Calcolo (2023) 60:35
https://doi.org/10.1007/s10092-023-00527-3

The Fréchet derivative of the tensor t-function

Kathryn Lund[1] · Marcel Schweitzer[2]

Received: 17 February 2023 / Revised: 22 May 2023 / Accepted: 24 May 2023

Abstract

The *tensor t-function*, a formalism that generalizes the well-known concept of matrix functions to third-order tensors, is introduced in Lund (Numer Linear Algebra Appl 27(3):e2288). In this work, we investigate properties of the Fréchet derivative of the tensor t-function and derive algorithms for its efficient numerical computation. Applications in condition number estimation and nuclear norm minimization are explored. Numerical experiments implemented by the `t-Frechet` toolbox hosted at https://gitlab.com/katlund/t-frechet illustrate properties of the t-function Fréchet derivative, as well as the efficiency and accuracy of the proposed algorithms.

Keywords Tensors · Multidimensional arrays · Tensor t-product · Matrix functions · Fréchet derivative · Block circulant matrices

Mathematics Subject Classification 15A69 · 65F60 · 65F35

1 Introduction

Functions of matrices play an important role in many areas of applied mathematics and scientific computing, e.g., in network analysis [11], exponential integrators [16], physical simulations [33] and statistical sampling [19]. This concept was generalized

The Fréchet derivative of the tensor t-function is defined, analyzed, and computed via multiple strategies with different performance profiles.

✉ Kathryn Lund
lund@mpi-magdeburg.mpg.de

Marcel Schweitzer
marcel@uni-wuppertal.de

[1] Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany

[2] School of Mathematics and Natural Sciences, Bergische Universität Wuppertal, 42097 Wuppertal, Germany

Published online: 16 June 2023

to functions of third-order tensors in [22], based on the tensor t-product formalism [5, 24, 25]; see also [32] for a further extension to so-called *generalized tensor functions*, which are functions of tensors with non-square faces. Functions (and generalized functions) of tensors have applications in deblurring of color images [34], tensor neural networks [10, 31], multilinear dynamical systems [17], and the computation of the tensor nuclear norm [4].

For functions of matrices, the Fréchet derivative is a well-established object with applications in, e.g., condition number estimation [1], analysis of complex networks [9, 35], and the solution of matrix optimization problems [37]. In this work, we consider the Fréchet derivative of functions of tensors, in order to generalize the above techniques to the tensor setting.

In addition to condition number estimation, the tensor Fréchet derivative has a number of potential applications, most notably in gradient descent procedures for nuclear norm minimization [3, 6, 18, 26, 29, 30, 38, 39]. Thanks to close connections with bivariate functions (see, e.g., [27] for the matrix function case), computational approaches for the tensor Fréchet derivative are a stepping stone towards solutions of tensor Lyapunov and Sylvester equations [28]. Furthermore, a generalization of the network sensitivity measures discussed in [9, 35] to multilayer networks (which can be represented as tensors) will also require a tensor Fréchet derivative.

This paper is organized as follows. In Sect. 2, we collect several important definitions and results concerning matrix functions, the Fréchet derivative, and the tensor t-product. Section 3 summarizes key results on the tensor t-function and introduces definitions and properties of its Fréchet derivative $L_f(\mathcal{A}, \mathcal{C})$, including explicit Kronecker forms. In Sect. 4 we discuss a number of methods for computing $L_f(\mathcal{A}, \mathcal{C})$, drawing on well understood techniques such as Krylov subspace methods for matrix functions and fast Fourier transforms. We examine applications such as the condition number of t-functions and the gradient of the tensor nuclear norm in Sect. 5. Finally, in Sect. 6 we compare the performance of different algorithms for small- and medium-scale problems, and we summarize our findings in Sect. 7.

2 Foundations

We recall important concepts from matrix function theory, Fréchet derivatives, and the t-product formalism that form the basis of this work.

2.1 Functions of matrices

Functions of matrices can be defined in many different ways, the three most popular of which are based on the Jordan canonical form, Hermite interpolation polynomials, and the Cauchy integral formula; see [15, Sect. 1.2] for a thorough treatment. We recall two of the definitions that are particularly important for our work.

Let $A \in \mathbb{C}^{n \times n}$ be a matrix with spectrum $\operatorname{spec}(A) := \{\lambda_j\}_{j=1,\ldots,N}$, where $N \leq n$ and the λ_j are distinct. Suppose that A has Jordan canonical form,

$$A = XJX^{-1} = X^{-1}\operatorname{diag}(J_{m_1}(\lambda_{j_1}), \ldots, J_{m_p}(\lambda_{j_\ell}))X, \tag{1}$$

where $J_m(\lambda_j)$ is an $m \times m$ Jordan block for an eigenvalue λ_j. Denote by n_j the *index* of λ_j, i.e., the size of the largest Jordan block associated to λ_j. (Note that eigenvalues may be repeated in the sequence $\{\lambda_{j_k}\}_{k=1}^{\ell}$). We then say that a function is *defined on the spectrum of* A if all the values $f^{(k)}(\lambda_j)$ for $k = 0, \ldots, n_j - 1$ and $j = 1, \ldots, N$ exist.

If f is defined on the spectrum of A with Jordan form (1), then we can define $f(A)$ via

$$f(A) := Xf(J)X^{-1},$$

where $f(J) := \operatorname{diag}(f(J_{m_1}(\lambda_{j_1})), \ldots, f(J_{m_p}(\lambda_{j_\ell})))$, and

$$f(J_{m_i}(\lambda_{j_k})) := \begin{bmatrix} f(\lambda_{j_k}) & f'(\lambda_{j_k}) & \frac{f''(\lambda_{j_k})}{2!} & \ldots & \frac{f^{(n_{j_k}-1)}(\lambda_{j_k})}{(n_{j_k}-1)!} \\ 0 & f(\lambda_{j_k}) & f'(\lambda_{j_k}) & \ldots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{f''(\lambda_{j_k})}{2!} \\ \vdots & & \ddots & \ddots & f'(\lambda_{j_k}) \\ 0 & \ldots & \ldots & 0 & f(\lambda_{j_k}) \end{bmatrix} \in \mathbb{C}^{m_i \times m_i}.$$

When A is diagonalizable with $\operatorname{spec}(A) = \{\lambda_j\}_{j=1,\ldots,n}$ (possibly no longer distinct) the Jordan form definition greatly simplifies to

$$f(A) = X\operatorname{diag}(f(\lambda_1), \ldots, f(\lambda_n))X^{-1},$$

where diag is the operator that maps an n-vector to its corresponding $n \times n$ diagonal matrix.

When f is analytic on a region that contains $\operatorname{spec}(A)$, we can alternatively define $f(A)$ via the Cauchy integral formula,

$$f(A) := \frac{1}{2\pi i}\int_{\Gamma} f(\zeta)(\zeta I - A)^{-1}\,\mathrm{d}\zeta,$$

where Γ is a path that winds around $\operatorname{spec}(A)$ exactly once.

When f is analytic, so that both of the above definitions can be applied, the two definitions are equivalent and yield the same result; see [15, Theorem 1.12].

2.2 The Fréchet derivative

In the most general case, the Fréchet derivative is defined for functions between normed vector spaces V, W (with respective norms $\|\cdot\|_V, \|\cdot\|_W$). Let $U \subset V$ be an open subset and let $f : U \longrightarrow W$. Then f is *Fréchet-differentiable* at $\boldsymbol{u} \in U$ if there exists a bounded linear operator $L(\boldsymbol{u}) : V \to W$ such that

$$\lim_{\|\boldsymbol{h}\|_V \to 0} \frac{\|f(\boldsymbol{u}+\boldsymbol{h}) - f(\boldsymbol{u}) - L(\boldsymbol{u})\boldsymbol{h}\|_W}{\|\boldsymbol{h}\|_V} = 0. \tag{2}$$

When $f : \mathbb{C}^{n\times n} \longrightarrow \mathbb{C}^{n\times n}$ is a function of a matrix, one usually denotes the Fréchet derivative of f at the matrix A as $L_f(A,\cdot)$ (see, e.g., [15, Chapter 3]) and rephrases the condition (2) using the matrix two-norm and Landau notation as

$$f(A+E) - f(A) = L_f(A,E) + o(\|E\|), \quad \text{for all } E \in \mathbb{C}^{n\times n}, \tag{3}$$

for an appropriate matrix norm $\|\cdot\|$. A sufficient condition for $L_f(A,\cdot)$ to exist is that f is $2n-1$ times continuously differentiable on a region containing $\operatorname{spec}(A)$ (see [15, Theorem 3.8]). If the Fréchet derivative exists, it is unique.

In particular, the Fréchet derivative of a matrix function is guaranteed to exist if f is analytic on a region containing $\operatorname{spec}(A)$, and in this case $L_f(A,E)$ has the integral representation

$$L_f(A,E) = \frac{1}{2\pi i}\int_\Gamma f(\zeta)(\zeta I - A)^{-1} E (\zeta I - A)^{-1}\, \mathrm{d}\zeta, \tag{4}$$

where Γ is again a path that winds around $\operatorname{spec}(A)$ exactly once; see, e.g., [15, 20]. In addition to being of theoretical interest, the integral representation also forms the basis of efficient computational methods for approximating $L_f(A,E)$, in particular when E is of low rank; see [20, 21, 27], as well as [36] for an extension to higher-order Fréchet derivatives.

Related is the *Gâteaux (or directional) derivative* of f at A, defined as

$$G_f(A,E) = \lim_{t\to 0} \frac{f(A+tE) - f(A)}{t}.$$

If f is Fréchet-differentiable at A, all its directional derivatives exist and we have $G_f(A,E) = L_f(A,E)$ for all $E \in \mathbb{C}^{n\times n}$. The converse is not necessarily true: even when all directional derivatives of f at A exist, f need not be Fréchet-differentiable at A.

2.3 Tensors and the t-product

In the context of this work, a *tensor* is viewed as a multidimensional array, i.e., a generalization of the concept of vectors and matrices to higher dimensions. We restrict ourselves to third-order tensors, i.e., arrays in $\mathbb{C}^{n\times m\times p}$, as the t-product introduced

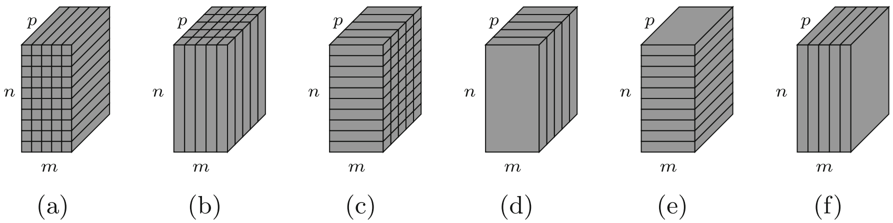

Fig. 1 Different views of a third-order tensor $\mathcal{A} \in \mathbb{C}^{n \times m \times p}$. **a** tube fibers: $\mathcal{A}(:, j, k)$; **b** column fibers: $\mathcal{A}(i, :, k)$; **c** row fibers: $\mathcal{A}(i, j, :)$; **d** frontal slices: $\mathcal{A}(i, :, :)$; **e** lateral slices: $\mathcal{A}(:, j, :)$; **f** horizontal slices: $\mathcal{A}(:, :, k)$

in [5, 24, 25] is only defined in this case. Figure 1 depicts the different "views" of a third-order tensor, which are useful for visualizing the forthcoming concepts. We define the (Frobenius) norm of a tensor $\mathcal{A} \in \mathbb{C}^{n \times m \times p}$, with $\mathcal{A}(i, j, k)$ denoting the ijkth entry, as

$$\|\mathcal{A}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} |\mathcal{A}(i, j, k)|^2}, \tag{5}$$

which can be seen as an analogue of the matrix Frobenius norm $\|\cdot\|_F$.

As the t-product formalism makes extensive use of block matrices, we introduce basic notations for these. Define the standard block unit vectors $\boldsymbol{E}_k^{np \times n} := \boldsymbol{e}_k^p \otimes I_n$, where $\boldsymbol{e}_k^p \in \mathbb{C}^p$ is the kth canonical unit vector in $\mathbb{C}^p$, and I_n is the $n \times n$ identity matrix. When the dimensions are clear from context, we drop the sub- or superscripts.

The tensor t-product [5, 24, 25] defines a way to multiply third-order tensors, based on viewing them as stacks of frontal slices (as in Fig. 1(d)). Let $\mathcal{A} \in \mathbb{C}^{n \times m \times p}$, $\mathcal{B} \in \mathbb{C}^{m \times s \times p}$ and denote their frontal faces, respectively, as $A^{(k)}$ and $B^{(k)}$, $k = 1, \ldots, p$. The operations `unfold` and `fold` transform the tensor $\mathcal{A}$ into a block vector of size $np \times m$ and vice versa, i.e.,

$$\texttt{unfold}(\mathcal{A}) := \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(p)} \end{bmatrix}, \text{ and } \texttt{fold}(\texttt{unfold}(\mathcal{A})) := \mathcal{A}.$$

Additionally, `bcirc` turns $\mathcal{A}$ into a block-circulant matrix of size $np \times mp$,

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

Note that the operators `fold`, `unfold`, and `bcirc` are linear. As a shorthand, we use the term *n-block circulant matrix* for a block circulant matrix with $n \times n$ blocks.

Using the above operators, the *t-product* of the tensors $\mathcal{A}$ and $\mathcal{B}$ is given as

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

Many important concepts well-known for matrices, such as an identity element, inverses, transposition, and eigendecomposition, can also be defined for third-order tensors within the t-product framework; see [5, 24, 25].

