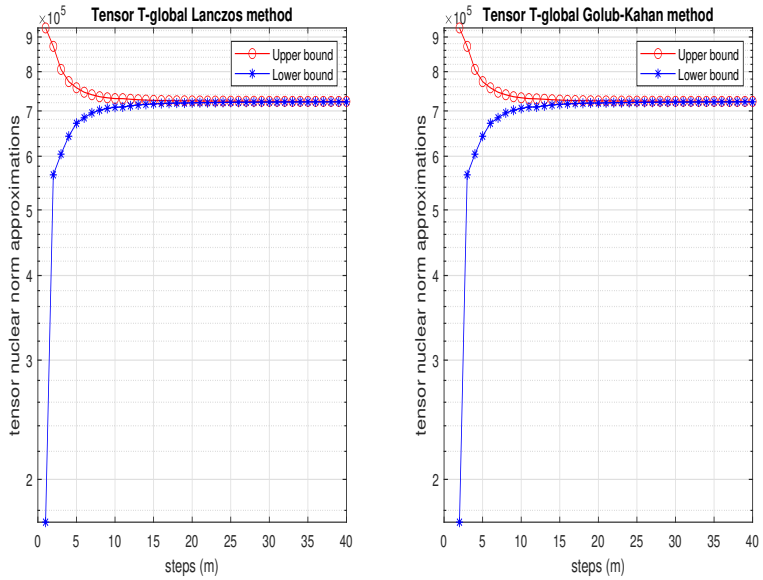

Fig. 4 Upper and lower bounds for $\frac{1}{s}\text{trace}_{(1)}\left(\mathcal{V}^T * \sqrt{\mathcal{A}^T * \mathcal{A}} * \mathcal{V}\right) \approx \|\mathcal{A}\|_\star$ for the observed Brezinski image.

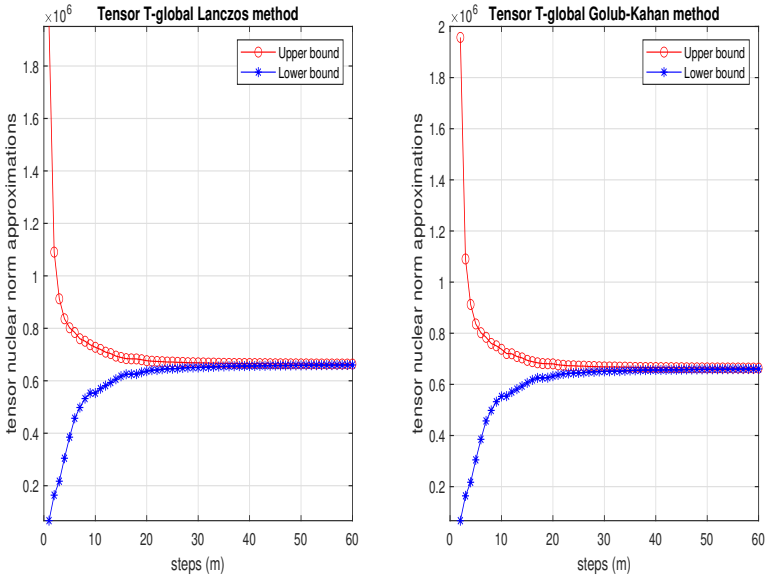

Fig. 5 Upper and lower bounds for $\frac{1}{s}\text{trace}_{(1)}\left(\mathcal{V}^T * \sqrt{\mathcal{A}^T * \mathcal{A}} * \mathcal{V}\right) \approx \|\mathcal{A}\|_\star$ for the recovered Brezinski image.

