



The tensor t-function: A definition for functions of third-order tensors

Kathryn Lund^{ID}

Charles University, Prague, Czech Republic

Correspondence

Kathryn Lund, Charles University, Prague, Czech Republic.
Email: kathryn.lund@karlin.mff.cuni.cz

Funding information

U.S. National Science Foundation, grant DMS-14188822; U.S. Department of Energy, grant DE-SC 00165783; Charles University, PRIMUS grant, project no. PRIMUS/19/SCI/11

Summary

A definition for functions of multidimensional arrays is presented. The definition is valid for third-order tensors in the tensor t-product formalism, which regards third-order tensors as block circulant matrices. The tensor function definition is shown to have similar properties as standard matrix function definitions in fundamental scenarios. To demonstrate the definition's potential in applications, the notion of network communicability is generalized to third-order tensors and computed for a small-scale example via block Krylov subspace methods for matrix functions. A complexity analysis for these methods in the context of tensors is also provided.

KEY WORDS

block circulant matrices, matrix functions, multidimensional arrays, network analysis, tensor t-product, tensors

MOS SUBJECT CLASSIFICATION

15A69; 65F10; 65F60

1 | INTRODUCTION

Functions of matrices—that is, $f(A)$, where f is a scalar function and A is a square matrix—have applications in a number of fields. They emerge as measures of centrality and communicability in networks¹ and as exponential integrators in differential equations.² As high-dimensional analogs of matrices, tensors also play crucial roles in network analysis³ and multidimensional differential equations.⁴ A variety of decompositions and algorithms have been developed over the years to extract and understand properties of tensors.⁵ A natural question is whether the notion of functions of tensors, defined in analogy to functions of matrices as a scalar function taking a tensor \mathcal{A} as its argument, could prove to be yet another useful tool for studying multidimensional data.

Unfortunately, the definition of such a notion is not nearly as straightforward for tensors as it is for matrices. For matrices, the definitions of integration, polynomials, eigendecompositions (ED), and singular value decompositions (SVD) are unique and well established throughout linear algebra, and all of these notions serve as building blocks for definitions of matrix functions, reducing to the same object under reasonable circumstances.^{6,7} With the abundance of generalizations for the ED and SVD in the world of tensors,^{5,8–14} there is no guarantee that a tensor function definition based on one type of decomposition is equivalent to a definition based on another type.

With that in mind, we embark on a first look at defining functions of tensors via a simple paradigm, the tensor t-product formalism.^{9,15,16} This paradigm effectively regards third-order tensors as block circulant matrices, and we exploit this fact to develop a tensor function definition $f(\mathcal{A}) * \mathcal{B}$ that effectively reduces to $f(A)\mathcal{B}$, that is, the action

of a matrix function on a block vector. This fact, along with properties of block circulant matrices, also allows for the transfer of important matrix function properties to our tensor function definition. Although the paradigm we consider is limited in applicability to only third-order tensors, it serves as an important first step in exploring tensor function definitions.

Thanks to the equivalence of our tensor function definition with the $f(A)\mathcal{B}$ problem, we have a number of options for computing $f(\mathcal{A}) * \mathcal{B}$, such as AlMohy,¹⁷ Frommer,¹⁸ Frommer,¹⁹ and Simoncini.²⁰ We focus on adapting the block Krylov subspace methods (KSMs) from Frommer et al.¹⁸, and in light of the so-called “curse of dimensionality,” we also present a computational complexity analysis for this algorithm in the tensor function context. We also propose modifications to the algorithm based on the discrete Fourier transform, which were shown in Kilmer et al.⁹ to increase computational efficiency for the tensor t-product.

This report proceeds as follows. We recapitulate matrix function definitions and properties in Section 1.1. Section 2 restates the tensor t-product framework and poses a definition for the *tensor t-function*, a new definition for a tensor function within this framework. We also present statements and proofs of t-function properties in analogy to the core properties of matrix functions. A possible application for the tensor t-exponential as a generalized communicability measure is discussed in Section 3. In Section 4, we show how block KSMs for matrix functions can be used to compute the tensor t-function and demonstrate the efficacy of these methods for the tensor t-exponential. We make concluding remarks in Section 5.

Before proceeding, we make a brief comment on syntax and disambiguation: the phrase “tensor function” already has an established meaning in physics.^{21,22} The most precise phrase for our object of interest would be “a function of a multidimensional array,” in analogy to “a function of a matrix.” However, since combinations of prepositional phrases can be cumbersome in English, we risk compounding literature searches by resorting to the term “tensor function.”

1.1 | Definitions of matrix functions

Following Higham⁷ and Frommer and Simoncini²³, we concern ourselves with the three main matrix function definitions, based on the Jordan canonical form, Hermite interpolating polynomials, and the Cauchy-Stieltjes integral form. In each case, the validity of the definition boils down to the differentiability of f on the spectrum of A . When f is analytic on the spectrum of A , all the definitions are equivalent, and we can switch between them freely.

Let $A \in \mathbb{C}^{n \times n}$ be a matrix with spectrum $\text{spec}(A) := \{\lambda_j\}_{j=1}^N$, where $N \leq n$ and the λ_j are distinct. An $m \times m$ Jordan block $J_m(\lambda)$ of an eigenvalue λ has the form

$$J_m(\lambda) = \begin{bmatrix} \lambda & 1 & & \\ & \ddots & & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix} \in \mathbb{C}^{m \times m}.$$

Suppose that A has Jordan canonical form

$$A = X J X^{-1} = X^{-1} \text{diag}(J_{m_1}(\lambda_{j_1}), \dots, J_{m_p}(\lambda_{j_\ell})) X, \quad (1)$$

with p blocks of sizes m_i such that $\sum_{i=1}^p m_i = n$, and where the values $\{\lambda_{j_k}\}_{k=1}^\ell \in \text{spec}(A)$. Note that eigenvalues may be repeated in the sequence $\{\lambda_{j_k}\}_{k=1}^\ell$. Let n_j denote the *index* of λ_j or the size of the largest Jordan block associated to λ_j .

A function is *defined on the spectrum of A* if all the following values exist:

$$f^{(k)}(\lambda_j), \quad k = 0, \dots, n_j - 1, j = 1, \dots, N.$$

Definition 1. Suppose $A \in \mathbb{C}^{n \times n}$ has Jordan form (1) and that f is defined on the spectrum of A . Then we define

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

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

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

Note that when A is diagonalizable with $\text{spec}(A) = \{\lambda_j\}_{j=1}^n$ (possibly no longer distinct), Definition 1 reduces to

$$f(A) = X \text{diag}(f(\lambda_1), \dots, f(\lambda_n)) X^{-1}.$$

Matrix powers are well defined, so a scalar polynomial evaluated on a matrix is naturally defined. Given $p(z) = \sum_{k=0}^m z^k c_k$, for some $\{c_k\}_{k=1}^m \subset \mathbb{C}$, we have that $p(A) := \sum_{k=1}^m A^k c_k$. Based on this, we can define nonpolynomial functions of matrices by using again derivatives as we did in Definition 1.

Definition 2. Suppose that f is defined on $\text{spec}(A)$, and let p with $\deg p \leq \sum_{j=1}^N n_j$ be the unique Hermite interpolating polynomial satisfying

$$p^{(k)}(\lambda_j) = f^{(k)}(\lambda_j), \quad \text{for all } k = 0, \dots, n_{j-1}, \quad j = 1, \dots, N.$$

We then define $f(A) := p(A)$.

Theorem 1 (Theorem 1.3 from Higham⁷). *For polynomials p and q and $A \in C^{n \times n}$, $p(A) = q(A)$ if and only if p and q take the same values on the spectrum of A .*

The proof follows by noting that the minimal polynomial of A —that is, the polynomial ψ of least degree such that $\psi(A) = 0$ —divides $p - q$, and consequences thereof.

Crucial for our methods and analysis is the Cauchy-Stieltjes integral definition.

Definition 3. Let $\mathbb{D} \subset \mathbb{C}$ be a region, and suppose that $f : \mathbb{D} \rightarrow \mathbb{C}$ is analytic with integral representation

$$f(z) = \int_{\Gamma} \frac{g(t)}{t - z} dt, \quad z \in \mathbb{D}, \tag{2}$$

with a path $\Gamma \subset \mathbb{C} \setminus \mathbb{D}$ and function $g : \Gamma \rightarrow \mathbb{C}$. Further suppose that the spectrum of A is contained in $\mathbb{C} \setminus \mathbb{D}$. Then we define

$$f(A) := \int_{\Gamma} g(t)(tI - A)^{-1} dt.$$

When f is analytic, $g = \frac{1}{2\pi i} f$, and Γ is a contour enclosing the spectrum of A , then Definition 2 reduces to the usual Cauchy integral definition.

Various matrix function properties will prove useful throughout our analysis. Their proofs follow by examining the polynomial and Jordan form definitions of matrix functions.