Transposition of tensors is defined face-wise, i.e., $\mathcal{A}^H$ is the $m \times n \times p$ tensor obtained by taking the conjugate transpose of each frontal slice of $\mathcal{A}$ and then reversing the order of the second through pth transposed slices. For tensors with $n \times n$ square faces, there is an identity tensor $\mathcal{I}_{n \times n \times p} \in \mathbb{C}^{n \times n \times p}$, whose first frontal slice is the $n \times n$ identity matrix I_n and whose remaining frontal slices are all zero, which fulfills

$$\mathcal{A} * \mathcal{I}_{n \times n \times p} = \mathcal{A} = \mathcal{I}_{n \times n \times p} * \mathcal{A}.$$

We drop the subscript on $\mathcal{I}$ when the dimensions are clear from context.

When $n = m$, a unique inverse tensor $\mathcal{A}^{-1}$ can be defined as expected: if there exists $\mathcal{B} \in \mathbb{C}^{n \times n \times p}$ such that

$$\mathcal{B} * \mathcal{A} = \mathcal{I} = \mathcal{A} * \mathcal{B}, \tag{6}$$

then $\mathcal{A}^{-1} := \mathcal{B}$.

If $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ has diagonalizable faces, i.e., $A^{(k)} = X^{(k)} D^{(k)} \left(X^{(k)}\right)^{-1}$, for all $k = 1, \ldots, p$, a tensor eigendecomposition can be defined via

$$\mathcal{A} = \mathcal{X} * \mathcal{D} * \mathcal{X}^{-1} \text{ and } \mathcal{A} * \texttt{vec}(\mathcal{X})_i = \texttt{vec}(\mathcal{X})_i * \boldsymbol{d}_i, \tag{7}$$

where $\mathcal{X}$ and $\mathcal{D}$ are the tensors whose faces are $X^{(k)}$ and $D^{(k)}$, respectively; $\texttt{vec}(\mathcal{X})_i$ are the $n \times 1 \times p$ lateral slices of $\mathcal{X}$ (see Fig. 1e); and $\boldsymbol{d}_j$ are the $1 \times 1 \times p$ tube fibers of $\mathcal{D}$ (see Fig. 1a).

2.4 Block circulant matrices and the discrete Fourier transform

It is well established that the discrete Fourier transform (DFT) unitarily diagonalizes circulant matrices [8], and in [24, 25] a block version of this result is shown to hold. Namely, letting F_p denote the $p \times p$ DFT and $\otimes$ the Kronecker product, it follows for $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ that

$$(F_p \otimes I_n)\texttt{bcirc}(\mathcal{A})(F_p^H \otimes I_n) = \text{blkdiag}(D_1, \ldots, D_p), \tag{8}$$

where each $D_i, i = 1, \ldots, p$ is an $n \times n$ matrix, and blkdiag works similarly to diag, but instead places matrices on the diagonal.

Another useful tool when working with block circulant matrices is the block circulant shift operator,

$$S_{n,p} := \begin{bmatrix} & & & I_n \\ I_n & & & \\ & \ddots & & \\ & & I_n & \end{bmatrix} \in \mathbb{R}^{np \times np}, \tag{9}$$

which is clearly unitary. Using $S_{n,p}$, define the transformation

$$\mathbb{S}_{n,p} : M \mapsto S_{n,p} M S_{n,p}^T. \tag{10}$$

A matrix $M \in \mathbb{C}^{np \times np}$ is block circulant if and only if $\mathbb{S}_{n,p}(M) = M$. In the following sections, when dimensions and block sizes are clear from the context, we omit the corresponding indices and just write S and $\mathbb{S}$.

3 The tensor t-function

In [22], a definition for functions of third-order tensors based on the t-product is given, generalizing the usual concept of matrix functions discussed in Sect. 2.1. Precisely, the action of the *tensor t-function* f of $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ on another tensor $\mathcal{B} \in \mathbb{C}^{n \times s \times p}$ is defined as

$$f(\mathcal{A}) * \mathcal{B} := \texttt{fold}(f(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})). \tag{11}$$

By taking $\mathcal{B}$ to be the identity tensor, $\mathcal{B} = \mathcal{I}_{n \times n \times p}$, one obtains the t-function $f(\mathcal{A})$ via

$$\begin{aligned} f(\mathcal{A}) &:= \texttt{fold}\big(f(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{I}_{n \times n \times p})\big) \\ &= \texttt{fold}\Big(f(\texttt{bcirc}(\mathcal{A}))\boldsymbol{E}_1^{np \times n}\Big). \end{aligned} \tag{12}$$

Note in particular that when $f(z) = z^{-1}$, we recover the definition of the tensor inverse (6); see [22, Theorem 5(iv)].

The definitions (11) and (12) boil down to evaluating the action of a matrix function (in the usual sense) on a block vector. The t-function therefore inherits many useful properties from matrix functions.

Theorem 1 (Theorem 6 in [22]) *Let $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$, and let $f : \mathbb{C} \to \mathbb{C}$ be defined on a region in the complex plane containing the spectrum of* $\texttt{bcirc}(\mathcal{A})$. *For part (iv), assume that $\mathcal{A}$ has an eigendecomposition as in equation* (7)*, with $\mathcal{A} * \texttt{vec}(\mathcal{X})_i = \mathcal{D} * \texttt{vec}(\mathcal{X})_i = \texttt{vec}(\mathcal{X})_i * \boldsymbol{d}_i$, $i = 1, \ldots, n$. Then it holds that*

(i) *$f(\mathcal{A})$ commutes with $\mathcal{A}$;*
(ii) *$f(\mathcal{A}^H) = f(\mathcal{A})^H$;*

(iii) $f(\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1}) = \mathcal{X} f(\mathcal{A}) \mathcal{X}^{-1}$*; and*
(iv) $f(\mathcal{D}) * \texttt{vec}(\mathcal{X})_i = \texttt{vec}(\mathcal{X})_i * f(\boldsymbol{d}_i)$*, for all* $i = 1, \ldots, n$.

3.1 The derivative of the tensor t-function

In view of (12), which defines the tensor t-function in terms of a matrix function of a block-circulant matrix, it appears natural to define its Fréchet derivative accordingly.

Lemma 1 *Let* $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ *and let* f *be* $2np - 1$ *times continuously differentiable on a region containing* $\mathrm{spec}(\texttt{bcirc}(\mathcal{A}))$. *Then the Fréchet derivative of* f *at* $\mathcal{A}$ *exists, and for any* $\mathcal{C} \in \mathbb{C}^{n \times n \times p}$,

$$L_f(\mathcal{A}, \mathcal{C}) = \texttt{fold}\Big(L_f(\texttt{bcirc}(\mathcal{A}), \texttt{bcirc}(\mathcal{C}))\boldsymbol{E}_1^{np \times n}\Big). \tag{13}$$

Proof The operator $L_f(\texttt{bcirc}(\mathcal{A}), \cdot)$ is the Fréchet derivative of f at a matrix of size $np \times np$, so its existence is guaranteed by [15, Theorem 3.8] under the assumptions of the lemma. Now consider the difference

$$\begin{aligned} f(\mathcal{A} + \mathcal{C}) - f(\mathcal{A}) = \texttt{fold}\Big(f(\texttt{bcirc}(\mathcal{A} + \mathcal{C}))\boldsymbol{E}_1^{np \times n}\Big) \\ - \texttt{fold}\Big(f(\texttt{bcirc}(\mathcal{A}))\boldsymbol{E}_1^{np \times n}\Big) \end{aligned} \tag{14}$$

Using linearity of `bcirc`, `fold`, and matrix multiplication, we can rewrite (14) as

$$\begin{aligned} & f(\mathcal{A} + \mathcal{C}) - f(\mathcal{A}) \\ & \quad = \texttt{fold}\Big(f\left(\texttt{bcirc}(\mathcal{A} + \mathcal{C})\right) \boldsymbol{E}_1^{np \times n} - f\left(\texttt{bcirc}(\mathcal{A})\right) \boldsymbol{E}_1^{np \times n}\Big) \\ & \quad = \texttt{fold}\Big(\left(f\left(\texttt{bcirc}(\mathcal{A} + \mathcal{C})\right) - f\left(\texttt{bcirc}(\mathcal{A})\right)\right) \boldsymbol{E}_1^{np \times n}\Big) \\ & \quad = \texttt{fold}\Big(\left(f\left(\texttt{bcirc}(\mathcal{A}) + \texttt{bcirc}(\mathcal{C})\right) - f\left(\texttt{bcirc}(\mathcal{A})\right)\right) \boldsymbol{E}_1^{np \times n}\Big) \\ & \quad = \texttt{fold}\Big(\Big(L_f\left(\texttt{bcirc}(\mathcal{A}), \texttt{bcirc}(\mathcal{C})\right) + o\left(\|\texttt{bcirc}(\mathcal{C})\|_F\right)\Big) \boldsymbol{E}_1^{np \times n}\Big) \\ & \quad = \texttt{fold}\Big(\Big(L_f\left(\texttt{bcirc}(\mathcal{A}), \texttt{bcirc}(\mathcal{C})\right)\Big) \boldsymbol{E}_1^{np \times n}\Big) + o\left(\|\texttt{bcirc}(\mathcal{C})\|_F\right), \end{aligned} \tag{15}$$

where we have used definition (3) in the second-to-last equality.

Due to the special structure of $\texttt{bcirc}(\mathcal{C})$, each of its $np \times n$ block-columns fulfills

$$\left\| [\texttt{bcirc}(\mathcal{C})]_{:,(i-1)\cdot n:i\cdot n} \right\|_F = \|\mathcal{C}\|_F, i = 1, \ldots, p,$$

so that in total $\|\texttt{bcirc}(\mathcal{C})\|_F = \sqrt{p}\,\|\mathcal{C}\|$. Therefore, $o(\|\texttt{bcirc}(\mathcal{C})\|_F) = o(\|\mathcal{C}\|_F)$ and it follows from (15) that (13) is indeed the Fréchet derivative of $f(\mathcal{A})$ in the sense of definition (2). □

If the assumptions of Lemma 1 are fulfilled, we also say that f is *t-Fréchet differentiable* at $\mathcal{A}$.

A similar relation holds for the Gâteaux derivative.

Proposition 1 *Let f be Gâteaux-differentiable at $\mathtt{bcirc}(\mathcal{A})$. Then f is Gâteaux-differentiable at $\mathcal{A}$, and*

$$G_f(\mathcal{A}, \mathcal{C}) = \mathtt{fold}\Big(G_f(\mathtt{bcirc}(\mathcal{A}), \mathtt{bcirc}(\mathcal{C}))\boldsymbol{E}_1^{np\times n}\Big). \tag{16}$$

Proof The proof follows directly from the definition of the Gâteaux derivative, by inserting the definition (12) of the tensor t-function and again exploiting the linearity of `fold` and `bcirc`. Consequently, we find

$$\begin{aligned}
G_f(\mathcal{A}, \mathcal{C}) &= \lim_{t\to 0} \frac{f(\mathcal{A}+t\mathcal{C}) - f(\mathcal{A})}{t} \\
&= \lim_{t\to 0} \frac{\mathtt{fold}\Big(f(\mathtt{bcirc}(\mathcal{A}+t\mathcal{C}))\boldsymbol{E}_1^{np\times n}\Big) - \mathtt{fold}\Big(f(\mathtt{bcirc}(\mathcal{A}))\boldsymbol{E}_1^{np\times n}\Big)}{t} \\
&= \lim_{t\to 0} \frac{\mathtt{fold}\Big(f(\mathtt{bcirc}(\mathcal{A}+t\mathcal{C}))\boldsymbol{E}_1^{np\times n} - f(\mathtt{bcirc}(\mathcal{A}))\boldsymbol{E}_1^{np\times n}\Big)}{t} \\
&= \lim_{t\to 0} \frac{\mathtt{fold}\Big((f(\mathtt{bcirc}(\mathcal{A}+t\mathcal{C})) - f(\mathtt{bcirc}(\mathcal{A})))\boldsymbol{E}_1^{np\times n}\Big)}{t} \\
&= \lim_{t\to 0} \frac{\mathtt{fold}\Big((f(\mathtt{bcirc}(\mathcal{A})+t\cdot\mathtt{bcirc}(\mathcal{C})) - f(\mathtt{bcirc}(\mathcal{A})))\boldsymbol{E}_1^{np\times n}\Big)}{t} \\
&= \mathtt{fold}\left(\lim_{t\to 0} \frac{(f(\mathtt{bcirc}(\mathcal{A})+t\cdot\mathtt{bcirc}(\mathcal{C})) - f(\mathtt{bcirc}(\mathcal{A})))}{t}\boldsymbol{E}_1^{np\times n}\right) \\
&= \mathtt{fold}\Big(G_f(\mathtt{bcirc}(\mathcal{A}), \mathtt{bcirc}(\mathcal{C}))\boldsymbol{E}_1^{np\times n}\Big),
\end{aligned}$$

which is exactly (16). □

Remark 1 As in the matrix case, when f is Fréchet-differentiable at $\mathcal{A}$, then its Fréchet and Gâteaux derivative coincide:

$$L_f(\mathcal{A}, \mathcal{C}) = G_f(\mathcal{A}, \mathcal{C}).$$

Remark 2 In the derivation of the Gâteaux derivative, one can observe that when $A, C \in \mathbb{C}^{np\times np}$ are both n-block circulant matrices, then $L_f(A, C) = G_f(A, C)$ is also n-block circulant.

3.2 Properties of the t-Fréchet derivative

As it is defined in terms of the Fréchet derivative of a matrix function, the t-Fréchet derivative (13) also inherits many of the properties of the matrix function derivative, which we collect in the following lemma.