Table 4 Example 2: Approximation of the tensor nuclear norm

Images	Methods	Time	Iterations	TNN Comp.	Rel. Err.
airplan (Observed)	TTGLM	0.53	6	$1.2371 \cdot 10^6$	0.0067
$m = n = 512$	TTGGKM	0.61	6	$1.2371 \cdot 10^6$	0.0067
$p = 3$, $s = 10$	t-SVD	0.83	$--$	$1.2455 \cdot 10^6$	$--$
airplan (Recovered)	TTGLM	0.82	15	$1.0453 \cdot 10^6$	0.0193
$m = n = 512$	TTGGKM	0.94	15	$1.0453 \cdot 10^6$	0.0193
$p = 3$, $s = 10$	t-SVD	1.02	$--$	$1.0981 \cdot 10^6$	$--$
fruits (Observed)	TTGLM	0.35	7	$1.0973 \cdot 10^6$	0.0064
$m = n = 512$	TTGGKM	0.64	7	$1.0973 \cdot 10^6$	0.0064
$p = 3$, $s = 10$	t-SVD	0.97	$--$	$1.1044 \cdot 10^6$	$--$
fruits (Recovered)	TTGLM	0.55	18	$1.0047 \cdot 10^6$	0.0031
$m = n = 512$	TTGGKM	0.83	18	$1.0047 \cdot 10^6$	0.0031
$p = 3$, $s = 10$	t-SVD	1.01	$--$	$1.0078 \cdot 10^6$	$--$
Brezinski (Observed)	TTGLM	0.46	9	$7.1899 \cdot 10^5$	0.0029
$m = n = 512$	TTGGKM	0.69	9	$7.1899 \cdot 10^5$	0.0029
$p = 3$, $s = 10$	t-SVD	0.86	$--$	$7.1693 \cdot 10^5$	$--$
Brezinski (Recovered)	TTGLM	0.56	24	$6.5854 \cdot 10^5$	0.0319
$m = n = 512$	TTGGKM	0.83	24	$6.5854 \cdot 10^5$	0.0319
$p = 3$, $s = 10$	t-SVD	0.94	$--$	$6.8022 \cdot 10^5$	$--$

Data availability

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

1. Arrigo, F., Benzi, M., Fenu, C.: Computation of generalized matrix functions. SIAM J. Matrix Anal. Appl. 37, 836–860 (2016)
2. Bentbib, A. H., El Hachimi, A., Jbilou, K., Ratnani, A.: A tensor regularized nuclear norm method for image and video completion. J. Optim. Theory Appl. 192, 401–425 (2022)
3. Ben-Israel, A., Greville, T. N. E.: Generalized Inverses: Theory and Applications. 2nd ed., Wiley, New York 2003
4. Cipolla, S., Redivo-Zaglia, M., Tudisco, F.: Shifted and extrapolated power methods for tensor ℓ^p-eigenpairs. Electron. Trans. Numer. Anal. 53, 1–27 (2020)
5. Davis, T., Hu, Y.: The SuiteSparse Matrix Collection, https://sparse.tamu.edu
6. Del Buono, N., Lopez, L., Politi, T.: Computation of functions of Hamiltonian and skew-symmetric matrices. Math. Comp. Simul. 79, 1284–1297 (2008)
7. El Guide, M., El Ichi, A., Jbilou, K., Sadaka, R.: On tensor GMRES and Golub-Kahan methods via the t-product for color image processing. Electron. J. Linear Algebra 37, 524–543 (2021)
8. El Ichi, A., Jbilou, K., Sadaka, R.: On tensor tubal-Krylov subspace methods. Linear Multilinear Algebra 2021. DOI: 10.1080/03081087.2021.1999381
9. Estrada, E., Higham, D. J.: Network properties revealed through matrix functions. SIAM Rev. 52, 696–714 (2010)
10. Fan, Q., Gao, S.: A mixture of nuclear norm and matrix factorization for tensor completion. J. Sci. Comput. 75, 43–64 (2018)
11. Fenu, C., Martin, D., Reichel, L., Rodriguez, G.: Block Gauss and anti-Gauss quadrature with application to networks. SIAM J. Matrix Anal. Appl. 34, 1655–1684 (2013)
12. Gautschi, W.: The interplay between classical analysis and (numerical) linear algebra - a tribute to Gene H. Golub. Electron. Trans. Numer. Anal. 13, 119–147 (2002)
13. Gautschi, W.: Orthogonal Polynomials: Computation and Approximation. Oxford University Press, Oxford (2004)
14. Golub, G. H., Van Loan, C. F.: Matrix Computations, 4th edition. Johns Hopkins University Press, Baltimore (2013)