Theorem 2 (Theorem 1.13 in Higham⁷). *Let $A \in C^{n \times n}$ and let f be defined on the spectrum of A . Then*

- (i) $f(A)A = Af(A)$;
- (ii) $f(A^T) = f(A)^T$;
- (iii) $f(XAX^{-1}) = Xf(A)X^{-1}$; and
- (iv) $f(\lambda) \in \text{spec}(f(A))$ for all $\lambda \in \text{spec}(A)$.

2 | A DEFINITION FOR TENSOR FUNCTIONS

We direct the reader now to Figure 1 for different “views” of a third-order tensor, which will be useful in visualizing the forthcoming concepts. We also make use of some notions from block matrices. Define the standard block unit vectors

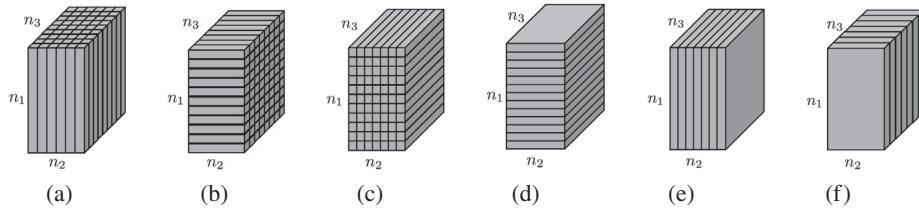


FIGURE 1 Different views of a third-order tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$. (a) column fibers: $\mathcal{A}(:, j, k)$; (b) row fibers: $\mathcal{A}(i, :, k)$; (c) tube fibers: $\mathcal{A}(i, j, :)$; (d) horizontal slices: $\mathcal{A}(i, :, :)$; (e) lateral slices: $\mathcal{A}(:, j, :)$; and (f) frontal slices: $\mathcal{A}(:, :, k)$

$\mathbf{E}_k^{np \times n} := \hat{\mathbf{E}}_k^p \otimes I_{n \times n}$, where $\hat{\mathbf{E}}_k^p \in \mathbb{C}^p$ is the vector of all zeros except for the k th entry, and $I_{n \times n}$ is the identity in $\mathbb{C}^{n \times n}$. When the dimensions are clear from context, we drop the superscripts. See Equation (3) for various ways of expressing $\mathbf{E}_1^{np \times n}$.

$$\mathbf{E}_1^{np \times n} = \begin{bmatrix} I_{n \times n} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \otimes I_{n \times n} = \text{unfold}(\mathcal{I}_{n \times n \times p}), \quad (3)$$

where `unfold` is defined shortly.

In Kilmer et al.⁹, Braman¹⁵, and Kilmer and Martin 16, a paradigm is proposed for multiplying third-order tensors, based on viewing a tensor as a stack of frontal slices (as in Figure 1f). We consider a tensor \mathcal{A} of size $m \times n \times p$ and \mathcal{B} of size $n \times s \times p$ and denote their frontal faces, respectively, as $A^{(k)}$ and $B^{(k)}$, $k = 1, \dots, p$. We also define the operations `bcirc`, `unfold`, `fold`, as

$$\text{bcirc}(\mathcal{A}) := \begin{bmatrix} A^{(1)} & A^{(p)} & A^{(p-1)} & \dots & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(p)} & \dots & A^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{(p)} & A^{(p-1)} & \ddots & A^{(2)} & A^{(1)} \end{bmatrix}, \quad (4)$$

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

The *t*-product of two tensors \mathcal{A} and \mathcal{B} is then given as

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

Note that the operators `fold`, `unfold`, and `bcirc` are linear.

The notion of transposition is defined facewise, that is, \mathcal{A}^* is the $n \times m \times p$ tensor obtained by taking the conjugate transpose of each frontal slice of \mathcal{A} and then reversing the order of the second through p th transposed slices.

For tensors with $n \times n$ square faces, there is a tensor identity $\mathcal{I}_{n \times n \times p} \in \mathbb{C}^{n \times n \times p}$, whose first frontal slice is the $n \times n$ identity matrix and whose remaining frontal slices are all the zero matrix. With $\mathcal{I}_{n \times n \times p}$, one can then define the notion of an inverse with respect to the t-product. Namely, $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{n \times n \times p}$ are inverses of each other if $\mathcal{A} * \mathcal{B} = \mathcal{I}_{n \times n \times p}$ and $\mathcal{B} * \mathcal{A} = \mathcal{I}_{n \times n \times p}$. The t-product formalism further gives rise to its own notion of polynomials, with powers of tensors defined as $\mathcal{A}^j := \underbrace{\mathcal{A} * \dots * \mathcal{A}}_{j \text{ times}}$.

Assuming that $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ has diagonalizable faces, we can also define a tensor eigendecomposition. That is, we have that $A^{(k)} = X^{(k)} D^{(k)} (X^{(k)})^{-1}$, for all $k = 1, \dots, p$, and define \mathcal{X} and \mathcal{D} to be the tensors whose faces are $X^{(k)}$ and $D^{(k)}$, respectively. Then

$$\mathcal{A} = \mathcal{X} * \mathcal{D} * \mathcal{X}^{-1} \quad \text{and} \quad \mathcal{A} * \vec{\mathcal{X}}_i = \vec{\mathcal{X}}_i * \mathbf{d}_i, \quad (5)$$

where $\vec{\mathcal{X}}_i$ are the $n \times 1 \times p$ lateral slices of \mathcal{X} (Figure 1e) and \mathbf{d}_j are the $1 \times 1 \times p$ tubal fibers of \mathcal{D} (Figure 1). We say that \mathcal{D} is *f-diagonal*, that is, that each of its frontal faces is a diagonal matrix.

The eigenvalue decomposition (5) is not unique. See Gleich et al. 24 for an alternative circulant-based interpretation of third-order tensors, as well as a deeper exploration of a unique canonical eigendecomposition for tensors.

2.1 | The tensor t-exponential

As motivation, we consider the solution to a multidimensional ordinary differential equation. Suppose that \mathcal{A} has square frontal faces, that is, that $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ and let $\mathcal{B} : [0, \infty) \rightarrow \mathbb{C}^{n \times s \times p}$ be an unknown function with $\mathcal{B}(0)$ given. With $\frac{d}{dt}$ acting elementwise, we consider the differential equation

$$\frac{d\mathcal{B}}{dt}(t) = \mathcal{A} * \mathcal{B}(t). \quad (6)$$

Unfolding both sides leads to

$$\frac{d}{dt} \begin{bmatrix} B^{(1)}(t) \\ \vdots \\ B^{(n)}(t) \end{bmatrix} = \text{bcirc}(\mathcal{A}) \begin{bmatrix} B^{(1)}(t) \\ \vdots \\ B^{(n)}(t) \end{bmatrix},$$

whose solution can be expressed in terms of the matrix exponential as

$$\begin{bmatrix} B^{(1)}(t) \\ \vdots \\ B^{(n)}(t) \end{bmatrix} = \exp(\text{bcirc}(\mathcal{A})t) \begin{bmatrix} B^{(1)}(0) \\ \vdots \\ B^{(n)}(0) \end{bmatrix}.$$

Folding both sides again leads to the *tensor t-exponential*,

$$\mathcal{B}(t) = \text{fold}(\exp(\mathcal{A}t)\text{unfold}(\mathcal{B}(0))) =: \exp(\mathcal{A}t) * \mathcal{B}(0). \quad (7)$$

2.2 | The tensor t-function

Using the tensor t-exponential as inspiration, we can define a more general notion for the scalar function f of a tensor $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ multiplied by a tensor $\mathcal{B} \in \mathbb{C}^{n \times s \times p}$ as

$$f(\mathcal{A}) * \mathcal{B} := \text{fold}(f(\text{bcirc}(\mathcal{A})) \cdot \text{unfold}(\mathcal{B})), \quad (8)$$

which we call the *tensor t-function*. Note that $f(\text{bcirc}(\mathcal{A})) \cdot \text{unfold}(\mathcal{B})$ is merely a matrix function times a block vector. If $\mathcal{B} = \mathcal{I}_{n \times n \times p}$, then by Equation (1) the definition for $f(\mathcal{A})$ reduces to

$$f(\mathcal{A}) := \text{fold}(f(\text{bcirc}(\mathcal{A})) \mathbf{E}_1^{np \times n}). \quad (9)$$

A natural question is whether the definition (8) behaves “as expected” in common scenarios. To answer this question, we require some results on block circulant matrices and the tensor t-product.

Theorem 3 (Theorem 5.6.5 in Davis²⁵). *Suppose $A, B \in \mathbb{C}^{np \times np}$ are block circulant matrices with $n \times n$ blocks. Let $\{\alpha_j\}_{j=1}^k$ be scalars. Then $A^T, A^*, \alpha_1 A + \alpha_2 B, AB, q(A) = \sum_{j=1}^k \alpha_j A^j$, and A^{-1} (when it exists) are also block circulant.*

Remark 1. From Equation (4), we can see that any block circulant matrix $C \in \mathbb{C}^{np \times np}$ can be represented by its first column $C\mathbf{E}_1^{np \times n}$. Let $\mathcal{C} \in \mathbb{C}^{n \times n \times p}$ be a tensor whose frontal faces are the block entries of $C\mathbf{E}_1^{np \times n}$. Then $\mathcal{C} = \text{fold}(C\mathbf{E}_1^{np \times n})$.