Lemma 2 *Let $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ and let g_1 and g_2 be t-Fréchet differentiable at $\mathcal{A}$. Then*

(i) *$f_1 = \alpha g_1 + \beta g_2$ is t-Fréchet differentiable at $\mathcal{A}$, and*

$$L_{f_1}(\mathcal{A}, \mathcal{C}) = \alpha L_{g_1}(\mathcal{A}, \mathcal{C}) + \beta L_{g_2}(\mathcal{A}, \mathcal{C}).$$

(ii) *$f_2 = g_1 g_2$ is t-Fréchet differentiable at $\mathcal{A}$, and*

$$L_{f_2}(\mathcal{A}, \mathcal{C}) = L_{g_1}(\mathcal{A}, \mathcal{C}) g_2(\mathcal{A}) + g_1(\mathcal{A}) L_{g_2}(\mathcal{A}, \mathcal{C}).$$

(iii) *If further h is t-Fréchet differentiable at $h(\mathcal{A})$, then $f_3 = h \circ g_1$ is t-Fréchet differentiable at $\mathcal{A}$, and*

$$L_{f_3}(\mathcal{A}, \mathcal{C}) = L_h(g_1(\mathcal{A}), L_{g_1}(\mathcal{A}, \mathcal{C})).$$

Proof Let A, C denote $\texttt{bcirc}(\mathcal{A})$, $\texttt{bcirc}(\mathcal{C})$, respectively. For part (i), observe that by (13), we have

$$\begin{aligned} L_{f_1}(\mathcal{A}, \mathcal{C}) &= \texttt{fold}\big(L_{f_1}(A, \mathcal{C}) \boldsymbol{E}_1^{np \times n}\big) \\ &= \texttt{fold}\Big(\Big(\alpha L_{g_1}(A, \mathcal{C}) + \beta L_{g_2}(A, \mathcal{C})\Big) \boldsymbol{E}_1^{np \times n}\Big) \\ &= \alpha \cdot \texttt{fold}\big(L_{g_1}(A, \mathcal{C}) \boldsymbol{E}_1^{np \times n}\big) + \beta \cdot \texttt{fold}\big(L_{g_2}(A, \mathcal{C}) \boldsymbol{E}_1^{np \times n}\big) \\ &= \alpha L_{g_1}(\mathcal{A}, \mathcal{C}) + \beta L_{g_2}(\mathcal{A}, \mathcal{C}), \end{aligned}$$

where the second equality follows from [15, Theorem 3.2] and the third equality follows from the linearity of $\texttt{fold}$. In a completely analogous fashion, part (ii) and (iii) follow from their respective matrix function counterparts [15, Theorem 3.3 & Theorem 3.4]. □

We also have an analogous relation to the integral representation (4).

Lemma 3 *Let f be analytic on a region containing* $\operatorname{spec}(\texttt{bcirc}(\mathcal{A}))$. *Then*

$$L_f(\mathcal{A}, \mathcal{C}) = \frac{1}{2\pi i} \int_\Gamma f(\zeta)(\zeta \mathcal{I} - \mathcal{A})^{-1} * \mathcal{C} * (\zeta \mathcal{I} - \mathcal{A})^{-1} \, \mathrm{d}\zeta,$$

where the inverse is defined as in (6).

Proof Let A_ζ, C denote $\texttt{bcirc}(\zeta \mathcal{I} - \mathcal{A})$, $\texttt{bcirc}(\mathcal{C})$, respectively. By (4) applied to $L_f(\mathcal{A}, \mathcal{C})$ and the linearity of $\texttt{fold}$, it follows that

$$L_f(\mathcal{A}, \mathcal{C}) = \frac{1}{2\pi i} \int_\Gamma f(\zeta) \texttt{fold}\Big(A_\zeta^{-1} C A_\zeta^{-1} \boldsymbol{E}_1^{np \times n}\Big) \, \mathrm{d}\zeta. \tag{17}$$

Noting that $A_\zeta^{-1} \boldsymbol{E}_1^{np \times n} = \texttt{unfold}\big((\zeta \mathcal{I} - \mathcal{A})^{-1}\big)$, we have

$$C A_\zeta^{-1} \boldsymbol{E}_1^{np \times n} = \texttt{unfold}\Big(\mathcal{C} * (\zeta \mathcal{I} - \mathcal{A})^{-1}\Big),$$

so that (17) becomes

$$\begin{aligned} L_f(\mathcal{A},\mathcal{C}) &= \frac{1}{2\pi i}\int_\Gamma f(\zeta)\texttt{fold}\Big(A_\zeta^{-1}\texttt{unfold}\Big(\mathcal{C}*(\zeta\mathcal{I}-\mathcal{A})^{-1}\Big)\Big)\,\mathrm{d}\zeta \\ &= \frac{1}{2\pi i}\int_\Gamma f(\zeta)(\zeta\mathcal{I}-\mathcal{A})^{-1}*\mathcal{C}*(\zeta\mathcal{I}-\mathcal{A})^{-1}\,\mathrm{d}\zeta. \end{aligned}$$

□

3.3 Explicit representation of the t-Fréchet derivative

An intuitive way to compute $L_f(\mathcal{A},\mathcal{C})$ for a particular direction tensor $\mathcal{C}$ is based on a well known relation for the matrix Fréchet derivative. For matrices $A, C \in \mathbb{C}^{np\times np}$, if f is $2np-1$ times continuously differentiable on a region containing $\mathrm{spec}(A)$, we have

$$f\left(\begin{bmatrix} A & C \\ O_{np\times np} & A\end{bmatrix}\right) = \begin{bmatrix} f(A) & L_f(A,C) \\ O_{np\times np} & f(A)\end{bmatrix}, \tag{18}$$

where $O_{np\times np}$ denotes an $np\times np$ matrix of zeros; see [15, eq. (3.16)]. Thus, $L_f(A,C)$ can be found by first evaluating f at a $2np\times 2np$ block upper triangular matrix and then extracting the top-right block,

$$L_f(A,C) = \begin{bmatrix} I_{np} & O_{np\times np}\end{bmatrix}\cdot f\left(\begin{bmatrix} A & C \\ O_{np\times np} & A\end{bmatrix}\right)\cdot\begin{bmatrix} O_{np\times np} \\ I_{np}\end{bmatrix}. \tag{19}$$

In the context of the Fréchet derivative of the t-function, (19) turns into

$$L_f(\mathcal{A},\mathcal{C}) = \texttt{fold}\left(\begin{bmatrix} I_{np} & O_{np\times np}\end{bmatrix}\cdot f\left(\begin{bmatrix} A & C \\ O_{np\times np} & A\end{bmatrix}\right)\cdot\begin{bmatrix} O_{np\times n} \\ I_n \\ O_{n(p-1)\times n}\end{bmatrix}\right),$$

where $A = \texttt{bcirc}(\mathcal{A})$, $C = \texttt{bcirc}(\mathcal{C})$, and we have used the fact that

$$\begin{bmatrix} O_{np\times np} \\ I_{np}\end{bmatrix} \boldsymbol{E}_1^{np\times n} = \begin{bmatrix} O_{np\times np} \\ I_{np}\end{bmatrix}\cdot\begin{bmatrix} I_n \\ O_{n(p-1)\times n}\end{bmatrix} = \begin{bmatrix} O_{np\times n} \\ I_n \\ O_{n(p-1)\times n}\end{bmatrix}.$$

We can thus explicitly write the Fréchet derivative of the t-function $f(\mathcal{A})$ in the direction $\mathcal{C}$ in terms of the product of a matrix function acting on a block vector, wherein the upper half of the resulting block vector is extracted and folded back into a tensor. In summary,

$$L_f(\mathcal{A},\mathcal{C}) = \texttt{fold}\left(\left[f\left(\begin{bmatrix} \texttt{bcirc}(\mathcal{A}) & \texttt{bcirc}(\mathcal{C}) \\ O_{np\times np} & \texttt{bcirc}(\mathcal{A})\end{bmatrix}\right)\begin{bmatrix} O_{np\times n} \\ I_n \\ O_{n(p-1)\times n}\end{bmatrix}\right]_{1:np,:}\right). \tag{20}$$

3.4 Kronecker forms of the t-Fréchet derivative

The Fréchet derivative induces a linear mapping $L_f(\mathcal{A}, \cdot) : \mathbb{C}^{n\times n\times p} \longrightarrow \mathbb{C}^{n\times n\times p}$. Thus, identifying $\mathbb{C}^{n\times n\times p}$ with $\mathbb{C}^{n^2 p}$, there is a matrix representation $K_f(\mathcal{A}) \in \mathbb{C}^{n^2 p\times n^2 p}$ such that for any $\mathcal{C} \in \mathbb{C}^{n\times n\times p}$

$$\mathtt{vec}\big(L_f(\mathcal{A}, \mathcal{C})\big) = K_f(\mathcal{A})\mathtt{vec}(\mathcal{C}), \tag{21}$$

where $\mathtt{vec}(\cdot)$ stacks the entries of a tensor into a column vector. The matrix $K_f(\mathcal{A})$ is also called the *Kronecker form* of the Fréchet derivative. (See, e.g, [15, Sect. 3.2] for the matrix function case.)

For computing the Kronecker form, one can simply evaluate the Fréchet derivative $L_f(\mathcal{A}, \cdot)$ on all tensors of the canonical basis $\{\mathcal{E}_{ijk} : i, j = 1, \dots, n, k = 1, \dots, p\}$ of $\mathbb{C}^{n\times n\times p}$ (i.e., $\mathcal{E}_{ijk}$ is a tensor with entry one at position (i, j, k) and all other entries zero). We summarize this discussion in the following definition.

Definition 1 Let f be t-Fréchet differentiable at $\mathcal{A} \in \mathbb{C}^{n\times n\times p}$. The *Kronecker form* of $L_f(\mathcal{A}, \cdot)$ is the matrix $K_f(\mathcal{A}) \in \mathbb{C}^{n^2 p\times n^2 p}$ with columns $\boldsymbol{k}_\ell$, $\ell = 1, \dots, n^2 p$ defined via

$$\boldsymbol{k}_{i+(k-1)n+(j-1)np} = \mathtt{vec}\big(L_f(\mathcal{A}, \mathcal{E}_{ijk})\big). \tag{22}$$

A simple computational procedure for forming the Kronecker form is outlined in Algorithm 1, where we use MATLAB-style colon notation, i.e., $a : b$ means all indices between (and including) a and b.

Algorithm 1 Kronecker form of the t-Fréchet derivative

```
Input: $f, \mathcal{A}$
Output: Kronecker form $K = K_f(\mathcal{A})$
for $i = 1, \dots, n$ do
    for $j = 1, \dots, n$ do
        for $k = 1, \dots, p$ do
            $\mathcal{Y} \leftarrow L_f(\mathcal{A}, \mathcal{E}_{ijk})$
            $K(1 : n^2 p, i + (k-1)n + (j-1)np) \leftarrow \mathtt{vec}(\mathcal{Y})$
        end for
    end for
end for
```

Remark 3 We note that the computational cost of Algorithm 1 is extremely high, making it infeasible even for medium scale problems (a situation that is similar already for matrix functions): computing a single Fréchet derivative $L_f(\mathcal{A}, \mathcal{E}_{ijk})$ using the relation (20) and a dense matrix function algorithm for evaluating f has a cost of $\mathcal{O}(n^3 p^3)$ for most practically relevant functions f. Then, forming $\mathcal{K}_f(\mathcal{A})$ via Algorithm 1 costs $\mathcal{O}(n^5 p^4)$ flops and requires $\mathcal{O}(n^4 p^2)$ storage. Thus, the Kronecker form can typically not be used in actual computations, but it is a useful theoretical tool, e.g., for defining condition numbers; see Sect. 5.1.

The tensor t-function is intimately related to matrix functions of block-circulant matrices. It is therefore interesting to examine the relationship between the Kronecker form $K_f(\mathcal{A})$ of the t-Fréchet derivative and the Kronecker form $K_f(\mathtt{bcirc}(\mathcal{A}))$ of the Fréchet derivative of the matrix function $f(\mathtt{bcirc}(\mathcal{A}))$. Note that $K_f(\mathtt{bcirc}(\mathcal{A})) \in \mathbb{C}^{n^2p^2 \times n^2p^2}$, so that both matrices cannot coincide, but it turns out that they are still highly related. To make the connection precise, we first need the following auxiliary result.

Proposition 2 *Let $\mathcal{E}_{ijk}$ be the unit tensor with a 1 only in position (i, j, k) and zeroes everywhere else. Then, with $E_{IJ} \in \mathbb{C}^{np \times np}$ as the matrix that is zero everywhere except for a 1 at $I = i + (k-1)n$, $J = j$,*[1]

$$\mathtt{bcirc}\big(\mathcal{E}_{ijk}\big) = \sum_{\ell=0}^{p-1} \mathcal{S}^{\ell}(E_{IJ}).$$

Proof The result immediately follows by noting that (I, J) as defined above is one particular nonzero entry of $\mathtt{bcirc}\big(\mathcal{E}_{ijk}\big)$, and, by the definition of $\mathcal{S}$, the sequence of matrices $\mathcal{S}^{\ell}(E_{IJ})$ cyclically moves through all other of its nonzero entries.[2] □

Due to the linearity of the Kronecker product, we thus have that

$$L_f(\mathtt{bcirc}(\mathcal{A}), \mathtt{bcirc}\big(\mathcal{E}_{ijk}\big)) = \sum_{\ell=0}^{p} L_f(\mathtt{bcirc}(\mathcal{A}), \mathcal{S}^{\ell}(E_{IJ})), \tag{23}$$

with E_{IJ} as defined in Proposition 2. The Fréchet derivatives on the right-hand side of (23), when vectorized, correspond to p columns of the Kronecker form $K_f(\mathtt{bcirc}(\mathcal{A}))$. Further, by (13) and (22), the first n^2p entries of the left-hand side of (23) correspond to a column of $K_f(\mathcal{A})$. Thus, each column of $K_f(\mathcal{A})$ equals the sum of (the first n^2p entries) of p columns of $K_f(\mathtt{bcirc}(\mathcal{A}))$, and each column of $K_f(\mathtt{bcirc}(\mathcal{A}))$ appears in *exactly one* of those sums.