15. Golub, G. H., Meurant, G.: Matrices, Moments and Quadrature with Applications. Princeton University Press, Princeton (2010)
16. Gu, S. H., Zhang, L., Zuo, W. M.: Weighted nuclear norm minimization with application to image denoising. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Columbus, OH, USA, 2862–2869 (2014)
17. Hawkins, J. B., Ben-Israel, A.: On generalized matrix functions, Linear Multilinear Algebra 1, 163–171 (1973)
18. Higham, N. J.: Functions of Matrices: Theory and Computation. SIAM, Philadelphia (2008)
19. Hochbruck M., Ostermann, A.: Exponential integrators. Acta Numer. 19, 209–286 (2010)
20. Hutchinson, M.: A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. Comm. Statist. Simul. 18, 1059–1076 (1989)
21. Hao, N., Kilmer, M. E., Braman, K., Hoover, R.C.: Facial recognition using tensor-tensor decompositions. SIAM J. Imaging Sci. 6, 437–463 (2013)
22. Hu, W., Tao, D., Zhang, W., Xie, Y., Yang, Y.: The twist tensor nuclear norm for video completion. IEEE Trans. Neural Netw. Learn. Syst. 28, 2961–2973 (2017)
23. Jbilou, K., Messaoudi, A., Sadok, H.: Global FOM and GMRES algorithms for matrix equations. Appl. Numer. Math. 31, 49–63 (1999)
24. Jbilou, K., Sadok, H., Tinzefte, A.: Oblique projection methods for multiple linear systems. Electron. Trans. Numer. Anal. 20, 119–138 (2005)
25. Kilmer, M. E., Braman, K., Hao, N., Hoover, R. C.: Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. SIAM J. Matrix Anal. Appl. 34, 148–172 (2013)
26. Khoromskij, B. N.: Tensor numerical methods for multidimensional PDES: Theoretical analysis and initial applications. ESAIM Proc. Surv. 48, 1–28 (2015)
27. Kilmer, M. E., Martin, C. D.: Factorization strategies for third-order tensors. Linear Algebra Appl. 435, 641–658 (2011)
28. Long, Z., Liu, Y., Chen, L., Zhu, C.: Low rank tensor completion for multiway visual data. Signal Proc. 155, 301–316 (2019)
29. Lund, K.: The tensor t-function: A definition for functions of third-order tensors. Numer. Linear Algebra Appl. 27, Art. e2288 (2020)
30. Madathil, B., George, S. N: Twist tensor total variation regularized-reweighted nuclear norm based tensor completion for video missing area recovery. Inform. Sci. 423, 376–397 (2018)
31. Martin, C.D., Shafer, R., LaRue, B.: An order-p tensor factorization with applications in imaging. SIAM J. Sci. Comput. 35, A474–A490 (2013)
32. Miao, Y., Qi, L., Wei, Y.: Generalized tensor function via the tensor singular value decomposition based on the t-product. Linear Algebra Appl. 590, 258–303 (2020)
33. Miao, Y., Qi, L., Wei, Y.: T-Jordan canonical form and T-Drazin inverse based on the T-product. Commun. Appl. Math. Comput. 3, 201–220 (2021)
34. Reichel, L., Ugwu, U. O.: Tensor Arnoldi-Tikhonov and GMRES-type methods for ill-posed problems with a t-product structure. J. Sci. Comput. 90, Art. 59 (2022)
35. Reichel, L., Ugwu, U. O.: The tensor Golub-Kahan-Tikhonov method applied to the solution of ill-posed problems with a t-product structure. Numer. Linear Algebra Appl. 29, Art. e2412 (2022)
36. Sun, W., Huang, L., So, H. C., Wang, J.: Orthogonal tubal rank-1 tensor pursuit for tensor completion. Signal Proc. 157, 213–224 (2019)
37. Ubaru, S., Chen, J., Saad, Y.: Fast estimation of tr(f(A)) via stochastic Lanczos quadrature. SIAM J. Matrix Anal. Appl. 38, 1075–1099 (2017)
38. Xu, J., Cheng Y., Ma, Y.: Weighted Schatten p-norm low rank error constraint for image denoising. Entropy (Basel) 158, (2021)