Lemma 1. *Let $\mathcal{A} \in \mathbb{C}^{m \times n \times p}$ and $\mathcal{B} \in \mathbb{C}^{n \times s \times p}$. Then*

$$(i) \text{unfold}(\mathcal{A}) = \text{bcirc}(\mathcal{A}) \mathbf{E}_1^{np \times n};$$

- (ii) $\text{bcirc}(\text{fold}(\text{bcirc}(\mathcal{A})\mathbf{E}_1^{np \times n})) = \text{bcirc}(\mathcal{A})$;
- (iii) $\text{bcirc}(\mathcal{A} * \mathcal{B}) = \text{bcirc}(\mathcal{A})\text{bcirc}(\mathcal{B})$;
- (iv) $\text{bcirc}(\mathcal{A}^j) = \text{bcirc}(\mathcal{A}^j)$, for all $j = 0, 1, \dots$; and
- (v) $(\mathcal{A} * \mathcal{B})^* = \mathcal{B}^* * \mathcal{A}^*$.

Proof. We drop the superscripts on $\mathbf{E}_1^{np \times n}$ for ease of presentation. Parts (i) and (ii) follow from Remark 7. To prove part (iii), we note by part (i) that

$$\begin{aligned}\text{bcirc}(\mathcal{A} * \mathcal{B}) &= \text{bcirc}(\text{fold}(\text{bcirc}(\mathcal{A})\text{unfold}(\mathcal{B}))) \\ &= \text{bcirc}(\text{fold}(\text{bcirc}(\mathcal{A})\text{bcirc}(\mathcal{B})\mathbf{E}_1)).\end{aligned}$$

Note that $\text{bcirc}(\mathcal{A})\text{bcirc}(\mathcal{B})$ is a block circulant matrix by Theorem 3. Then by part (iii),

$$\text{bcirc}(\text{fold}(\text{bcirc}(\mathcal{A})\text{bcirc}(\mathcal{B})\mathbf{E}_1)) = \text{bcirc}(\mathcal{A})\text{bcirc}(\mathcal{B}).$$

Part (iv) follows by induction on part (iii). Part (v) is the same as Kilmer and Martin¹⁶, Lemma 3.16. ■

Let \mathcal{D} be an $n \times n \times p$ *f-diagonal* tensor, that is, a tensor whose $n \times n$ frontal slices are diagonal matrices. Alternatively, one can think of such a tensor as an $n \times n$ matrix nonzero tube fibers on the diagonal, and zero tube fibers everywhere else (Figure 1c). The following theorem summarizes the relationship between the block circulant of \mathcal{D} and those of its tube fibers.

Theorem 4. Let $\mathcal{D} \in \mathbb{C}^{n \times n \times p}$ be f-diagonal, and let $\{\mathbf{d}_i\}_{i=1}^n \subset \mathbb{C}^{1 \times 1 \times p}$ denote its diagonal tube fibers. Then the spectrum of $\text{bcirc}(\mathcal{D})$ is identical to the union of the spectra of $\text{bcirc}(\mathbf{d}_i)$, $i = 1, \dots, n$.

Proof. We begin by deriving an expression for $\text{bcirc}(\mathcal{D})$ in terms of the $p \times p$ circulant matrices $\text{bcirc}(\mathbf{d}_i)$. Denote each slice as $D^{(k)}$, $k = 1, \dots, p$, with diagonal entries denoted as $d_i^{(k)}$, for $i = 1, \dots, n$, that is,

$$D^{(k)} = \begin{bmatrix} d_1^{(k)} & & & \\ & \ddots & & \\ & & d_n^{(k)} & \end{bmatrix}.$$

Then we can express $\text{bcirc}(\mathcal{D})$ as follows:

$$\begin{aligned}\text{bcirc}(\mathcal{D}) &= \begin{bmatrix} D^{(1)} & D^{(p)} & \cdots & D^{(2)} \\ D^{(2)} & D^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & D^{(p)} \\ D^{(p)} & \cdots & D^{(2)} & D^{(1)} \end{bmatrix} \\ &= \begin{bmatrix} d_1^{(1)} & & & & & & & \\ & d_1^{(p)} & & & & & & \\ & & \ddots & & & & & \\ & & & d_n^{(1)} & & & & \\ & & & & d_1^{(1)} & & & \\ & & & & & \ddots & & \\ & & & & & & \ddots & \\ & & & & & & & d_n^{(2)} \\ d_1^{(2)} & & & & & & & \\ & \ddots & & & & & & \\ & & d_n^{(2)} & & & & & \\ & & & d_1^{(1)} & & & & \\ & & & & \ddots & & & \\ & & & & & d_1^{(p)} & & \\ & & & & & & \ddots & \\ & & & & & & & d_n^{(p)} \\ d_1^{(p)} & & & & & & & \\ & \ddots & & & & & & \\ & & d_n^{(p)} & & & & & \\ & & & d_1^{(2)} & & & & \\ & & & & \ddots & & & \\ & & & & & d_1^{(1)} & & \\ & & & & & & \ddots & \\ & & & & & & & d_n^{(2)} \\ & & & & & & & \\ & & & & & & & d_n^{(1)} \end{bmatrix}.\end{aligned}$$

Collecting the highlighted elements, note that the block circulant of the first tube fiber is given as

$$\text{bcirc}(\mathbf{d}_1) = \begin{bmatrix} d_1^{(1)} & d_1^{(p)} & \dots & d_1^{(2)} \\ d_1^{(2)} & d_1^{(1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & d_1^{(p)} \\ d_1^{(p)} & \dots & d_1^{(2)} & d_1^{(1)} \end{bmatrix}$$

Defining

$$\hat{I}_1 := \begin{bmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \in \mathbb{C}^{n \times n},$$

it holds that

$$\begin{aligned}
& \text{bcirc}(\mathbf{d}_1) \otimes \widehat{I}_1 \\
&= \begin{bmatrix} d_1^{(1)} & & & & d_1^{(2)} \\ & \ddots & & & \ddots & & \\ & & d_1^{(p)} & & & & \\ & & & 0 & & 0 & \\ & & & & \ddots & & \\ & & & & & 0 & \\ d_1^{(2)} & & & d_1^{(1)} & & & \\ & \ddots & & & \ddots & & \\ & & 0 & & 0 & & \\ & & & \ddots & & & \\ = & & & & & d_1^{(p)} & \\ & \vdots & & & & & \\ & & & \ddots & & & \\ & & & & d_1^{(2)} & & \\ d_1^{(p)} & & & & d_1^{(1)} & & \\ & \ddots & & & & & \\ & & 0 & & & 0 & \\ & & & \ddots & & & \\ & & & & 0 & & \\ & & & & & 0 & \\ \end{bmatrix}
\end{aligned}$$

Noting the same pattern for each $i = 1, \dots, n$, it is not hard to see that

$$\text{bcirc}(\mathcal{D}) = \sum_{i=1}^n \text{bcirc}(\mathbf{d}_i) \otimes \hat{I}_i, \quad (10)$$

where $\hat{I}_i \in \mathbb{C}^{n \times n}$ is zero everywhere except for the i th entry, which is one.

It is known that a circulant matrix is unitarily diagonalizable by the discrete Fourier transform (DFT); see Davis 25, Section 3.2. Then for a $p \times p$ circulant matrix C , and with F_p denoting the $p \times p$ DFT, $F_p^* C F_p = \Lambda$, where $\Lambda \in \mathbb{C}^{p \times p}$ is diagonal. Since each $\text{bcirc}(\mathbf{d}_i)$ is a $p \times p$ circulant matrix, there exists for each $i = 1, \dots, n$ a diagonal $\Lambda_i \in \mathbb{C}^{p \times p}$ such that

$$F_p \text{bcirc}(\mathbf{d}_i) F_p = \Lambda_i. \quad (11)$$

We also have the following useful property of the Kronecker product for matrices A , B , C , and D such that the products AC and BD exist; see Horn and Johnson²⁶, Lemma 4.2.10:

$$(A \otimes B)(C \otimes D) = (AB) \times (CD). \quad (12)$$

Consequently,

$$\begin{aligned}
(F_p \otimes I_{n \times n}) \text{bcirc}(\mathcal{D})(F_p^* \otimes I_{n \times n}) &= (F_p \otimes I_{n \times n}) \left(\sum_{i=1}^n \text{bcirc}(\mathbf{d}_i) \otimes \hat{I}_i \right) (F_p^* \otimes I_{n \times n}) \\
&= \sum_{i=1}^n (F_p \otimes I_{n \times n}) \text{bcirc}(\mathbf{d}_i) \otimes \hat{I}_i (F_p^* \otimes I_{n \times n}) \\
&= \sum_{i=1}^n (F_p \text{bcirc}(\mathbf{d}_i) F_p^*) \otimes (I_{n \times n} \hat{I}_i I_{n \times n}), \text{ by Equation (12)} \\
&= \sum_{i=1}^n \Lambda_i \otimes \hat{I}_i, \text{ by Equation (11).}
\end{aligned}$$

Noting that $F_p \otimes I_{n \times n}$ is unitary and that the matrix $\Lambda := \sum_{i=1}^n \Lambda_i \otimes \hat{I}_i$ is a diagonal matrix whose entries are precisely the diagonal entries of all the Λ_i concludes the proof. ■

Corollary 1. Let $\mathcal{D} \in \mathbb{C}^{n \times n \times p}$ be f -diagonal, and let $\{\mathbf{d}_i\}_{i=1}^n \subset \mathbb{C}^{1 \times 1 \times p}$ denote its diagonal tube fibers. Then a function f being defined on the spectrum of $\text{bcirc}(\mathcal{D})$ is equivalent to f being defined on the union of the spectra of $\text{bcirc}(\mathbf{d}_i)$, $i = 1, \dots, n$.