The indices of the columns of K_f $(\mathtt{bcirc}(\mathcal{A}))$ that contribute to a particular column of $K_f(\mathcal{A})$ can be obtained by carefully inspecting how the index (I, J) is moved around under the cyclical shifts $\mathcal{S}^{\ell}$.

Lemma 4 *Let $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$, let f be analytic on a region containing the spectrum of $\mathtt{bcirc}(\mathcal{A})$, and let $K_1 := K_f$ $(\mathcal{A})$ and $K_2 := K_f$ $(\mathtt{bcirc}(A))$ denote the Kronecker forms of the Fréchet derivatives of the t-function $f(\mathcal{A})$ and the matrix function $f(\mathtt{bcirc}(\mathcal{A}))$, respectively. Then, for $c := i + (k-1)n + (j-1)np$, we have*

$$K_1(:, c) = \sum_{\alpha=1}^{p} K_2(1 : n^2, c + s_\alpha),$$

[1] In other words, $E_{IJ} = \boldsymbol{e}_1^T \otimes \mathtt{unfold}\big(\mathcal{E}_{ijk}\big)$, $\boldsymbol{e}_1 \in \mathbb{C}^p$ is the matrix that is zero everywhere except its first $np \times n$ block column, which is $\mathtt{unfold}\big(\mathcal{E}_{i,j,k}\big)$.

[2] As $\mathcal{S}^p(E_{IJ}) = E_{IJ}$, one could also start with (I, J) corresponding to any other particular nonzero entry of $\mathtt{bcirc}\big(\mathcal{E}_{ijk}\big)$, not necessarily the one given in the assertion.

where

$$s_\alpha = \begin{cases} 0, & \textit{if } \alpha = 1 \\ s_{\alpha-1} + n^2p - np + 1, & \textit{if } \alpha = p - k + 2 \\ s_{\alpha-1} + n^2p + n, & \textit{otherwise.} \end{cases}$$

Proof The result follows from Proposition 2 by observing how $\mathcal{S}$ acts on a unit matrix E_{IJ}. The application of $\mathcal{S}$ cyclically shifts each block of the matrix one block column to the right and one block row down. Thus, as all blocks are $n \times n$, as long as the single nonzero entry of E_{IJ} is not in the last block row or column, it is moved by exactly n entries to the right and n entries down, corresponding to $n^2p + n$ entries when vectorizing. Due to our choice of E_{IJ} in Proposition 2, its nonzero entry lies in the kth block of the first block column. Therefore, this nonzero entry reaches the last block row after $p - k$ applications of $\mathcal{S}$ and then moves to the first block row with the $p - k + 1$st application. Thus, it moves n positions to the right and $n(p - 1)$ positions *up*. This corresponds to $n^2p - np + 1$ entries after vectorization. □

To verify that Lemma 4 is indeed true and to get a better handle on the rather unintuitive indexing scheme, the reader is encouraged to run and examine the script `test_t_func_cond.m` in the `t-frechet` code repository described in Sect. 6.

A further interesting observation is obtained by viewing the relations we have derived so far “in the opposite direction." It then turns out that it is sufficient to compute n^2 Fréchet derivatives in order to obtain all columns of the $n^2p^2 \times n^2p^2$ matrix $K_f(\texttt{bcirc}(\mathcal{A}))$ (and thus, in light of Lemma 4, all columns of $K_f(\mathcal{A})$ as well). This is due to the following result.

Proposition 3 *Let* $\mathcal{A} \in \mathbb{C}^{n\times n\times p}$ *and let* f *be analytic on a region containing* $\operatorname{spec}(\texttt{bcirc}(\mathcal{A}))$. *Further, let* S *denote the shift matrix defined in* (9) *and let* $E_{IJ} \in \mathbb{C}^{n^2p^2\times n^2p^2}$ *be a matrix with* 1 *only in position* (I, J) *and* 0 *everywhere else. Then, for any integers* $\ell_1, \ell_2 \geq 0$,

$$L_f(\texttt{bcirc}(\mathcal{A}), S^{\ell_1}E_{IJ}(S^T)^{\ell_2}) = S^{\ell_1}\left(L_f(\texttt{bcirc}(\mathcal{A}), E_{IJ})\right)(S^T)^{\ell_2}.$$

Proof By [15, Eq. (3.24)], for any $C \in \mathbb{C}^{np\times np}$ we have the relation

$$L_f(\texttt{bcirc}(\mathcal{A}), C) = \sum_{\alpha=1}^{\infty} a_\alpha \sum_{\beta=1}^{\alpha} \texttt{bcirc}(\mathcal{A})^{\beta-1}C\texttt{bcirc}(\mathcal{A})^{\alpha-\beta}, \tag{24}$$

using the power series representation $f(z) = \sum_{\alpha=0}^{\infty} a_\alpha z^\alpha$. Inserting $S^{\ell_1}E_{IJ}(S^T)^{\ell_2}$ instead of C in relation (24), we find that

$$\begin{aligned} &L_f(\texttt{bcirc}(\mathcal{A}), S^{\ell_1}E_{IJ}(S^T)^{\ell_2})) \\ &\quad= \sum_{\alpha=1}^{\infty} a_\alpha \sum_{\beta=1}^{\alpha} \texttt{bcirc}(\mathcal{A})^{\beta-1}S^{\ell_1}E_{IJ}(S^T)^{\ell_2}\texttt{bcirc}(\mathcal{A})^{\alpha-\beta} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{\alpha=1}^{\infty} a_\alpha \sum_{\beta=1}^{\alpha} S^{\ell_1} \mathtt{bcirc}(\mathcal{A})^{\beta-1} (S^T)^{\ell_1} S^{\ell_1} E_{IJ} (S^T)^{\ell_2} S^{\ell_2} \mathtt{bcirc}(\mathcal{A})^{\alpha-\beta} (S^T)^{\ell_2} \\
&= S^{\ell_1} \left(\sum_{\alpha=1}^{\infty} a_\alpha \sum_{\beta=1}^{\alpha} \mathtt{bcirc}(\mathcal{A})^{\beta-1} E_{IJ} \mathtt{bcirc}(\mathcal{A})^{\alpha-\beta} \right) (S^T)^{\ell_2} \\
&= S^{\ell_1} L_f(\mathtt{bcirc}(\mathcal{A}), E_{IJ}) (S^T)^{\ell_2},
\end{aligned}
$$

where for the second equality we have used the fact that powers of block circulant matrices are block circulant (and thus invariant under $\mathcal{S}$), and the third equality follows from the fact that S is unitary. □

As a special case, by choosing $\ell_1 = \ell_2$, Proposition 3 states that the shift operator $\mathcal{S}$ defined in (10) can be "pulled out" of the Fréchet derivative,

$$
L_f(\mathtt{bcirc}(\mathcal{A}), \mathcal{S}^{\ell}(E_{IJ})) = \mathcal{S}^{\ell} \left(L_f(\mathtt{bcirc}(\mathcal{A}), E_{IJ}) \right).
$$

In particular, choosing $\ell_1 = 0$ or $\ell_2 = 0$ (and denoting the other one simply by ℓ), Proposition 3 reveals that all Fréchet derivatives $L_f(\mathtt{bcirc}(\mathcal{A}), S^{\ell} E_{IJ})$ and $L_f(\mathtt{bcirc}(\mathcal{A}), E_{IJ}(S^T)^{\ell})$ have exactly the same entries for any $\ell = 0, \ldots, p-1$, just shifted. It thus suffices to compute one of these Fréchet derivatives and then obtain the others essentially for free by applying S and/or S^T. In total, it is enough to compute $L_f(\mathtt{bcirc}(\mathcal{A}), E_{IJ})$ for $I, J = 1, \ldots, n$, as all other canonical basis matrices E_{IJ} can be generated by appropriate shifts.

Remark 4 For "tubal vectors" $\mathcal{A} \in \mathbb{C}^{1\times 1\times p}$, as they appear in certain tensor neural networks [10, 31], the preceding discussion implies that *all* columns of $K_f(\mathcal{A}) \in \mathbb{C}^{p\times p}$ are shifted copies of the same vector. Thus, in this case, $K_f(\mathcal{A})$ is a circulant matrix.

4 Computing the t-Fréchet derivative

The primary challenge in computing with tensors is the so-called "curse of dimensionality," to which the t-product formalism is not immune. At the same time, due to the equivalence with functions of block circulant matrices, the tools at our disposal are largely limited by what has been developed for matrix functions in general. We discuss viable approaches, along with potential tricks for reducing the overall complexity of computing the t-Fréchet derivative.

4.1 A basic block Krylov subspace method

We recall from (17) in the proof of Lemma 3 that

$$
L_f(\mathcal{A}, \mathcal{C}) = \mathtt{fold}\left(\frac{1}{2\pi i} \int_{\Gamma} f(\zeta) A_\zeta^{-1} C A_\zeta^{-1} \, \mathrm{d}\zeta \cdot \boldsymbol{E}_1^{np\times n} \right), \tag{25}
$$

where $A_\zeta := \mathtt{bcirc}(\zeta\mathcal{I} - \mathcal{A})$ and $C := \mathtt{bcirc}(\mathcal{C})$. The integral term appearing in (25) can be approximated by a block Krylov algorithm when the direction term C is of low rank and can thus be written in the form $C = \boldsymbol{C}_1\boldsymbol{C}_2^H$ with $\boldsymbol{C}_1, \boldsymbol{C}_2 \in \mathbb{C}^{np\times r}$, $r \ll np$.

Remark 5 As an illustration, let us focus on the special case that $\mathcal{C}$ is a rank-one tensor in the sense of the CP tensor format, i.e., that each entry fulfills

$$\mathcal{C}(i,j,k) = \boldsymbol{u}(i)\cdot\boldsymbol{v}(j)\cdot\boldsymbol{w}(k), \qquad \boldsymbol{u},\boldsymbol{v}\in\mathbb{C}^n, \boldsymbol{w}\in\mathbb{C}^p.$$

In this case, the kth frontal face of $\mathcal{C}$ is of the form $C^{(k)} = \boldsymbol{w}(k)\boldsymbol{u}\boldsymbol{v}^T$ and thus

$$\mathtt{bcirc}(\mathcal{C}) := \begin{bmatrix} \boldsymbol{w}(1)\boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{w}(p)\boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{w}(p-1)\boldsymbol{u}\boldsymbol{v}^T & \cdots & \boldsymbol{w}(2)\boldsymbol{u}\boldsymbol{v}^T \\ \boldsymbol{w}(2)\boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{w}(1)\boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{w}(p)\boldsymbol{u}\boldsymbol{v}^T & \cdots & \boldsymbol{w}(3)\boldsymbol{u}\boldsymbol{v}^T \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{w}(p)\boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{w}(p-1)\boldsymbol{u}\boldsymbol{v}^T & \cdots & \boldsymbol{w}(2)\boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{w}(1)\boldsymbol{u}\boldsymbol{v}^T \end{bmatrix}. \tag{26}$$

The matrix (26) has rank at most p,[3] and the low rank factors can be given explicitly in terms of $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$.

Of particular interest is the case in which all three vectors $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$ are canonical unit vectors, which arises, e.g., when measuring the sensitivity of $f(\mathcal{A})$ with respect to changes in one specific entry of $\mathcal{A}$ [9, 35]. Also interesting is when just two of the three vectors are unit vectors, which would occur when measuring the sensitivity with respect to changes in the same entry across all frontal, horizontal, or lateral slices of $\mathcal{A}$.

We define a block Krylov subspace as the block span

$$\mathcal{K}_d(A, \boldsymbol{C}) := \operatorname{span}^{\square}\{\boldsymbol{C}, A\boldsymbol{C}, \ldots, A^{d-1}\boldsymbol{C}\} \subset \mathbb{C}^{np\times r},$$

where d is a small positive integer denoting the iteration index. For more details on the theory and implementation of block Krylov subspaces, see, e.g., [12, 14].