An immediate consequence of Theorem 4 is that a function f being defined on the spectrum of $\text{bcirc}(\mathcal{D})$ is equivalent to f being defined on the spectra of $\text{bcirc}(\mathbf{d}_i)$, $i = 1, \dots, n$. The interpolating polynomials for $f(\text{bcirc}(\mathcal{D}))$ and $f(\text{bcirc}(\mathbf{d}_i))$, $i = 1, \dots, n$, are also related.

The following theorem ensures that definition (8) is well defined when f is a polynomial, when \mathcal{A} and \mathcal{B} are second-order tensors (i.e., matrices), and when f is the inverse function.

Theorem 5. Let $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ and $\mathcal{B} \in \mathbb{C}^{n \times s \times p}$.

(i) If $f \equiv q$, where q is a polynomial, then the tensor t -function definition (8) matches the polynomial notion in the t -product formalism, that is,

$$\text{fold}(q(\text{bcirc}(\mathcal{A})) \cdot \text{unfold}(\mathcal{B})) = \text{fold}(\text{bcirc}(q(\mathcal{A})) \cdot \text{unfold}(\mathcal{B})).$$

(ii) Let q be the scalar polynomial guaranteed by Definition 2 so that $f(\text{bcirc}(\mathcal{A})) = q(\text{bcirc}(\mathcal{A}))$. Then $f(\mathcal{A}) * \mathcal{B} = q(\mathcal{A}) * \mathcal{B}$.

(iii) If \mathcal{A} is a matrix and \mathcal{B} a block vector (i.e., if $p = 1$), then $f(\mathcal{A}) * \mathcal{B}$ reduces to the usual matrix function definition.

(iv) If $f(z) = z^{-1}$, then $f(\mathcal{A}) * \mathcal{A} = \mathcal{A} * f(\mathcal{A}) = I_{n \times n \times p}$.

Proof. For part (i), let $q(z) = \sum_{j=1}^m c_j z^j$. Then by Lemma 1(iv) and the linearity of fold , we have that

$$\begin{aligned}
\text{fold}(q(\text{bcirc}(\mathcal{A})) \cdot \text{unfold}(\mathcal{B})) &= \text{fold} \left(\sum_{j=1}^m c_j \text{bcirc}(\mathcal{A})^j \cdot \text{unfold}(\mathcal{B}) \right) \\
&= \sum_{j=1}^m c_j \text{fold}(\text{bcirc}(\mathcal{A}^j) \cdot \text{unfold}(\mathcal{B})) \\
&= \sum_{j=1}^m c_j \text{bcirc}(\mathcal{A}^j) * \mathcal{B} \\
&= \text{fold}(\text{bcirc}(q(\mathcal{A})) \cdot \text{unfold}(\mathcal{B})).
\end{aligned}$$

Part (ii) is a special case of part (i). As for part (iii), since $p = 1$, we have that $\text{fold}(\mathcal{A}) = \text{bcirc}(\mathcal{A}) = \mathcal{A} = \text{unfold}(\mathcal{A})$, and similarly for \mathcal{B} . Then the definition of $f(\mathcal{A}) * \mathcal{B}$ reduces immediately to the matrix function case. Part (iv) follows by carefully unwrapping the definition of $f(\mathcal{A})$:

$$\begin{aligned}
f(\mathcal{A}) * \mathcal{A} &= \text{fold}(\text{bcirc}(\mathcal{A})^{-1} \text{unfold}(\mathcal{A})) \\
&= \text{fold}(\text{bcirc}(\mathcal{A})^{-1} \text{bcirc}(\mathcal{A}) E_1^{np \times n}), \text{ by Lemma 1(i)} \\
&= \text{fold}(E_1^{np \times n}) = I_{n \times n \times p}.
\end{aligned}$$

Likewise with the other product:

$$\begin{aligned}\mathcal{A} * f(\mathcal{A}) &= \text{fold}(\text{bcirc}(\mathcal{A}) \text{unfold}(\text{fold}(\text{bcirc}(\mathcal{A})^{-1} \text{unfold}(\mathcal{I}_{n \times n \times p})))) \\ &= \text{fold}(\text{bcirc}(\mathcal{A}) \text{bcirc}(\mathcal{A})^{-1} \mathbf{E}_1^{np \times n}) \\ &= \text{fold}(\mathbf{E}_1^{np \times n}) = \mathcal{I}_{n \times n \times p}.\end{aligned}$$

The definition (8) possesses generalized versions of many of the core properties of matrix functions.

Theorem 6. Let $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$, and let $f : \mathbb{C} \rightarrow \mathbb{C}$ be defined on a region in the complex plane containing the spectrum of $\text{bcirc}(\mathcal{A})$. For part (iv), assume that \mathcal{A} has an eigendecomposition as in Equation (5), with $\mathcal{A} * \vec{\mathcal{X}}_i = \mathcal{D} * \vec{\mathcal{X}}_i = \vec{\mathcal{X}}_i * \mathbf{d}_i$, $i = 1, \dots, n$. Then it holds that

- (i) $f(\mathcal{A})$ commutes with \mathcal{A} ;
- (ii) $f(\mathcal{A}^*) = f(\mathcal{A})^*$;
- (iii) $f(\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1}) = \mathcal{X}f(\mathcal{A})\mathcal{X}^{-1}$; and
- (iv) $f(\mathcal{D}) * \vec{\mathcal{X}}_i = \vec{\mathcal{X}}_i * f(\mathbf{d}_i)$, for all $i = 1, \dots, n$.

Proof. For parts (i) to (iii), it suffices by Theorem 5(ii) to show that the statements hold for $f(z) = \sum_{j=1}^m c_j z^j$. Part (i) then follows immediately. To prove part (ii), we need only show that $(\mathcal{A}^j)^* = (\mathcal{A}^*)^j$ for all $j = 0, 1, \dots$, which follows by induction from Lemma 1(v). Part (iii) also follows inductively. The base cases $j = 0, 1$ clearly hold. Assume for some $j = k$, $(\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1})^k = \mathcal{X}(\mathcal{A})^k \mathcal{X}^{-1}$, and then note that

$$\begin{aligned}(\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1})^{k+1} &= (\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1})^k * (\mathcal{X} * \mathcal{A} * \mathcal{X}^{-1}) \\ &= \mathcal{X} * (\mathcal{A})^k * \mathcal{X}^{-1} * \mathcal{X} * \mathcal{A} * \mathcal{X}^{-1} = \mathcal{X} * (\mathcal{A})^{k+1} * \mathcal{X}^{-1}.\end{aligned}$$

For part (iv), we fix $i \in \{1, \dots, n\}$. By Corollary 1, f being defined on $\text{spec}(\text{bcirc}(\mathcal{D}))$ implies that it is also defined on $\text{spec}(\text{bcirc}(\mathbf{d}_i))$. Let q and q_i be the polynomials guaranteed by Theorem 2 such that $f(\text{bcirc}(\mathcal{D})) = q(\text{bcirc}(\mathcal{D}))$ and $f(\text{bcirc}(\mathbf{d}_i)) = q_i(\text{bcirc}(\mathbf{d}_i))$. By Theorem 4, $\text{spec}(\text{bcirc}(\mathbf{d}_i)) \subset \text{spec}(\text{bcirc}(\mathcal{D}))$, so by Theorem 4, it follows that $q_i(\text{bcirc}(\mathbf{d}_i)) = q(\text{bcirc}(\mathbf{d}_i))$. Then it suffices to prove part (iv) for \mathcal{D}^j , $j = 1, 0, \dots$. The cases $j = 0, 1$ clearly hold, and we assume the statement holds for some $j = k \geq 1$. Then

$$\mathcal{D}^{k+1} * \vec{\mathcal{X}}_i = \mathcal{D} * (\mathcal{D}^k * \vec{\mathcal{X}}_i) = \mathcal{D} * \vec{\mathcal{X}}_i * \mathbf{d}_i^k = \vec{\mathcal{X}}_i * \mathbf{d}_i^{k+1}.$$

Remark 2. When \mathcal{A} has an eigendecomposition $\mathcal{X} * \mathcal{D} * \mathcal{X}^{-1}$ as in Equation (5), then by Theorems 4 and 6, an equivalent definition for $f(\mathcal{A})$ is given as

$$f(\mathcal{A}) = \mathcal{X} * \begin{bmatrix} f(\mathbf{d}_1) & & \\ & \ddots & \\ & & f(\mathbf{d}_n) \end{bmatrix} * \mathcal{X}^{-1},$$

where the inner matrix should be regarded three-dimensionally, with its elements being tube fibers (cf. Figure 1c).