The Krylov subspace algorithm from [21, 27] for approximating

$$\frac{1}{2\pi i}\int_\Gamma f(\zeta) A_\zeta^{-1}\boldsymbol{C}_1\boldsymbol{C}_2^H A_\zeta^{-1}\,\mathrm{d}\zeta \tag{27}$$

now proceeds by building orthonormal bases $\boldsymbol{\mathcal{V}}_d, \boldsymbol{\mathcal{W}}_d \in \mathbb{C}^{np\times dr}$ of the two block Krylov subspaces $\mathcal{K}_d(A, \boldsymbol{C}_1)$ and $\mathcal{K}_d(A^H, \boldsymbol{C}_2)$, with $A := \mathtt{bcirc}(\mathcal{A})$, yielding the following block Arnoldi decompositions:

[3] Letting W denote the circulant matrix of $\boldsymbol{w}$, we have $\mathtt{bcirc}(\mathcal{C}) = W \otimes \boldsymbol{u}\boldsymbol{v}^T$. As $\operatorname{rank}(W \otimes \boldsymbol{u}\boldsymbol{v}^T = \operatorname{rank}(W)\operatorname{rank}(\boldsymbol{u}\boldsymbol{v}^T)$ and clearly $\operatorname{rank}(W) \le p$ and $\operatorname{rank}(\boldsymbol{u}\boldsymbol{v}^T) \le 1$, the assertion holds.

$$A\boldsymbol{\mathcal{V}}_d = \boldsymbol{\mathcal{V}}_d \mathcal{G}_d + G_{d+1,d} \boldsymbol{V}_{d+1} \boldsymbol{E}_{d+1}^H$$
$$A^H \boldsymbol{\mathcal{W}}_d = \boldsymbol{\mathcal{W}}_d \mathcal{H}_d + H_{d+1,d} \boldsymbol{W}_{d+1} \boldsymbol{E}_{d+1}^H.$$

Both $\mathcal{G}_d = \boldsymbol{\mathcal{V}}_d^H A \boldsymbol{\mathcal{V}}_d$ and $H_d = \boldsymbol{\mathcal{W}}_d^H A^H \boldsymbol{\mathcal{W}}_d$ are $dr \times dr$ block upper Hessenberg matrices. An approximation $\widetilde{L}_d$ of (27) is then extracted from the tensorized Krylov subspace $\mathcal{K}_d(A^H, \boldsymbol{C}_2) \otimes \mathcal{K}_d(A, \boldsymbol{C}_1)$ via

$$\widetilde{L}_d := \boldsymbol{\mathcal{V}}_d X_d \boldsymbol{\mathcal{W}}_d^H,$$

where X_d is the $dr \times dr$ upper right block of

$$f\left(\begin{bmatrix} \mathcal{G}_d & (\boldsymbol{\mathcal{V}}_d^H \boldsymbol{C}_1)(\boldsymbol{\mathcal{W}}_d^H \boldsymbol{C}_2)^H \\ & \mathcal{H}_d^H \end{bmatrix}\right).$$

In light of (25), the final approximation for the Fréchet derivative is then given by

$$L_f(\mathcal{A}, \mathcal{C}) \approx \widetilde{\mathcal{L}}_d := \texttt{fold}\Big(\widetilde{L}_d \cdot \boldsymbol{E}_1^{np \times n}\Big).$$

4.2 Using the DFT to improve parallelism

Consider again (20), specifically the argument of f. Thanks to (8) and Theorem 1(iii), we can write

$$f\left(\begin{bmatrix} \texttt{bcirc}(\mathcal{A}) & \texttt{bcirc}(\mathcal{C}) \\ & \texttt{bcirc}(\mathcal{A}) \end{bmatrix}\right) = \mathcal{F}^H f\left(\begin{bmatrix} \mathcal{D}^A & \mathcal{D}^C \\ & \mathcal{D}^A \end{bmatrix}\right) \mathcal{F} \tag{28}$$

with $\mathcal{D}^A = \text{blkdiag}(D_1^A, \ldots, D_p^A)$, $\mathcal{D}^C = \text{blkdiag}(D_1^C, \ldots, D_p^C)$, and

$$\mathcal{F} = \begin{bmatrix} F_p \otimes I_n & \\ & F_p \otimes I_n \end{bmatrix}.$$

Using (18), we can rewrite (28) as

$$f\left(\begin{bmatrix} \texttt{bcirc}(\mathcal{A}) & \texttt{bcirc}(\mathcal{C}) \\ & \texttt{bcirc}(\mathcal{A}) \end{bmatrix}\right) = \mathcal{F}^H \begin{bmatrix} f(\mathcal{D}^A) & L_f(\mathcal{D}^A, \mathcal{D}^C) \\ & f(\mathcal{D}^A) \end{bmatrix} \mathcal{F}. \tag{29}$$

The following theorem, which can be seen as a Daleckiĭ-Kreĭn-type result for block diagonal matrices, will be helpful.

Theorem 2 *Let $A, C \in \mathbb{C}^{np \times np}$ be block diagonal matrices with $n \times n$ blocks, $A =$* blkdiag$(A_1, \ldots, A_p)$*, $C =$* blkdiag$(C_1, \ldots, C_p)$ *and let f be analytic on a region containing* spec(A).

Then $L_f(A, C) = \mathrm{blkdiag}(L_1, \ldots, L_p)$ *with*

$$L_i = L_f(A_i, E_i), \quad i = 1, \ldots, p. \tag{30}$$

Proof When A and C are block diagonal, then for any $k \geq 1$, we have

$$\begin{bmatrix} A & C \\ & A \end{bmatrix}^k = \begin{bmatrix} A^k & M^{(k)} \\ & A^k \end{bmatrix} \tag{31}$$

where $M^{(k)} = \mathrm{blkdiag}(M_1^{(k)}, \ldots, M_p^{(k)})$ with

$$M_i^{(k)} = \sum_{j=1}^{k} A_i^{j-1} C_i A_i^{k-j}, \qquad i = 1, \ldots, p. \tag{32}$$

Let

$$f(z) = \sum_{k=0}^{\infty} a_k z^k$$

be the power series representation of the analytic function f. Then, by (31)–(32), we have

$$f\left(\begin{bmatrix} A & C \\ & A \end{bmatrix}\right) = \begin{bmatrix} f(A) & L \\ & f(A) \end{bmatrix}, \tag{33}$$

where $L = \mathrm{blkdiag}(L_1, \ldots, L_p)$ and

$$L_i = \sum_{k=1}^{\infty} a_k M_i^{(k)} = \sum_{k=1}^{\infty} a_k \sum_{j=1}^{k} A_i^{j-1} C_i A_i^{k-j}. \tag{34}$$

By [15, Eq. (3.24)], the right-hand side of (34) coincides with $L_f(A_i, C_i)$ and by (18), the matrix L in (33) equals $L_f(A, C)$, thus completing the proof. □

Corollary 1 *Let* $\mathcal{A}, \mathcal{C} \in \mathbb{C}^{n \times n \times p}$ *and let* f *be* $2np - 1$ *times continuously differentiable on a region containing* $\mathrm{spec}(\mathtt{bcirc}(\mathcal{A}))$. *Further, let*

$$(F_p \otimes I_n)\mathtt{bcirc}(\mathcal{A})(F_p^H \otimes I_n) = \mathcal{D}^A$$

and

$$(F_p \otimes I_n)\mathtt{bcirc}(\mathcal{C})(F_p^H \otimes I_n) = \mathcal{D}^C$$

with $\mathcal{D}^A = \mathrm{blkdiag}(D_1^A, \ldots, D_p^A)$, $\mathcal{D}^C = \mathrm{blkdiag}(D_1^C, \ldots, D_p^C)$. *Then*

$$L_f(\mathcal{A}, \mathcal{C}) = \mathtt{fold}\left((F_p^H \otimes I_n) \begin{bmatrix} \frac{1}{\sqrt{p}} L_1 \\ \vdots \\ \frac{1}{\sqrt{p}} L_p \end{bmatrix} \right), \tag{35}$$

where the diagonal blocks L_i, $i = 1, \ldots, p$ *are given by*

$$L_i = L_f(D_i^A, D_i^C), \qquad i = 1, \ldots, p. \tag{36}$$

Proof Under the assumptions of the theorem, the existence of the Fréchet derivative is guaranteed by Lemma 1. By combining (20) with (29), we have

$$L_f(\mathcal{A}, \mathcal{C}) = \mathtt{fold}\left(\left[\mathcal{F}^H \begin{bmatrix} f(\mathcal{D}^A) & L_f(\mathcal{D}^A, \mathcal{D}^C) \\ & f(\mathcal{D}^A) \end{bmatrix} \mathcal{F} \cdot \begin{bmatrix} O_{np \times n} \\ I_n \\ O_{n(p-1) \times n} \end{bmatrix} \right]_{1:np,:} \right). \tag{37}$$

According to Theorem 2, we have $L_f(\mathcal{D}^A, \mathcal{D}^C) = \mathrm{blkdiag}(L_1, \ldots, L_p)$ where the diagonal blocks are given by

$$L_i = L_f(D_i^A, D_i^C), \qquad i = 1, \ldots, p.$$

Further, by the definition of $\mathcal{F}$, it holds that

$$\mathcal{F} \cdot \begin{bmatrix} O_{np \times n} \\ I_n \\ O_{n(p-1) \times n} \end{bmatrix} = \begin{bmatrix} F_p \otimes I_n & \\ & F_p \otimes I_n \end{bmatrix} \cdot \begin{bmatrix} O_{np \times n} \\ \boldsymbol{e}_1^p \otimes I_n \end{bmatrix} = \begin{bmatrix} O_{np \times n} \\ F_p \boldsymbol{e}_1^p \otimes I_n . \end{bmatrix}.$$

We therefore have

$$\begin{aligned} &\mathcal{F}^H \begin{bmatrix} f(\mathcal{D}^A) & L_f(\mathcal{D}^A, \mathcal{D}^C) \\ & f(\mathcal{D}^A) \end{bmatrix} \mathcal{F} \cdot \begin{bmatrix} O_{np \times n} \\ I_n \\ O_{n(p-1) \times n} \end{bmatrix} \\ &= \mathcal{F}^H \begin{bmatrix} L_f(\mathcal{D}^A, \mathcal{D}^C) \cdot (F_p \boldsymbol{e}_1^p \otimes I_n) \\ f(\mathcal{D}^A) \cdot (F_p \boldsymbol{e}_1^p \otimes I_n) \end{bmatrix} \\ &= \begin{bmatrix} (F_p^H \otimes I_n) \cdot L_f(\mathcal{D}^A, \mathcal{D}^C) \cdot (F_p \boldsymbol{e}_1^p \otimes I_n) \\ (F_p^H \otimes I_n) \cdot f(\mathcal{D}^A) \cdot (F_p \boldsymbol{e}_1^p \otimes I_n) \end{bmatrix}. \end{aligned} \tag{38}$$

We now focus on the upper half of (38), as only this block is needed for evaluating (37). Due to the structure of $L_f(\mathcal{D}^A, \mathcal{D}^C)$, we have

$$L_f(\mathcal{D}^A, \mathcal{D}^C) \cdot (F_p \boldsymbol{e}_1^p \otimes I_n) = \mathrm{blkdiag}(L_1, \dots, L_p) \cdot (F_p \boldsymbol{e}_1^p \otimes I_n)$$
$$= \begin{bmatrix} \frac{1}{\sqrt{p}} L_1 \\ \vdots \\ \frac{1}{\sqrt{p}} L_p \end{bmatrix}, \qquad (39)$$

where we have used that the DFT matrix fulfills $F_p \boldsymbol{e}_1^p = \frac{1}{\sqrt{p}} \mathbf{1}$. Inserting (38) and (39) into (37) completes the proof. □

Corollary 1 shows that by applying a DFT, the computation of the t-Fréchet derivative can be decoupled into the evaluation of p Fréchet derivatives of $n \times n$ matrices that are completely independent of one another, thus giving rise to an embarrassingly parallel method. However, as the matrices D_i^A, D_i^C occurring in (36) are in general dense and unstructured, computing these Fréchet derivatives is only feasible for moderate values of n (but possibly large p).

5 Applications of the t-Fréchet derivative

In this section, we briefly discuss two applications of the t-Fréchet formalism, namely condition number estimation for tensor functions and the gradient of the tensor nuclear norm.

5.1 The condition number of the t-function

In practical applications, one often works with noisy or uncertain data, and additionally any computation in floating point arithmetic introduces rounding errors. Therefore, when working with the tensor t-function in practice, it is very important to understand how sensitive it is to perturbations in the data. This is measured by *condition numbers*.

The (absolute) condition number of the t-function can be defined by simply extending the well-known concept of condition number of scalar and matrix functions (see, e.g., [15, Chapter 3]), yielding

$$\mathrm{cond}_{\mathrm{abs}}(f, \mathcal{A}) := \lim_{\varepsilon \to 0} \sup_{\|\mathcal{C}\| \leq \varepsilon} \frac{\|f(\mathcal{A} + \mathcal{C}) - f(\mathcal{A})\|}{\varepsilon},$$

where for our setting, $\|\cdot\|$ denotes the norm (5), but can in principle also be any other tensor norm. A relative condition number can be readily defined as

$$\mathrm{cond}_{\mathrm{rel}}(f, \mathcal{A}) := \lim_{\varepsilon \to 0} \sup_{\|\mathcal{C}\| \leq \varepsilon \|f(\mathcal{A})\|} \frac{\|f(\mathcal{A} + \mathcal{C}) - f(\mathcal{A})\|}{\varepsilon \, \|f(\mathcal{A})\|} = \mathrm{cond}_{\mathrm{abs}}(f, \mathcal{A}) \frac{\|\mathcal{A}\|}{\|f(\mathcal{A})\|}.$$

Completely analogously to the matrix function case, the condition number of the t-function can be related to the norm of its Fréchet derivative.

Lemma 5 *Let f and $\mathcal{A}$ be such that $L_f(\mathcal{A},\cdot)$ exists and denote*

$$\left\|L_f(\mathcal{A})\right\| := \max_{\mathcal{C}\neq 0} \frac{\left\|L_f(\mathcal{A},\mathcal{C})\right\|}{\|\mathcal{C}\|}. \tag{40}$$

Then the absolute and relative condition number of $f(\mathcal{A})$ are given by

$$\begin{aligned}\mathrm{cond}_{\mathrm{abs}}(f,\mathcal{A}) &= \left\|L_f(\mathcal{A})\right\|,\\ \mathrm{cond}_{\mathrm{rel}}(f,\mathcal{A}) &= \frac{\left\|L_f(\mathcal{A})\right\|\,\|\mathcal{A}\|}{\|f(\mathcal{A})\|}.\end{aligned}$$

Proof The proof follows by using exactly the same line of argument as in the proof of [15, Theorem 3.1] for the matrix function case, which only requires linearity of the Fréchet derivative and working in a finite-dimensional space and thus holds verbatim in our setting. □

Lemma 5 relates the condition number of the t-Fréchet derivative to the tensor-operator norm $\left\|L_f(\mathcal{A})\right\|$, the computation of which might not be immediately clear (as the quantities on the right-hand side of (40) are third-order tensors). The next result relates it to the spectral norm of the Kronecker form $\left\|K_f(\mathcal{A})\right\|$.