3 | CENTRALITY AND COMMUNICABILITY OF A THIRD-ORDER NETWORK

We use the term *network* to denote an undirected, unweighted graph with n nodes. The graph, and by extension, the network, can be represented by its *adjacency matrix* $A \in \mathbb{R}^{n \times n}$. The ij th entry of A is 1 if nodes i and j are connected, and 0 otherwise. As a rule, a node is not connected to itself, so $A_{ii} = 0$. The centrality of the i th node is defined as $\exp(A)_{ii}$, while the communicability between nodes i and j is defined as $\exp(A)_{ij}$.

These notions can be extended to higher order situations. Suppose we are concerned instead about triplets, instead of pairs, of nodes. Then it is possible to construct an adjacency tensor \mathcal{A} , where a 1 at entry \mathcal{A}_{ijk} indicates that distinct nodes i, j , and k are connected and 0 otherwise. Information will, however, be lost if only the adjacency tensor is considered, since pairwise connectivity is stored in the adjacency matrix. Multilayer networks, such as those describing a city's bus, metro, and tram systems, constitute a more natural application, and several notions of centrality are explored in Domenico et al.²⁷. Alternatively, it is not hard to imagine a time-dependent network stored as a tensor, where each frontal face corresponds to a sampling of the network at discrete times.²⁸ In any of these situations, we could compute the communicability of a triple as $\exp(\mathcal{A})_{ijk}$, where $\exp(\mathcal{A})$ is our tensor t-exponential. Centrality for a node i would thus be defined as $\exp(\mathcal{A})_{iii}$.

4 | COMPUTING THE TENSOR T-FUNCTION

For the tensor t-function to be viable in practice, we need efficient methods for approximating $f(\mathcal{A}) * \mathcal{B}$ numerically. The t-eigendecomposition and t-Krylov methods of Kilmer et al.⁹ are potential options, but a full eigendecomposition may be expensive to compute for large tensors, and crafting t-Krylov methods for tensor functions may not be necessary, given the equivalence between $f(\mathcal{A}) * \mathcal{B}$ and the $f(\mathcal{A})\mathcal{B}$ problem. We therefore adapt the block KSMs of,¹⁸ designed specifically for such problems, and further optimize the computations by taking advantage of the block circulant structure of $\text{bcirc}(\mathcal{A})$.

4.1 | Block KSMs for matrix functions

We recount here the block Krylov subspace framework from Frommer et al.¹⁸, given a scalar function f defined on an $np \times np$ matrix A and a block vector $\mathbf{B} \in \mathbb{C}^{np \times n}$. This framework allows us to treat different inner products and norms simultaneously, some of which are computationally more advantageous than others; see Section 4.2.

Let $\mathbb{S} \subset \mathbb{C}^{n \times n}$ be a *-subalgebra with identity.

Definition 4. A mapping $\langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}$ from $\mathbb{C}^{np \times n} \times \mathbb{C}^{np \times n}$ to \mathbb{S} is called a *block inner product onto \mathbb{S}* if it satisfies the following conditions for all $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{C}^{np \times n}$ and $C \in \mathbb{S}$:

- (i) \mathbb{S} -linearity: $\langle\langle \mathbf{X} + \mathbf{Y}, \mathbf{Z}C \rangle\rangle_{\mathbb{S}} = \langle\langle \mathbf{X}, \mathbf{Z} \rangle\rangle_{\mathbb{S}}C + \langle\langle \mathbf{Y}, \mathbf{Z} \rangle\rangle_{\mathbb{S}}C$;
- (ii) symmetry: $\langle\langle \mathbf{X}, \mathbf{Y} \rangle\rangle_{\mathbb{S}} = \langle\langle \mathbf{Y}, \mathbf{X} \rangle\rangle_{\mathbb{S}}^*$;
- (iii) definiteness: $\langle\langle \mathbf{X}, \mathbf{X} \rangle\rangle_{\mathbb{S}}$ is positive definite if \mathbf{X} has full rank, and $\langle\langle \mathbf{X}, \mathbf{X} \rangle\rangle_{\mathbb{S}} = 0$ if and only if $\mathbf{X} = 0$.

Definition 5. A mapping N which maps all $\mathbf{X} \in \mathbb{C}^{np \times n}$ with full rank on a matrix $N(\mathbf{X}) \in \mathbb{S}$ is called a *scaling quotient* if for all such \mathbf{X} , there exists $\mathbf{Y} \in \mathbb{C}^{np \times n}$ such that $\mathbf{X} = \mathbf{Y}N(\mathbf{X})$ and $\langle\langle \mathbf{Y}, \mathbf{Y} \rangle\rangle_{\mathbb{S}} = I_s$.

Definition 6. Let $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{np \times n}$.

- (i) \mathbf{X}, \mathbf{Y} are $\langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}$ -orthogonal, if $\langle\langle \mathbf{X}, \mathbf{Y} \rangle\rangle_{\mathbb{S}} = 0$.
- (ii) \mathbf{X} is $\langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}$ -normalized if $N(\mathbf{X}) = I$.
- (iii) $\{\mathbf{X}_1, \dots, \mathbf{X}_m\} \subset \mathbb{C}^{np \times n}$ is $\langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}$ -orthonormal if $\langle\langle \mathbf{X}_i, \mathbf{X}_j \rangle\rangle_{\mathbb{S}} = \delta_{ij}I$, where δ_{ij} is the Kronecker delta.

We say that a set of vectors $\{\mathbf{X}_j\}_{j=1}^m \subset \mathbb{C}^{np \times n}$ \mathbb{S} -spans a space $\mathcal{K} \subset \mathbb{C}^{np \times n}$ and write $\mathcal{K} = \text{span}^{\mathbb{S}}\{\mathbf{X}_j\}_{j=1}^m$, where

$$\text{span}^{\mathbb{S}}\{\mathbf{X}_j\}_{j=1}^m := \left\{ \sum_{j=1}^m \mathbf{X}_j \Gamma_j : \Gamma_j \in \mathbb{S} \text{ for all } j = 1, \dots, m \right\} \subset \mathbb{C}^{np \times n}.$$

The set $\{\mathbf{X}_j\}_{j=1}^m$ constitutes an $\langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}$ -orthonormal basis for \mathcal{K} if m is the dimension of \mathcal{K} , $\mathcal{K} = \text{span}^{\mathbb{S}}\{\mathbf{X}_j\}_{j=1}^m$, and $\{\mathbf{X}_j\}_{j=1}^m$ are orthonormal.

We define the m th block Krylov subspace for $A \in \mathbb{C}^{np \times np}$ and $\mathbf{B} \in \mathbb{C}^{np \times n}$ as

$$\mathcal{K}_m^{\mathbb{S}}(A, \mathbf{B}) = \text{span}^{\mathbb{S}}\{\mathbf{B}, A\mathbf{B}, \dots, A^{m-1}\mathbf{B}\}.$$

There exist many choices for \mathbb{S} , $\langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}$, and N ; see Frommer et al.¹⁸ or Elbouyahyaoui²⁹ for further details. We consider only the *classical* and *global* paradigms because of the potential to speed up convergence and the low computational effort per iteration, respectively:

	Classical	Global
\mathbb{S}	$I_n \otimes \mathbb{C}$	$\mathbb{C}I_n$
$\langle\langle \mathbf{X}, \mathbf{Y} \rangle\rangle_{\mathbb{S}}$	$\text{diag}(\mathbf{X}^* \mathbf{Y})$	$\frac{1}{n} \text{trace} (\mathbf{X}^* \mathbf{Y}) I_n$
$N(\mathbf{X})$	R , where $\mathbf{X} = \mathbf{Q}R$	$\frac{1}{\sqrt{n}} \ \mathbf{X}\ _{\text{F}} I_n$

Algorithm 1 is the generalized block Arnoldi procedure. We assume that Algorithm 1 runs to completion without breaking down, that is, that we obtain

- (i) a $\langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}$ -orthonormal basis $\{\mathbf{V}_k\}_{k=1}^{m+1} \subset \mathbb{C}^{np \times n}$, such that each \mathbf{V}_k has full rank and $\mathcal{K}_m^{\mathbb{S}}(A, \mathbf{B}) = \text{span}^{\mathbb{S}}\{\mathbf{V}_k\}_{k=1}^m$, and
- (ii) a block upper Hessenberg matrix $\mathcal{H}_m \in \mathbb{S}^{m \times m}$ and $H_{m+1,m} \in \mathbb{S}$,

all satisfying the *block Arnoldi relation*

$$A\mathcal{V}_m = \mathcal{V}_m \mathcal{H}_m + \mathbf{V}_{m+1} H_{m+1,m} \mathbf{E}_m^*, \quad (13)$$

where $\mathcal{V}_m = [\mathbf{V}_1 | \dots | \mathbf{V}_m] \in \mathbb{C}^{np \times nm}$, and $(\mathcal{H}_m)_{ij} = H_{ij}$. Note that \mathcal{H}_m has dimension $mn \times mn$ so long as $m \ll p$, \mathcal{H}_m will be significantly smaller than A . Otherwise, it will be necessary to partition the right-hand side \mathbf{B} and compute the action of $f(A)$ on each partition separately. Furthermore, in the global paradigm, \mathcal{H}_m has a Kronecker structure $H \otimes I_n$, where $H \in \mathbb{C}^{m \times m}$, so the storage of \mathcal{H}_m can be reduced.

Algorithm 1. Block Arnoldi

- 1: **Given:** $A, \mathbf{B}, \mathbb{S}, \langle\langle \cdot, \cdot \rangle\rangle_{\mathbb{S}}, N, m$
 - 2: Compute $B = N(\mathbf{B})$ and $\mathbf{V}_1 = \mathbf{B}B^{-1}$
 - 3: **for** $k = 1, \dots, m$ **do**
 - 4: Compute $\mathbf{W} = A\mathbf{V}_k$
 - 5: **for** $j = 1, \dots, k$ **do**
 - 6: $H_{j,k} = \langle\langle \mathbf{V}_j, \mathbf{W} \rangle\rangle_{\mathbb{S}}$
 - 7: $\mathbf{W} = \mathbf{W} - \mathbf{V}_j H_{j,k}$
 - 8: **end for**
 - 9: Compute $H_{k+1,k} = N(\mathbf{W})$ and $\mathbf{V}_{k+1} = \mathbf{W}H_{k+1,k}^{-1}$
 - 10: **end for**
 - 11: **return** $B, \mathcal{V}_m = [\mathbf{V}_1 | \dots | \mathbf{V}_m], \mathcal{H}_m = (H_{j,k})_{j,k=1}^m, \mathbf{V}_{m+1}$, and $H_{m+1,m}$
-

The article¹⁸ also establishes theory for a block full orthogonalization method for functions of matrices (B(FOM)²). The B(FOM)² approximation is defined as

$$\mathbf{F}_m := \mathcal{V}_m f(\mathcal{H}_m) \mathbf{E}_1 B, \quad (14)$$

which indeed reduces to a block FOM approximation when $f(z) = z^{-1}$.³⁰

With the end application being tensors, restarts will be necessary to mitigate memory limitations imposed by handling higher order data. Restarts for B(FOM)² are developed in detail in Frommer et al.¹⁸ for functions with Cauchy-Stieltjes representations, including the matrix exponential; we present here a high-level summary of the procedure as Algorithm 2. Restarts are performed by approximating an error function via adaptive quadrature rules; this step is represented by $\Delta_m^{(k)}$ in line 6.

Algorithm 2. B(FOM)²(m): block full orthogonalization method for functions of matrices with restarts

- 1: Given $f, A, \mathbf{B}, \mathbb{S}, \langle \cdot, \cdot \rangle_{\mathbb{S}}, N, m, t, \text{tol}$
 - 2: Run Algorithm 4.1 with inputs $A, \mathbf{B}, \mathbb{S}, \langle \cdot, \cdot \rangle_{\mathbb{S}}, N$, and m and store $\mathcal{V}_{m+1}^1, \mathcal{H}_m^1$, and B^1
 - 3: Compute and store $\mathbf{F}_m^{(1)} = \mathcal{V}_m^{(1)} f(\mathcal{H}_m^{(1)}) \mathbf{E}_1 B$
 - 4: **for** $k = 1, 2, \dots$, until convergence **do**
 - 5: Run Algorithm 4.1 with inputs $A, \mathbf{V}_{m+1}^{(k)}, \mathbb{S}, \langle \cdot, \cdot \rangle_{\mathbb{S}}, N$, and m and store $\mathcal{V}_{m+1}^{(k+1)}$ in place of the previous basis
 - 6: Compute error approximation $\tilde{\mathbf{D}}_m^{(k)}$
 - 7: Update $\mathbf{F}_m^{(k+1)} := \mathbf{F}_m^{(k)} + \tilde{\mathbf{D}}_m^{(k)}$
 - 8: **end for**
 - 9: **return** $\mathbf{F}_m^{(k+1)}$
-

4.2 | Computational complexity and optimizations

Operation counts for B(FOM)² are not provided in Frommer et al.¹⁸, so we present our own calculations here. We first consider the two versions of block Arnoldi, classical and global, and then look at single cycle of B(FOM)².

4.2.1 | Sparse matrix times a block vector

We assume A is block circulant, where each $n \times n$ block is sparse with $\mathcal{O}(n)$ nonzero entries. We also assume that $p = \mathcal{O}(n)$. Then A itself can be stored in $\mathcal{O}(n^2)$ entries and does not need to be formed explicitly; see also the discussion in Kilmer et al.⁹. However, the action of A on a block vector will cost the same as for a matrix with $\mathcal{O}(n^3)$ nonzero entries because the full block circulant has p^2 blocks, so the product AV has complexity $\mathcal{O}(n^4)$.

4.2.2 | Classical Block Arnoldi

In addition to the action of A on block vectors, the main kernels of the classical version of Algorithm 1 are the QR factorization for block vectors, the inner product $\mathbf{X}^* \mathbf{Y}$, and multiplication between block vectors and small square matrices of the form \mathbf{VC} . The Householder QR factorization costs $\mathcal{O}(n^6)$,³¹ the inner product $\mathcal{O}(n^4)$, and the \mathbf{VC} product $\mathcal{O}(n^4)$. Running m steps of Algorithm 1 requires $m + 1$ QR factorizations, $m AV$ products, and $\frac{1}{2}m(m + 1)$ inner products and \mathbf{VC} products, with some negligible addition, for a total of

$$\mathcal{C}_{\text{cl-Arnoldi}} = \mathcal{O}((m + 1)n^6 + m(m + 2)n^4). \quad (15)$$

It is clear that the algorithm is dominated by QR factorizations.

4.2.3 | Global Block Arnoldi

The global version of Algorithm 1 can be made much cheaper than the classical one because the global algorithm effectively reduces to running an Arnoldi routine on vectors of size $n^2 p \times 1$ with the Frobenius norm and inner product. Thus the QR factorization is replaced by $\mathcal{O}(n^3)$, the cost of normalizing a vector; the inner product is also now only $\mathcal{O}(n^3)$, and the \mathbf{VC} product reduces to multiplying \mathbf{V} by a scalar, which is again $\mathcal{O}(n^3)$. The total cost of global Arnoldi is consequently dominated by AV products:

$$\mathcal{C}_{\text{gl-Arnoldi}} = \mathcal{O}\left(mn^4 + \left(\frac{1}{2}(m + 1)(m + 2)\right)n^3\right). \quad (16)$$

4.2.4 | Classical B(FOM)²

The computation of Equation (14) can be broken into three stages: $f(\mathcal{H}_m)$, a VC -type product, and the evaluation of the basis \mathcal{V}_m on the resulting matrix. The matrix function should be computed via a direct algorithm, such as the Schur-Parlett Algorithm, described in Chapter 9 of Higham⁷. Without function-specific information, `funm` requires up to $\mathcal{O}(m^4n^4)$; function-specific algorithms, or algorithms that take advantage of the eigenvalue distribution of A (described in other chapters of Higham⁷) may be cheaper. The product $f(\mathcal{H}_m)\mathbf{E}_1$ does not require computation, since we can just extract the first block column from $f(\mathcal{H}_m)$; $f(\mathcal{H}_m)\mathbf{E}_1\mathbf{B}$ then requires only $\mathcal{O}(mn^3)$. Finally, \mathcal{V}_m applied to an $mn \times n$ matrix requires $\mathcal{O}(mn^4)$. Including the Arnoldi cost (15), the total for computing (14) is then

$$\mathcal{C}_{\text{cl-B(FOM)}^2} = \mathcal{O}\left((m+1)n^6 + (m^4 + m(m+3))n^4 + mn^3\right). \quad (17)$$

4.2.5 | Global B(FOM)²

The same three stages apply for the global version of B(FOM)² as for the classical, but we can make many computations cheaper. The matrix function $f(\mathcal{H}_m) = f(H_m) \otimes I_n$, where the Kronecker product need not be formed explicitly, so the cost reduces to $\mathcal{O}(m^4)$. The matrix B can be regarded as a scalar, and using the same column-extracting trick, $f(\mathcal{H}_m)\mathbf{E}_1\mathbf{B}$ comes at a negligible cost, $\mathcal{O}(m)$. Finally, the product with the basis \mathcal{V}_m can be reduced to taking scalar combinations of the basis vectors, amounting to $\mathcal{O}((m-1)n^3)$. The total for Equation (14), including the Arnoldi costs (16) is

$$\mathcal{C}_{\text{gl-B(FOM)}^2} = \mathcal{O}\left(mn^4 + \left(\frac{1}{2}(m+1)(m+2) + m - 1\right)n^3 + m^4\right). \quad (18)$$

4.2.6 | Restarts

Determining the computational complexity for restarted B(FOM)² is challenging because the quadrature rule is adaptive, and the number of nodes per restart cycle plays a crucial role in how much work is done. Typically, the cost per additional cycle should be less than computing the first step, and it should decrease as the algorithm approaches convergence because the quadrature rule can be approximated progressively less accurately.³² In the worst-case scenario, however, the cost of successive cycles may be as expensive as the first, so it is reasonable to regard (17) and (18) as upper bounds.