Lemma 6 *Let f and $\mathcal{A}$ be such that $L_f(\mathcal{A},\cdot)$ exists and denote by $K_f(\mathcal{A})$ the Kronecker form of the Fréchet derivative, as defined in (21). Then*

$$\left\|L_f(\mathcal{A})\right\| = \left\|K_f(\mathcal{A})\right\|_2. \tag{41}$$

Proof By the definition of the tensor norm (5), it is clear that $\|\mathcal{B}\| = \|\mathtt{vec}(\mathcal{B})\|_2$ for any tensor $\mathcal{B}$. Thus

$$\left\|L_f(\mathcal{A})\right\| = \max_{\mathcal{C}\neq 0} \frac{\left\|\mathtt{vec}\big(L_f(\mathcal{A},\mathcal{C})\big)\right\|_2}{\|\mathtt{vec}(\mathcal{C})\|_2} = \max_{\mathcal{C}\neq 0} \frac{\left\|K_f(\mathcal{A})\mathtt{vec}(\mathcal{C})\right\|_2}{\|\mathtt{vec}(\mathcal{C})\|_2} = \left\|K_f(\mathcal{A})\right\|_2.$$

□

For realistic problem sizes, it will typically not be feasible to compute the condition number of $f(\mathcal{A})$ via (41). This is already the case for functions of $n \times n$ matrices, and it becomes even more prohibitive in the tensor setting. As outlined at the end of Sect. 3.4, simply forming the Kronecker form $K_f(\mathcal{A})$ has cost $\mathcal{O}(n^5 p^4)$ and requires $\mathcal{O}(n^4 p^2)$ storage. Even for moderate values of n and p, this is typically not possible.

Instead, we need to approximate the condition number. As a rough estimate is usually sufficient, a few steps of power iteration typically give a satisfactory result, as one is mainly interested in the order of magnitude of the condition number, so that more than one significant digit is seldom needed. Algorithm 2 is a straightforward adaptation of [15, Algorithm 3.20], which computes an estimate of $\left\|K_f(A)\right\|_2$ by applying power iteration to the Hermitian matrix $K_f(A)^H K_f(A)$, exploiting that a matrix vector multiplication $K_f(A)\boldsymbol{v}$ is equivalent to the evaluation of $L_f(A, \mathtt{unvec}(\boldsymbol{v}))$, where

$\texttt{unvec}(\boldsymbol{v})$ maps the vector $\boldsymbol{v}$ to an unstacked matrix of the same size as A. In line 4, the function $\overline{f}$ is defined via $\overline{f}(z) = \overline{f(\overline{z})}$.

Remark 6 As Algorithm 2 boils down to a matrix power iteration, its asymptotic convergence rate is linear and depends on the magnitude of the ratio between the eigenvalue of largest and second largest magnitude of the Hermitian matrix $K_f(\mathcal{A})^H K_f(\mathcal{A})$; see e.g., [13, Eq. (7.3.5)]. It is quite difficult, however, to give meaningful a priori bounds on this ratio, as we do not have explicit formulas for the eigenvalues or singular values of $K_f(\mathcal{A})$ available (in terms of spectral quantities related to $\mathcal{A}$), and deriving such relations is well beyond the scope of this work.

Also, note that typically only $\mathcal{O}(1)$ iterations of Algorithm 2 are sufficient due to the rather low accuracy requirements in condition number estimation; see our experiments reported in Sect. 6.3 as well as, e.g., [15, 23] for the matrix function case. In these early iterations, the asymptotic convergence rate will likely not be descriptive concerning the actual behavior of the method, as it does not capture the fast reduction of contributions from eigenvectors corresponding to small eigenvalues.

Algorithm 2 Power iteration for the t-Fréchet derivative

```
    Input:  $f$, $\mathcal{A}$, tol, max_it
    Output: Estimate $\gamma \approx \|L(\mathcal{A})\|$
 1: Choose $\mathcal{C}_1 \in \mathbb{C}^{n \times n \times p}$ at random
 2: for $k = 1, \ldots,$ max_it do
 3:     $\mathcal{B}_{k+1} \leftarrow L_f(\mathcal{A}, \mathcal{C}_k)$
 4:     $\mathcal{C}_{k+1} \leftarrow L_{\overline{f}}(\mathcal{A}^H, \mathcal{B}_{k+1})$
 5:     $\gamma_{k+1} \leftarrow \|\mathcal{C}_{k+1}\| \,/\, \|\mathcal{B}_{k+1}\|$
 6:     if $|\gamma_{k+1} - \gamma_k| \leq$ tol $\cdot \gamma_{k+1}$ then
 7:         break
 8:     end if
 9: end for
10: $\gamma \leftarrow \gamma_{k+1}$
```

Algorithm 2 is necessarily sequential with respect to calls of $L_f(\mathcal{A}, \cdot)$. An alternative algorithm that would lend itself naturally to parallelization (especially in the case that $n \ll p$) stems from Lemma 4 and Proposition 3, and is a variant implementation of Algorithm 1. In the first phase, $K_f(\texttt{bcirc}(\mathcal{A}))$ is computed but in a reduced fashion, whereby only n^2 applications of $L_f(\texttt{bcirc}(\mathcal{A}), \cdot)$ are required, thanks to the shift relation proven in Proposition 2. This first step can be trivially parallelized, as it is known a priori exactly on which unit matrices to call $L_f(\texttt{bcirc}(\mathcal{A}), \cdot)$. In the second phase, the columns of $K_f(\mathcal{A})$ are assembled via Lemma 4. While Algorithm 1 can similarly be trivially parallelized, the approach outlined in Algorithm 3 guarantees n^2 calls to $L_f(\texttt{bcirc}(\mathcal{A}), \cdot)$ overall, as opposed to $n^2 p$ in Algorithm 1.

Algorithm 3 Kronecker form of the t-Fréchet derivative (efficient approach)

```
Input: $f, \mathcal{A}$
Output: Kronecker form $K = K_f(\mathcal{A})$
Allocate memory for $n^2$ matrices $\mathcal{Y}_{IJ} \in \mathbb{C}^{np \times np}, I, J = 1, \dots, n$
for $I = 1, \dots, n$ do
    for $J = 1, \dots, n$ do
        $\mathcal{Y}_{IJ} \leftarrow L_f(\texttt{bcirc}(\mathcal{A}), E_{IJ})$, $E_{IJ} = \boldsymbol{e}_1^T \otimes \texttt{unfold}(\mathcal{E}_{ijk})$
    end for
end for
for $i = 1, \dots, n$ do
    for $j = 1, \dots, n$ do
        for $k = 1, \dots, p$ do
            $\mathcal{X}_\ell \leftarrow \texttt{vec}\big(\mathcal{S}^{\ell-1}(\mathcal{Y}_{ij})\big)$, $\ell = 1, \dots, p$
            $K(:, i + (k-1)n + (j-1)np) \leftarrow \sum_{\ell=1}^{p} \mathcal{X}_\ell(1 : n^2 p)$
        end for
    end for
end for
```

We end this section by briefly discussing the connection between conditioning of the t-function $f(\mathcal{A})$ and the matrix function $f(\texttt{bcirc}(\mathcal{A}))$. In light of (40) and the definition of $f(\mathcal{A})$ in terms of block circulant matrices, it is immediate that

$$\mathrm{cond}_{\mathrm{abs}}(f, \mathcal{A}) \leq \mathrm{cond}_{\mathrm{abs}}(f, \texttt{bcirc}(\mathcal{A})), \tag{42}$$

where $\mathrm{cond}_{\mathrm{abs}}(f, \texttt{bcirc}(\mathcal{A}))$ denotes the matrix function condition number in the Frobenius norm: the left-hand side of (42), when interpreted in terms of the underlying matrix function, only allows *structured, block-circulant perturbations*, while the right-hand side measures conditioning with respect to *any* perturbation. Often, such structured condition numbers can be significantly lower than unstructured condition numbers; see, e.g., [2, 7]. In our experiments, we have actually observed equality in (42) in most test cases, at least up to machine precision, but it is also possible to construct examples in which the two condition numbers disagree by a large margin; see, e.g., the test script `test_cond_counter_ex.m` in our code suite. It might be an interesting question for further research to find out whether there are conditions on f and/or $\mathcal{A}$ that guarantee equality holds in (42).

5.2 The gradient of the tensor nuclear norm

In this section, we highlight an example application of how our framework for the t-Fréchet derivative can be useful for deriving certain theoretical results in a rather straightforward fashion.

The *nuclear norm* of a tensor is typically defined in terms of a tensor singular value decomposition (see, e.g., [30]), but it was recently shown that it can also be computed in terms of the t-square root as

$$\|\mathcal{A}\|_\star = \mathrm{trace}_{(1)}(\sqrt{\mathcal{A}^T * \mathcal{A}}),$$

where $\mathrm{trace}_{(1)}$ denotes the trace of the first frontal slice; see [4, Lemma 6]. Tensor nuclear norm minimization is an important tool in image completion, low-rank tensor completion, denoising, seismic data reconstruction, and principal component analysis; see, e.g., [3, 6, 18, 26, 29, 30, 38, 39]. In these applications, it can be of interest to compute the *gradient* of the tensor nuclear norm for a gradient descent scheme.[4] We will now derive an explicit formula for the gradient of $\|\mathcal{A}\|_\star$ in terms of t-functions, which is reminiscent of similar results in the matrix case.

To do so, we first collect some auxiliary results on the $\mathrm{trace}_{(1)}$ operator. Clearly, $\mathrm{trace}_{(1)}$ is linear, and by direct computation, it is easy to verify that

$$\|\mathcal{A}\| = \sqrt{\mathrm{trace}_{(1)}(\mathcal{A}^T * \mathcal{A})},$$

where $\|\cdot\|$ is the tensor norm defined in (5) and that

$$\langle \mathcal{A}, \mathcal{B} \rangle := \mathrm{trace}_{(1)}(\mathcal{B}^T * \mathcal{A}) \tag{43}$$

defines an inner product on $\mathbb{C}^{n\times n\times p}$ (which corresponds to the standard inner product on $\mathbb{C}^{n^2 p}$ for the vectorized tensors).

Further, the $\mathrm{trace}_{(1)}$ operator inherits the cyclic property of the trace, with respect to the t-product.

Lemma 7 *Let $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{n\times n\times p}$. Then*

$$\mathrm{trace}_{(1)}(\mathcal{A} * \mathcal{B}) = \mathrm{trace}_{(1)}(\mathcal{B} * \mathcal{A}).$$

Proof By the definition of the t-product $\mathcal{A} * \mathcal{B} := \mathtt{fold}(\mathtt{bcirc}(\mathcal{A})\mathtt{unfold}(\mathcal{B}))$, the first face of $\mathcal{A} * \mathcal{B}$ is the first $n \times n$ block of $\mathtt{bcirc}(\mathcal{A})\mathtt{unfold}(\mathcal{B})$, which is given by

$$[\mathtt{bcirc}(\mathcal{A})\mathtt{unfold}(\mathcal{B})]_{1:n,:} = A^{(1)}B^{(1)} + A^{(p)}B^{(2)} + \cdots + A^{(2)}B^{(p)}. \tag{44}$$

Similarly, the first face of $\mathcal{B} * \mathcal{A}$ is

$$[\mathtt{bcirc}(\mathcal{B})\mathtt{unfold}(\mathcal{A})]_{1:n,:} = B^{(1)}A^{(1)} + B^{(p)}A^{(2)} + \cdots + B^{(2)}A^{(p)}. \tag{45}$$

Using the linearity and the cyclic property of the trace, it is clear that the traces of (44) and (45) agree, thus proving the result of the lemma. □

Lemma 7 together with Lemma 3 leads to a useful representation for the derivative of $\mathrm{trace}_{(1)}(f(\mathcal{A}))$ when f is analytic, involving the derivative of the scalar function f. By a slight abuse of notation, we write the Fréchet derivative (in the sense of the general definition (2)) of $\mathrm{trace}_{(1)}$ at a tensor $\mathcal{M}$ as $L_{\mathrm{trace}_{(1)}}(\mathcal{M}, \cdot)$, although it is clearly not a t-function.

[4] We note that the tensor nuclear norm is clearly not differentiable at all tensors $\mathcal{A}$, so one might also need to consider subgradients in certain applications, but this is well beyond the scope of this paper. We therefore only focus on the differentiable case here.

Lemma 8 *Let $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ and let f be analytic on a region containing the spectrum of* $\texttt{bcirc}(\mathcal{A})$. *Then*

$$L_{\text{trace}_{(1)} \circ f}(\mathcal{A}, \mathcal{C}) = \text{trace}_{(1)}(f'(\mathcal{A}) * \mathcal{C}).$$

Proof By the linearity of $\text{trace}_{(1)}$ we directly obtain

$$L_{\text{trace}_{(1)}}(\mathcal{A}, \mathcal{C}) = \text{trace}_{(1)}(\mathcal{C}).$$

As the chain rule, Lemma 2*(iii)*, also holds more generally for any Fréchet differentiable functions, not necessarily t-functions, we have

$$L_{\text{trace}_{(1)} \circ f}(\mathcal{M}, \mathcal{C}) = L_{\text{trace}_{(1)}}(f(\mathcal{M}), L_f(\mathcal{M}, \mathcal{C})) = \text{trace}_{(1)}(L_f(\mathcal{M}, \mathcal{C})). \quad (46)$$

By Lemma 3, we can further rewrite (46) as

$$\begin{aligned} L_{\text{trace}_{(1)} \circ f}(\mathcal{M}, \mathcal{C}) &= \text{trace}_{(1)} \left(\frac{1}{2\pi i} \int_\Gamma f(\zeta)(\zeta\mathcal{I} - \mathcal{A})^{-1} * \mathcal{C} * (\zeta\mathcal{I} - \mathcal{A})^{-1} \, \mathrm{d}\zeta \right). \\ &= \text{trace}_{(1)} \left(\frac{1}{2\pi i} \int_\Gamma f(\zeta)(\zeta\mathcal{I} - \mathcal{A})^{-2} \, \mathrm{d}\zeta * \mathcal{C} \right). \quad (47) \end{aligned}$$

where we have used the cyclic property of $\text{trace}_{(1)}$ with respect to the t-product from Lemma 7 for the second equality. The integral in (47) is the Cauchy integral representation of $f'(\mathcal{A})$, thus completing the proof. □

We are now in a position to state the main result of this section. Note that using the inner product (43), the gradient of the nuclear norm can be characterized by imposing the condition

$$L_{\|\cdot\|_\star}(\mathcal{A}, \mathcal{C}) = \langle \mathcal{C}, \nabla_\mathcal{A} \|\mathcal{A}\|_\star \rangle = \text{trace}_{(1)} \left(\nabla_\mathcal{A} \|\mathcal{A}\|_\star\right)^T * \mathcal{C}, \quad (48)$$

for all $\mathcal{C} \in \mathbb{C}^{n \times n \times p}$.

Theorem 3 *Let $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ be such that $(\mathcal{A}^T * \mathcal{A})^{-1/2}$ is defined. Then $\|\cdot\|_\star$ is differentiable at $\mathcal{A}$ and*

$$\nabla_A \|\mathcal{A}\|_\star = \mathcal{A} * (\mathcal{A}^T * \mathcal{A})^{-1/2}.$$

Proof Define $f(\mathcal{M}) = \mathcal{M}^T * \mathcal{M}$, $g(\mathcal{M}) = \sqrt{\mathcal{M}}$, so that $\|\mathcal{A}\|_\star = (\text{trace}_{(1)} \circ g \circ f)(\mathcal{A})$, where f is *not* a tensor t-function in the usual sense. As before, with slight abuse of notation, we write $L_f(\mathcal{M}, \cdot)$ for its Fréchet derivative. From the definition of the t-product, it is straightforward to verify that

$$L_f(\mathcal{A}, \mathcal{C}) = \mathcal{A}^T * \mathcal{C} + \mathcal{C}^T * \mathcal{A}. \quad (49)$$

Using the chain rule and Lemma 8, we have

$$\begin{aligned} L_{\mathrm{trace}_{(1)} \circ g \circ f}(\mathcal{A}, \mathcal{C}) &= L_{\mathrm{trace}_{(1)} \circ g}(f(\mathcal{A}), L_f(\mathcal{A}, \mathcal{C})) \\ &= \mathrm{trace}_{(1)}(g'(f(\mathcal{A})) * L_f(\mathcal{A}, \mathcal{C})). \end{aligned} \tag{50}$$

As g is the square root, we have $g'(f(\mathcal{A})) = \frac{1}{2}(\mathcal{A}^T * \mathcal{A})^{-1/2}$, so that by combining (49) and (50), we find

$$\begin{aligned} L_{\|\cdot\|_\star}(\mathcal{A}, \mathcal{C}) &= L_{\mathrm{trace}_{(1)} \circ g \circ f}(\mathcal{A}, \mathcal{C}) \\ &= \frac{1}{2} \mathrm{trace}_{(1)}((\mathcal{A}^T * \mathcal{A})^{-1/2} * \mathcal{A}^T * \mathcal{C} + (\mathcal{A}^T \mathcal{A})^{-1/2} * \mathcal{C}^T * \mathcal{A}) \\ &= \frac{1}{2} \mathrm{trace}_{(1)}((\mathcal{A}^T * \mathcal{A})^{-1/2} * \mathcal{A}^T * \mathcal{C}) \\ &\quad + \frac{1}{2} \mathrm{trace}_{(1)}(\mathcal{C}^T * \mathcal{A} * (\mathcal{A}^T * \mathcal{A})^{-1/2}) \\ &= \mathrm{trace}_{(1)}((\mathcal{A}^T * \mathcal{A})^{-1/2} * \mathcal{A}^T * \mathcal{C}), \end{aligned} \tag{51}$$

where we have used the cyclic property of $\mathrm{trace}_{(1)}$ for the second equality and the fact that $\mathrm{trace}_{(1)}(\mathcal{M}^T) = \mathrm{trace}_{(1)}(\mathcal{M})$, which directly follows from the definition of tensor t-transposition, together with the linearity of $\mathrm{trace}_{(1)}$ for the third equality. Comparing (51) and (48) shows that

$$\nabla_A \|\mathcal{A}\|_\star = \mathcal{A} * (\mathcal{A}^T * \mathcal{A})^{-1/2},$$

thus concluding the proof. □

To illustrate the theory, the script `test_t_nuclear_norm.m` in our code suite implements a simple gradient descent scheme with backtracking line search for nuclear norm minimization, based on Theorem 3.

6 Numerical experiments

In this section, we detail a software framework for studying the performance of the proposed algorithms and present numerical results from several small- to medium-scale experiments.

6.1 Implementation details

We have developed our own modular toolbox, `t-Frechet`, hosted at https://gitlab.com/katlund/t-frechet. The basic syntax is derived from `bfomfom`[5] and `LowSyncBlockArnoldi`.[6] We note that in contrast to an existing t-product toolbox

[5] https://gitlab.com/katlund/bfomfom-main.

[6] https://gitlab.mpi-magdeburg.mpg.de/lund/low-sync-block-arnoldi.

Table 1 Features of numerical approaches for computing $L_f(\mathcal{A}, \mathcal{C})$

Approach	Operator (Op.)	Op. size	No. of Op.	Sparse op.?	Transpose required?	Restarts allowed?
`bcirc`, (20)	`bcirc`($\mathcal{A}$)	$np \times np$	1	Y	N	Y
`low-rank`, Sect. 4.1	`bcirc`($\mathcal{A}$)	$np \times np$	2	Y	Y	N
`dft`, Cor. 1	$D_{\mathcal{A}}$	$n \times n$	p	N	N	Y

Note that for `low-rank`, the number of operators refers to the fact that the transpose is needed, which is nontrivial if $\mathcal{A}$ is only known implicitly or via a black-box routine. As for `dft`, $D_{\mathcal{A}}$ represents all p subproblems

`Tensor-tensor-product-toolbox`,[7] a tensor $\mathcal{A}$ in `t-Frechet` is encoded as a MATLAB struct with fields `mat` and `dim`, which store `unfold`($\mathcal{A}$) and $\mathcal{A}$'s dimensions as a vector $[n\, m\, p]$, respectively. Such tensor structs allow us to work with sparse tensors via built-in MATLAB functions and compute the actions of block circulant matrices without ever explicitly forming the full $np \times mp$ matrix. Our toolbox has been tested in MATLAB 2019b, 2022a, and 2023a on Ubuntu and Windows machines.

Table 1 summarizes features of the three methods for approximating $L_f(\mathcal{A}, \mathcal{C})$ that we have derived throughout the text. Regarding the `dft` approach, note that equation (35) can be trivially implemented on (dense) third-order arrays in MATLAB, thanks to `fft` and `ifft`; see comments in [25] as well as our test script `test_dft`. A number of additional test scripts are included in `t-Frechet` that we do not discuss here; we have, however, kept them public to encourage further engagement with the community.

6.2 Comparing performance of t-Fréchet implementations

We consider a simple example for examining the performance of the proposed solvers by taking $f(z) = \exp(z)$ and $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ such that each face of $\mathcal{A}$ is a finite differences stencil for the spatial components of the two-dimensional convection-diffusion equation

$$u_t = -\Delta(u_{xx} + u_{yy}) + \nu(u_x + u_y)$$

with the convection parameter ν drawn p times uniformly from the interval $[0, 200]$. We restrict both spatial variables to the unit square and take $\sqrt{n}$ points in each direction, where $n \in \{36, 144, 576\}$. The direction tensor $\mathcal{C}$ is dense and its entries are randomly drawn from the normal distribution.

All scripts are executed in MATLAB R2022a on 16 threads of a single, standard node of the Linux Cluster Mechthild at the Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg, Germany.[8] We report the total run time

[7] https://github.com/canyilu/Tensor-tensor-product-toolbox.

[8] A standard node comprises 2 Intel Xeon Silver 4110 (Skylake) CPUs with 8 Cores each (64KB L1 cache, 1024KB L2 cache), a clockrate of 2.1 GHz (3.0 GHz max), and 12MB shared L3 cache each.

to reach a tolerance of 10^{-6}, percentage speed-up, number of times the operator (see Table 1) is called, and the final error for all three approaches. Each approach is run 10 times, and the reported times are an average over these runs. Unless otherwise mentioned, B(FOM)^2 [12] with the classical inner product and block modified Gram-Schmidt was employed to compute the matrix functions. Note that aside from node-level multithreading, all algorithms are run in serial.

6.2.1 Small problem: $n = 36, p = 10$

The performance is similar for all algorithms for this small problem size, which leads to matrix function problems of size 360 × 360 for `bcirc` and `low-rank`, and 36 × 36 for `dft`. However, both `low-rank` and `dft` converge very quickly—1 and 2 iterations, respectively—and achieve high accuracy. Recall that both the `low-rank` and `dft` approaches rely on multiple operators per iteration. Accuracy for `dft` is measured as an average across all subproblems. See Table 6.2.1 for performance data and Figure 6.2.1 for error plots of `bcirc`.

Configuration	Time (s)	% Speed-up	Op. count	Final error
`bcirc`	0.41	0.00	13	6.1252e-07
`low-rank`	0.19	53.56	2	4.4444e-15
`dft`	0.14	65.33	20	2.5940e-15

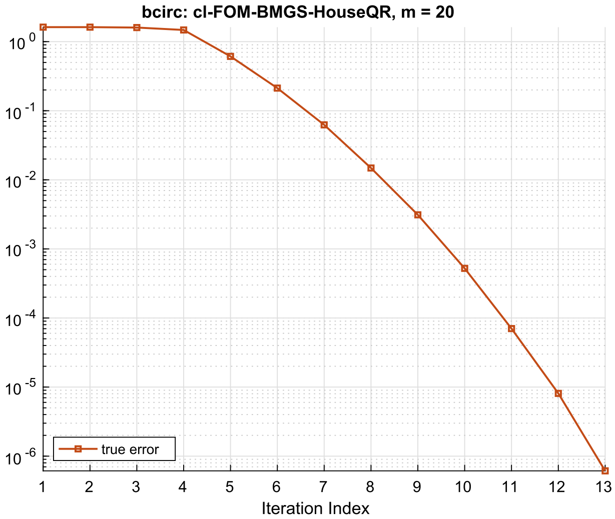

Fig. 2 Error plot for `bcirc` configuration of the Fréchet derivative solver on the small problem, $n = 36$, $p = 10$

6.2.2 Medium problem: $n = 144, p = 10$

With a larger problem size we begin to see clear performance differences among the three methods. Matrix function problems are now 1440×1440 for `bcirc` and `low-rank`, and 144×144 for `dft`. Both `bcirc` and `low-rank` struggle to compete with `dft`, which is an order of magnitude faster, due to computing with much smaller matrices. Furthermore, `dft` has no apparent accuracy issues, achieving near machine precision in 2 iterations, while `low-rank` achieves a similar accuracy in 1 iteration and `bcirc` just passes the desired tolerance after 14 iterations. See Table 6.2.2 for performance data and Figure 6.2.2 for error plots of `bcirc`.

Configuration	Time (s)	% Speed-up	Op. count	Final error
`bcirc`	6.78	0.00	14	4.3093e-07
`low-rank`	3.94	41.91	2	6.8581e-15
`dft`	0.70	89.74	20	7.3250e-15

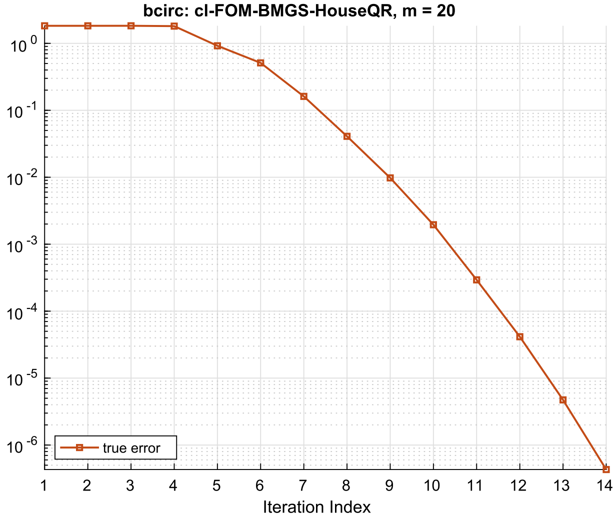

Fig. 3 Error plot for `bcirc` configuration of the Fréchet derivative solver on the medium problem, $n = 144$, $p = 10$

6.2.3 Large problem: $n = 576, p = 10$

As we quadruple the problem size, the situation remains nearly identical to when $n = 144$. The `dft` approach remains significantly faster than either `bcirc`, which still struggles to achieve better accuracy, and `low-rank`, which despite requiring only 1 iteration is overall as slow as `bcirc`. See Table 6.2.3 for performance data and Figure 6.2.3 for error plots of `bcirc`. Note that due to the longer run time for this problem, we averaged timings over 5 instead of 10 runs.