4.3 | Block diagonalization and the discrete Fourier transform

Per recommendations in Kilmer et al.⁹ and Kilmer and Martin¹⁶, we can improve the computational effort of $f(\mathcal{A}) * \mathcal{B}$ by taking advantage of the fact that `bcirc(A)` can be block diagonalized by the discrete Fourier transform (DFT) along the tubal fibers of \mathcal{A} . Let F_p denote the DFT of size $p \times p$. Then we have that

$$(F_p \otimes I_n) \text{bcirc}(\mathcal{A}) (F_p^* \otimes I_n) = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_p \end{bmatrix} =: D,$$

where D_k are $n \times n$ matrices. Then by Theorem 2(iii),

$$f(\text{bcirc}(\mathcal{A})) = (F_p^* \otimes I_n) f(D) (F_p \otimes I_n).$$

Since each D_i , $i = 1, \dots, p$ may be a full $n \times n$ matrix, applying D itself to block vectors will still take $\mathcal{O}(n^4)$ operations. However, this structure is easier to parallelize and requires less memory-movement than using \mathcal{A} directly, which will play an important role in high-performance applications.

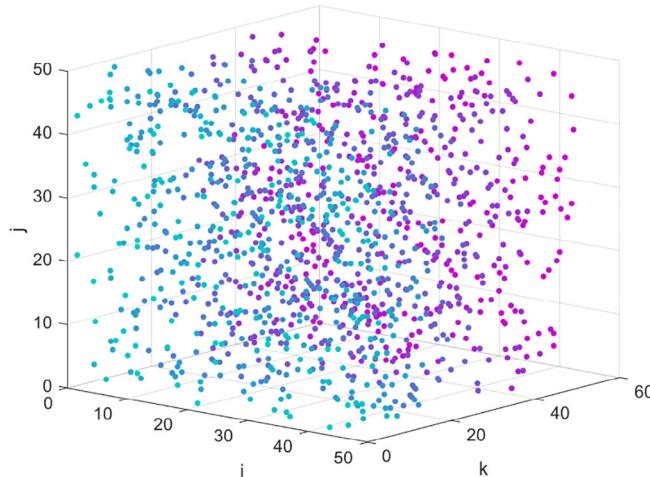


FIGURE 2 Sparsity structure for \mathcal{A} . Blue indicates that a face is closer to the “front” and pink farther to the “back;” see Figure 1f for how the faces are oriented

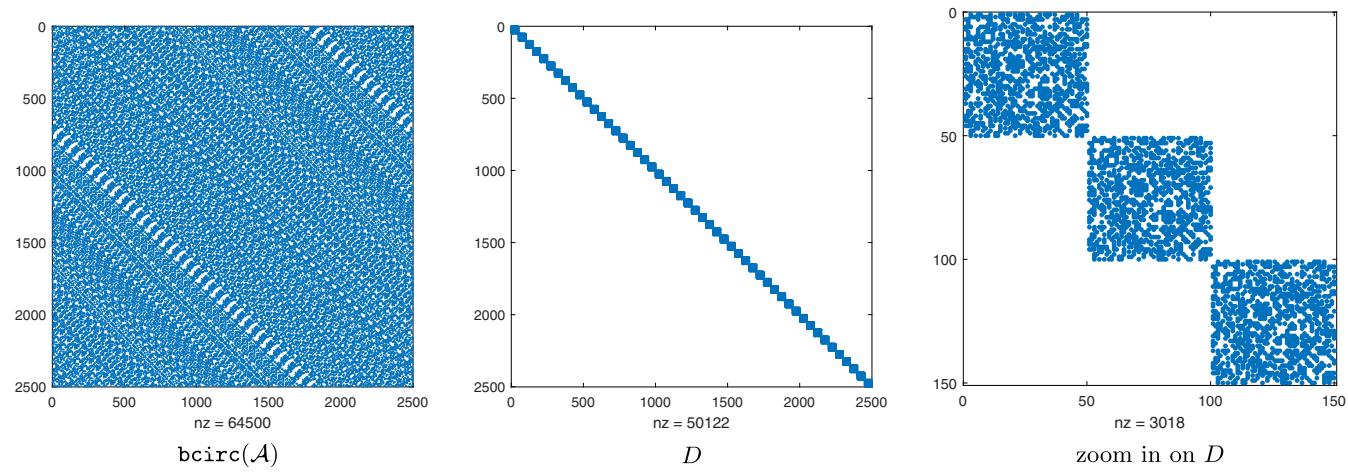


FIGURE 3 Sparsity patterns for block circulants

4.4 | The tensor t-exponential on a small third-order network

We take $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ to be a tensor whose p frontal faces are each adjacency matrices for an undirected, unweighted network. More specifically, the frontal faces of \mathcal{A} are symmetric, and the entries are binary. The sparsity structure of this tensor is given in Figure 2 for $n = p = 50$. Note that we must actually compute $\exp(\mathcal{A}) * \mathcal{I} = \text{fold}(\exp(\text{bcirc}(\mathcal{A}))\mathbf{E}_1)$ (cf. Definition (9)). With $n = p = 50$, this leads to a 2500×2500 matrix function times a 2500×50 block vector. The sparsity patterns of $\text{bcirc}(\mathcal{A})$ and D are shown in Figure 3. The block matrix D is determined by applying MATLAB's fast Fourier transform to $\text{bcirc}(\mathcal{A})$. Note that $\text{bcirc}(\mathcal{A})$ is not symmetric, but it has a banded structure. It should also be noted that while the blocks of D appear to be structurally identical, they are not numerically equal.

We compute $\exp(\mathcal{A}) * \mathcal{I}$ with the classical and global versions of B(FOM)² using the MATLAB software package `bfomfom`.^{*} We run MATLAB 2019a on Windows 10 on a laptop with 16 GB RAM and an Intel i7 processor at 1.80 GHz. The convergence behavior of each version is displayed in Figure 4, where we report the relative error per restart cycle, that is, per m iterations of Algorithm 1. The restart cycle length is $m = 5$, and the error tolerance is 10^{-12} . The methods based on D (case (A)) are only a little less accurate than those based on $\text{bcirc}(\mathcal{A})$ (case (B)), and they require the same

*The script `tensor_texp_network_v2.m` used to generate our results can be found at <https://gitlab.com/katlund/bfomfom-main>, along with the main body of code.

FIGURE 4 Convergence plots for (a) classical and global methods on $\exp(D)F_p \otimes I_n E_1$ and (b) classical and global methods on $\exp(\text{bcirc}(\mathcal{A}))E_1$. $m = 5$

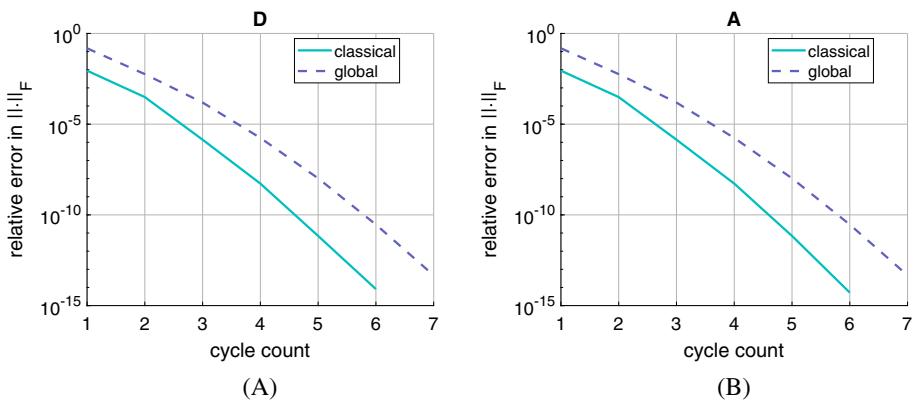
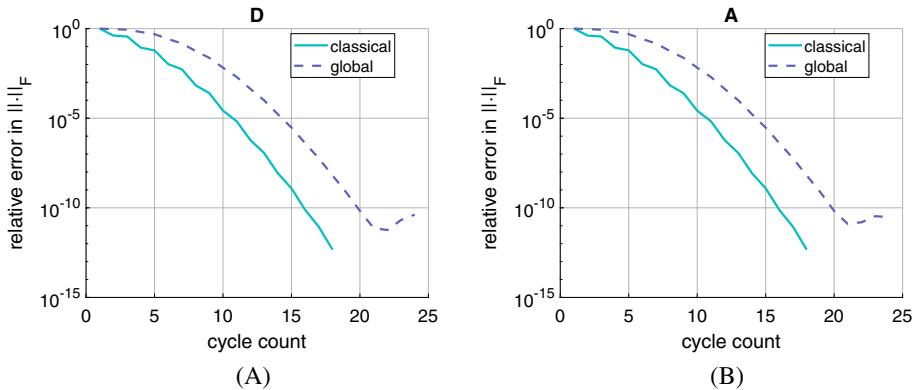


TABLE 1 Number of cycles needed to converge to 10^{-12} for different basis sizes m

	$m = 2$	$m = 5$	$m = 10$	$m = 15$
Classical	18	6	3	2
Global	—	7	3	2

FIGURE 5 Convergence plots for (a) classical and global methods on $\exp(D)F_p \otimes I_n E_1$ and (b) classical and global methods on $\exp(\text{bcirc}(\mathcal{A}))E_1$. $m = 2$



number of iterations to converge. The global methods require only one more cycle than the classical ones, but considering the computational complexity per cycle (cf. Equations (17) and (18)), it is clear that the global methods require far less work overall.