Configuration	Time (s)	% Speed-up	Op. count	Final error
`bcirc`	262	0.00	14	9.0656e-07
`low-rank`	204	22.02	2	2.2185e-14
`dft`	12.1	95.40	20	1.5919e-14

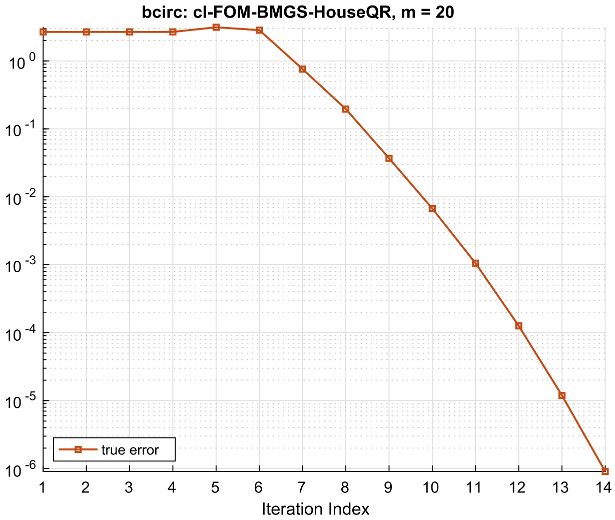

Fig. 4 Error plot for `bcirc` configuration of the Fréchet derivative solver on the large problem, $n = 576$, $p = 10$

6.3 Accuracy and effort of t-condition number solvers

For testing condition number algorithms, we fix the t-Fréchet solver to be an "exact" (non-iterative) method. We then study how different approaches fare with respect to the number of times they invoke a t-Fréchet solver, simply denoted as `t_frechet`. We take $f(z) = \exp(z)$ and $\mathcal{A}$ a dense $n \times n \times p$ tensor, whose entries are drawn randomly from the normal distribution. We set a tolerance of 10^{-2} for the power iteration, and we compare it with the "full" Kronecker form approach (Algorithm 1), which we also treat as ground truth, and the "efficient" Kronecker form approach (Algorithm 3).

For all the tests in this section, we only look at a single run, as computing the full Kronecker form is time-consuming.

6.3.1 Big faces: $n = 20, p = 5$

For the first example, we consider the case where $n > p$. Results are summarized in Table 6.3.1. The power iteration is clearly the winning method here, with only 8 calls to `t_frechet` necessary to achieve the desired tolerance. While the efficient Kronecker approach does reduce the overall time in comparison to the full Kronecker approach, it is not competitive with the power iteration.

Method	Time (s)	`t_frechet` calls	Time (s) per call	Accuracy
Power iteration	0.03	8	3.43e-02	3.3923e-03
Efficient Kronecker	7.95	400	1.99e-02	4.3122e-16
Full Kronecker	31.8	2000	1.59e-02	0.0000e+00

6.3.2 All things equal: $n = 10, p = 10.$

We now examine the scenario where $n = p$. Results are found in Table 6.3.2. The power iteration remains significantly faster than both Kronecker form competitors, and it still achieves the desired tolerance.

Method	Time (s)	`t_frechet` calls	Time (s) per call	Accuracy
Power iteration	0.01	6	1.57e-02	3.1937e-03
Efficient Kronecker	1.16	100	1.16e-02	3.6995e-16
Full Kronecker	12.1	1000	1.21e-02	0.0000e+00

6.3.3 Many faces: $n = 5, p = 50$

We finally consider $n \ll p$; see Table 6.3.3 for the results. The power iteration remains overwhelmingly faster than the efficient Kronecker approach, and still achieves the desired tolerance.

Method	Time (s)	`t_frechet` calls	Time (s) per call	Accuracy
Power iteration	1.33	6	2.21e-01	1.3325e-06
Efficient Kronecker	16.5	25	6.60e-01	0.0000e+00
Full Kronecker	189	1250	1.51e-01	0.0000e+00

A clear drawback of the analysis in this section is that, in practice, one will not be able to compute Fréchet derivatives with high accuracy. However, in most applications that require a condition number, accuracy is unimportant. In which case it is sufficient to replace the inner `t_frechet` solves of the power iteration with, for example, the `dft` approach from Corollary 1.

When accuracy is important, however, the efficient Kronecker approach may be a viable competitor to the power iteration. In all examples, we see that the time per `t_frechet` evaluation is roughly the same per method. Because all the `t_frechet` problems are known a priori and they are far fewer than in the full Kronecker approach, the efficient Kronecker procedure is trivially parallelizable, unlike the power iteration, which is necessarily serial. In the case with many faces (i.e., $n < p$), where relatively few `t_frechet` calls overall are necessary, a simple parallelization could easily give the efficient Kronecker approach an edge.

7 Conclusions

Thanks to the block circulant structure imposed by the t-product formalism, we have been able to take advantage of a rich mathematical framework not only in the definition of the Fréchet derivative of the tensor t-function but also in the development of efficient and accurate algorithms for its numerical approximation. We have proven a number of useful properties of the t-Fréchet derivative, including a Daleckiĭ-Kreĭn-type result. An expression for the gradient of the nuclear norm has also been derived and its utility demonstrated in a gradient descent scheme for nuclear norm minimization. We have affirmed the indispensability of the discrete Fourier transform (DFT) in accelerating the computation of the t-Fréchet derivative itself, as the DFT decouples the problem into p smaller problems that each converge in few iterations. We have further shown the utility of the t-Fréchet derivative in t-function condition number estimation. A tailored power iteration algorithm has proven efficient for reliably computing the condition number at a high tolerance. We have also demonstrated that the full Kronecker form of the t-Fréchet derivative can be computed in p times less work than a direct approach thanks to symmetries evoked by the block circulant structure. Finally, we have developed and made public a modular t-product toolbox that will prove foundational in exploring further, more challenging applications.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Al-Mohy, A.H., Higham, N.J.: Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation. SIAM J. Matrix Anal. Appl. **30**(4), 1639–1657 (2009). https://doi.org/10.1137/080716426
2. Arslan, B., Noferini, V., Tisseur, F.: The structured condition number of a differentiable map between matrix manifolds, with applications. SIAM J. Matrix Anal. Appl. **40**(2), 774–799 (2019). https://doi.org/10.1137/17M114894
3. Bentbib, A.H., El Ghomari, M., Jbilou, K., Reichel, L.: The global Golub–Kahan method and Gauss quadrature for tensor function approximation. Numer. Algorithms (2022). https://doi.org/10.1007/s11075-022-01392-x
4. Bentbib, A.H., El Hachimi, A., Jbilou, K., Ratnani, A.: A tensor regularized nuclear norm method for image and video completion. J. Opt. Th. Appl. **192**(2), 401–425 (2022). https://doi.org/10.1007/s10957-021-01947-3
5. Braman, K.: Third-order tensors as linear operators on a space of matrices. Linear Algebra Appl. **433**(7), 1241–1253 (2010). https://doi.org/10.1016/j.laa.2010.05.025
6. Canyi, L., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, Shuicheng: Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Trans. Pattern Anal. Mach. Intell. **42**(4), 925–938 (2020). https://doi.org/10.1109/TPAMI.2019.2891760
7. Davies, P.: Structured conditioning of matrix functions. Electron. J. Linear Algebra, 11:132–161 (2004). https://doi.org/10.13001/1081-3810.1128
8. Davis, P.J.: Circulant Matrices, 2nd edn. AMS Chelsea Publishing, Providence (2012)
9. De la Cruz Cabrera, O., Jin, J., Noschese, S., Reichel, L.: Communication in complex networks. Appl. Numer. Math. **172**:186–205 (2022). https://doi.org/10.1016/j.apnum.2021.10.005
10. Elizabeth, N., Lior, H., Haim, A., Misha K.: Stable tensor neural networks for rapid deep learning (2018). arXiv: 1811.06569
11. Estrada, E., Higham, D.J.: Network properties revealed through matrix functions. SIAM Rev. **52**(4), 696–714 (2010). https://doi.org/10.1137/090761070
12. Frommer, Andreas, Lund, Kathryn, Szyld, Daniel B.: Block Krylov subspace methods for functions of matrices. Electron. Trans. Numer. Anal. **47**, 100–126 (2017). https://doi.org/10.1553/etna_vol47s100
13. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins Studies in the Mathematical Sciences, 4th edn. Johns Hopkins University Press, Baltimore (2013)
14. Gutknecht, M. H.: Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. In: Siddiqi, A. H., Duff, I. S., Christensen, O. (eds) Mod. Math. Model. Methods Algorithms Real World Syst., pages 420–447, New Delhi, Anamaya (2007)
15. Higham, N. J.: Functions of matrices: theory and computation. Applied Mathematics. SIAM Publications, Philadelphia (2008). https://doi.org/10.1137/1.9780898717778
16. Hochbruck, M., Ostermann, A.: Exponential integrators. Acta Numer. **19**, 209–286 (2010). https://doi.org/10.1017/S0962492910000048
17. Hoover, R. C., Caudle, K., Braman, K.: A new approach to multilinear dynamical systems and control (2021). arXiv: 2108.13583
18. Hosono, K., Ono, S., Miyata, T.: Weighted tensor nuclear norm minimization for color image denoising. In: 2016 IEEE Int. Conf. Image Process. ICIP, pages 3081–3085. IEEE (2016). https://doi.org/10.1109/ICIP.2016.7532926
19. Ilić, M., Turner, I.W., Simpson, D.P.: A restarted Lanczos approximation to functions of a symmetric matrix. IMA J. Numer. Anal. **30**(4), 1044–1061 (2010). https://doi.org/10.1093/imanum/drp003
20. Kandolf, P., Relton, S.D.: A block Krylov method to compute the action of the Fréchet derivative of a matrix function on a vector with applications to condition number estimation. SIAM J. Sci. Comput. **39**(4), A1416–A1434 (2017). https://doi.org/10.1137/16M1077969
21. Kandolf, P., Koskela, A., Relton, S.D., Schweitzer, M.: Computing low-rank approximations of the Fréchet derivative of a matrix function using Krylov subspace methods. Numer. Lin. Alg. Appl. **28**(6), e2401 (2021). https://doi.org/10.1002/nla.2401
22. Kathryn, L.: The tensor t-function: A definition for functions of third-order tensors. Numer Linear Algebra Appl. **27**(3) (2020). https://doi.org/10.1002/nla.2288
23. Kenney, C., Laub, A.J.: Condition estimates for matrix functions. SIAM J. Matrix Anal. Appl. **10**(2), 191–209 (1989). https://doi.org/10.1137/0610014

24. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algebra Appl. **435**(3), 641–658 (2011). https://doi.org/10.1016/j.laa.2010.09.020
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**(1), 148–172 (2013). https://doi.org/10.1137/110837711
26. Kreimer, N., Stanton, A., Sacchi, M.D.: Tensor completion based on nuclear norm minimization for 5D seismic data reconstruction. Geophysics **78**(6), 1942–2156 (2013). https://doi.org/10.1190/geo2013-0022.1
27. Kressner, D.: A Krylov subspace method for the approximation of bivariate matrix functions. In *Structured matrices in numerical linear algebra*, pages 197–214. Springer-Verlag, Cham (2019). https://doi.org/10.1007/978-3-030-04088-8_10
28. Liu, W., Jin, X.: A study on T-eigenvalues of third-order tensors. Linear Algebra Appl. **612**, 357–374 (2021). https://doi.org/10.1016/j.laa.2020.11.004
29. Liu, M., Zhang, X., Tang, L.: Real color image denoising using t-product- based weighted tensor nuclear norm minimization. IEEE Access **7**, 182017–182026 (2019). https://doi.org/10.1109/ACCESS.2019.2960078
30. Lu, C., Peng, X., Wei, Y.: Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In: 2019 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, pages 5989–5997, Long Beach, CA, USA. IEEE (2019). https://doi.org/10.1109/CVPR.2019.00615
31. Malik, O.A., Ubaru, S., Horesh, L., Kilmer, M. E., Avron, H.: Tensor graph neural networks for learning on time varying graphs. In: NeurIPS 2019 Workshop Graph Represent. Learn (2019)
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). https://doi.org/10.1016/j.laa.2019.12.035
33. Neuberger, H.: Exactly massless quarks on the lattice. Phys. Lett. B **417**(1–2), 141–144 (1998). https://doi.org/10.1016/S0370-2693(97)01368-3
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**(1), 1–39 (2022). https://doi.org/10.1007/s10915-021-01719-1
35. Schweitzer, M.: Sensitivity of matrix function based network communicability measures: computational methods and a priori bounds (2023). arXiv e-print 2303.01339. https://doi.org/10.48550/arXiv.2303.01339
36. Schweitzer, M.: Integral representations for higher-order Fréchet derivatives of matrix functions: quadrature algorithms and new results on the level-2 condition number. Linear Algebra Appl. **656**, 247–276 (2023). https://doi.org/10.1016/j.laa.2022.10.005
37. Thanou, D., Dong, X., Kressner, D., Frossard, P.: Learning heat diffusion graphs. IEEE Trans Signal Inf. Process Netw. **3**(3), 484–499 (2017). https://doi.org/10.1109/TSIPN.2017.2731164
38. Yuan, M., Zhang, C.-H.: On tensor completion via nuclear norm minimization. Found. Comput. Math. **16**(4), 1031–1068 (2016). https://doi.org/10.1007/s10208-015-9269-5
39. Zhang, X., Ng, M.K.: A corrected tensor nuclear norm minimization method for noisy low-rank tensor completion. SIAM J. Imaging Sci. **12**(2), 1231–1273 (2019). https://doi.org/10.1137/18M1202311