Table 1 shows that for larger m , both classical and global methods require the same number of cycles (for either D - or $\text{bcirc}(\mathcal{A})$ -based approaches). For smaller values of m , global methods cannot attain the desired tolerance because they exceed the maximum number of quadrature nodes allowed to perform the error update in line 6 of Algorithm 2. See, however, Figure 5 for the convergence behavior of the global method when $m = 2$. It still attains a high level of accuracy with much less work overall than the classical method.

5 | CONCLUSION

The main purpose of this report is to establish a first notion for functions of multidimensional arrays and demonstrate that it is feasible to compute this object with well understood tools from the matrix function literature. Our definition for the tensor t-function $f(\mathcal{A}) * \mathcal{B}$ shows versatility and consistency, and our numerical results indicate that block KSMs can compute $f(\mathcal{A}) * \mathcal{B}$ with few iterations and still achieve high accuracy. In particular, the global block KSM shows promise for moderate sizes, since its overall workload is significantly smaller than that of its classical counterpart. For smaller basis sizes, which are more favorable in the context of large tensors, global methods may struggle to converge, and remedies for this situation remain an open problem. One potential solution, that should first be explored for simple matrix functions,

is to switch between global and classical paradigms in some optimal way so as to minimize overall computational effort while maximizing attainable accuracy.

The second aim of this report is to invite fellow researchers to pursue the many open problems posed by this new definition and to devise tensor function definitions for other paradigms. Other key problems include exploring applications of $f(\mathcal{A}) * \mathcal{B}$ in real-life scenarios and comparing our definition of communicability for a third-order network to existing network analysis tools.

ACKNOWLEDGEMENTS

The author would like to thank Misha Kilmer for useful conversations and the images used in Figure 1, Andreas Frommer and Daniel B. Szyld for comments on Theorem 6, Francesca Arrigo for suggesting a new application in Section 3, and the anonymous referee for multiple insights.

CONFLICT OF INTEREST

This author declares no conflicts of interest.

ORCID

Kathryn Lund  <https://orcid.org/0000-0001-9851-6061>

REFERENCES

1. Estrada E, Higham DJ. Network properties revealed through matrix functions. *SIAM Rev.* 2010;52:696–714. <https://doi.org/10.1137/090761070>.
2. Hochbruck M, Ostermann A. Exponential integrators. *Acta Numer.* 2010;19:209–286. <https://doi.org/10.1017/S0962492910000048>.
3. Cichocki A. *Era of big data processing: A new approach via tensor networks and tensor decompositions*. Tech. Report; 2014. arXiv:1403.2048v4.
4. Khoromskij BN. Tensor numerical methods for multidimensional PDES: Theoretical analysis and initial applications. *ESAIM Proc Surv.* 2015;48:1–28. <https://doi.org/10.1051/proc/201448001>.
5. Kolda TG, Bader BW. Tensor decompositions and applications. *SIAM Rev.* 2008;51:455–500. <https://doi.org/10.1137/07070111X>.
6. Arrigo F, Benzi M, Fenu C. Computation of generalized matrix functions. *SIAM J Matrix Anal Appl.* 2016;37:836–860. <https://doi.org/10.1137/15M1049634>.
7. Higham NJ. *Functions of Matrices*. Philadelphia, PA: SIAM, 2008.
8. De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM J Matrix Anal Appl.* 2000;21:1253–1278. <https://doi.org/10.1137/S0895479896305696>.
9. Kilmer ME, Braman K, Hao N, Hoover RC. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J Matrix Anal Appl.* 2013;34:148–172. <https://doi.org/10.1137/110837711>.
10. Kolda TG, Mayo JR. Shifted power method for computing tensor eigenpairs. *SIAM J Matrix Anal Appl.* 2011;32:1095–1124.
11. Lim LH. Singular values and eigenvalues of tensors: A variational approach. Paper presented at: Proceedings of the IEEE International Workshop on Computational Advances Multi-Sensor Adaptive Processing (CAMSAP '05); vol. 3, 2005:129–132. Available from: <http://ieeexplore.ieee.org/document/1574201/>.
12. Ng M, Qi L, Zhou G. Finding the largest eigenvalue of a nonnegative tensor. *SIAM J Matrix Anal Appl.* 2009;31:1090–1099.
13. Qi L. Eigenvalues of a real supersymmetric tensor. *J Symb Comput.* 2005;40:1302–1324. <https://doi.org/10.1016/j.jsc.2005.05.007>.
14. Qi L. Eigenvalues and invariants of tensors. *J Math Anal Appl.* 2007;325:1363–1377. <https://doi.org/10.1016/j.jmaa.2006.02.071>.
15. Braman K. Third-order tensors as linear operators on a space of matrices. *Linear Algebra Appl.* 2010;433:1241–1253. <https://doi.org/10.1016/j.laa.2010.05.025>.
16. Kilmer ME, Martin CD. Factorization strategies for third-order tensors. *Linear Algebra Appl.* 2011;435:641–658. <https://doi.org/10.1016/j.laa.2010.09.020>.
17. Al-Mohy AH, Higham NJ. Computing the action of the matrix exponential with an application to exponential integrators. *SIAM J Sci Comput.* 2011;33:488–511. <https://doi.org/10.1137/100788860>.
18. Frommer A, Lund K, Szyld DB. Block Krylov subspace methods for functions of matrices. *Electron Trans Numer Anal.* 2017;47:100–126.
19. Frommer A, Lund K, Szyld DB. Block Krylov subspace methods for functions of matrices II: Modified block FOM. *SIAM J Matrix Anal Appl.* 2020.
20. Simoncini V, Lopez L. Analysis of projection methods for rational function approximation to the matrix exponential. *SIAM J Numer Anal.* 2006;44:613–635. <https://doi.org/10.1137/05062590>.
21. Betten J. *Creep Mechanics*. 3rd ed. Berlin, Germany: Springer, 2008.
22. Zheng Q-S. Theory of representations for tensor functions- A unified invariant approach to constitutive equations. *Appl Mech Rev.* 1994;47:545. <https://doi.org/10.1115/1.3111066>.
23. Frommer A, Simoncini V. Matrix functions. In: Schilders WHA, vander Vorst HA, Rommes J, editors. *Model order reduction: Theory research aspects applications of mathematics in industry*. Volume 13. Berlin, Germany: Springer, 2008; p. 275–304.

24. Gleich DF, Greif C, Varah JM. The power and arnoldi methods in an algebra of circulants. *Numer Linear Algebra Appl.* 2013;20:809–831. <https://doi.org/10.1002/nla>.
25. Davis PJ. Circulant Matrices. 2nd ed. Providence, RI: AMS Chelsea Publishing, 2012.
26. Horn RA, Johnson CR. Topics in Matrix Analysis. Cambridge, MA: Cambridge University Press, 1991.
27. Domenico MD, Solé-Ribalta A, Omodei E, Gómez S, Arenas A. Centrality in interconnected multilayer networks. *Phys D Nonlinear Phenom.* 2013;323:1–12. <https://doi.org/10.1016/j.physd.2016.01.002>.
28. Zhang Z, Ely Z, Aeron S, Hao N, Kilmer ME. Novel methods for multilinear data completion and de-noising based on tensor-SVD. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition;* 2014:3842–3849. <https://doi.org/10.1109/CVPR.2014.485>.
29. Elbouyahyaoui L, Messaoudi A, Sadok H. Algebraic properties of the block GMRES and block Arnoldi methods. *Electron Trans Numer Anal.* 2008;33:207–220.
30. Saad Y. Iterative Methods for Sparse Linear Systems. 2nd ed. Philadelphia, PA: SIAM, 2003.
31. Trefethen LN, Bau DI. Numerical linear algebra. Philadelphia, PA: SIAM, 1997.
32. Frommer A, Güttel S, Schweitzer M. Efficient and stable Arnoldi restarts for matrix functions based on quadrature. *SIAM J Matrix Anal Appl.* 2014;35:661–683.

How to cite this article: Lund K. The tensor t-function: A definition for functions of third-order tensors. *Numer Linear Algebra Appl.* 2020;e2288. <https://doi.org/10.1002/nla.2288>