

STRUCTURED MATRIX APPROXIMATIONS VIA TENSOR DECOMPOSITIONS*

MISHA E. KILMER[†] AND ARVIND K. SAIBABA[‡]

Abstract. We provide a computational framework for approximating a class of structured matrices; here, the term structure is very general, and may refer to a regular sparsity pattern (e.g., block banded), or be more highly structured (e.g., symmetric block Toeplitz). The goal is to uncover *additional latent structure* that will in turn lead to computationally efficient algorithms when the new structured matrix approximations are employed in place of the original operator. Our approach has three steps: map the structured matrix to tensors, use tensor compression algorithms, and map the compressed tensors back to obtain two different matrix representations—sum of Kronecker products and block low-rank format. The use of tensor decompositions enables us to uncover latent structure in the problem and leads to compressed representations of the original matrix that can be used efficiently in applications. The resulting matrix approximations are memory efficient, easy to compute with, and preserve the error that is due to the tensor compression in the Frobenius norm. Our framework is quite general. We illustrate the ability of our method to uncover block-low-rank format on structured matrices from two applications: system identification and space-time covariance matrices. In addition, we demonstrate that our approach can uncover the sum of structured Kronecker products structure on several matrices from the SuiteSparse collection. Finally, we show that our framework is broad enough to encompass and improve on other related results from the literature, as we illustrate with the approximation of a three-dimensional blurring operator.

Key words. structured matrices, tensor decompositions, Kronecker products, system identification, image deblurring, covariance matrices

MSC codes. 65F55, 65F50, 65F99

DOI. 10.1137/21M1418290

1. Introduction. Structured matrices are ubiquitous in many applications such as signal processing, image deblurring, finite difference discretizations of partial differential equations (PDEs), and control theory. For example, modeling a system in which the operator is spatially invariant in both the vertical and horizontal directions, with zero boundary conditions (e.g., blur in an image) leads, upon discretization, to a matrix that is block Toeplitz with Toeplitz blocks (BTTB). Discretization of PDE models often leads to matrices that are at least block banded and perhaps possess additional structure. Matrices with such structure are advantageous for iterative linear solvers such as Krylov subspace methods because the required matrix-vector products per iteration can usually be done efficiently by leveraging the structure.

However, many applications demand a factorization of the operator that reveals information about the rank or will allow for a low-rank approximation. For large scale

*Received by the editors May 7, 2021; accepted for publication (in revised form) by L. Grigori June 21, 2022; published electronically November 18, 2022.

<https://doi.org/10.1137/21M1418290>

Funding: The work of the authors was partially supported by the National Science Foundation (NSF) grants DMS 1821148, DMS-1821149, and Tufts T-Tripods Institute NSF HDR grant CCF-1934553. The work of the first author was also partially supported by the NSF grant DMS-1439786 and the Simons Foundation grant 50736 while in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the “Model and dimension reduction in uncertain and dynamic systems” program.

[†]Department of Mathematics, Tufts University, Medford, MA 02155 USA (misha.kilmer@tufts.edu).

[‡]Department of Mathematics, North Carolina State University, Raleigh, NC 27695 USA (asaibab@ncsu.edu).

problems, the classes of structured matrices for which such decompositions are easily and directly computable are limited, e.g., matrices that are diagonalized or block-diagonalized by fast transform based methods, or matrices generated by a single Kronecker product. One of the purposes of this paper, therefore, is to expand the classes of structured matrices for which low-rank factorizations can be obtained. Specifically, we give a framework for computing block low-rank format which is applicable to a general class of structured matrices.

There are many applications in which the operator is expressed naturally as a sum of Kronecker products [31]. If the number of the terms in the sum is small, the Kronecker structure can be leveraged to compute matrix-vector products quickly, or to develop preconditioners or approximate truncated SVD filters (e.g., [7]). Storage of the relevant components needed to perform products with the operator or its transpose can be highly efficient.

Our goal in this paper therefore is to compute operator approximations that discover either/both the Kronecker sum structure and block low-rank structure while preserving other underlying matrix structure, such as (block) Toeplitz or Hankel, (block-)bandedness, symmetry and positive (semi)definiteness. We propose to arrive at these operator approximations in an innovative way in three steps: first, we remove redundancies through a matrix-to-tensor mapping, with the order of the tensor as determined by properties of the matrix; second, we utilize tensor-decomposition methods at our disposal to compute a tensor approximation; finally, we obtain our matrix approximation by mapping the tensor approximation back to an operator by inverting our matrix-to-tensor mapping.

Contributions. We summarize the contents of the paper and highlight the major contributions. We develop a tensor-based framework for computing structure-preserving matrix approximations for use in applications from scientific computing.

1. We develop a mapping between structured block matrices to tensors. The tensors can then be compressed and stored in the appropriate tensor format. We then relate the error in tensor representation to the matrix representation and show that these errors are the same in the Frobenius norm.
2. We use the canonical polyadic (CP) and Tucker formats for compressing the tensor, and we show how they can be used for two different efficient matrix representations—Kronecker product sums and block low-rank formats. Our methods are structurepreserving in the sense that both of these efficient matrix representations retain important features from the original structured block matrices. In particular, we pay special attention to preserving positive definiteness and semidefiniteness.
3. We demonstrate the computational benefits of our approach on two different application areas: eigensystem realization algorithm for system identification and space-time covariance matrices, which yield block Hankel and (symmetric) block Toeplitz matrices, respectively. We also demonstrate our approach on several sparse block tridiagonal test matrices from the SuiteSparse matrix collection.
4. We show how to extend this approach that we developed for structured block matrices to multilevel structured block matrices. We show how to compute efficient Kronecker product sums after compressing the tensor using the Tucker format.
5. We discuss how our tensor-based approaches are related to and can recover existing algorithms from the literature and how we might use our approach to improve on some of them.

Relation to previous literature. The present approach differs from the methods in the literature on direct operator approximation by use of tensor train tensor approximations and related tensor formats [22, 23, 24, 25], which are particularly effective methods in generating compact, approximate representations to high dimensional operators in order to solve linear systems. Our methodology and goals are different. We also mention the work in [3] that has similar goals to ours but is different both in the way the tensors are constructed and compressed. Furthermore, in our approach, several types of structured approximations are possible, including Kronecker-sum, and block-structured low-rank approximation. The appeal of our approach is the general nature of the term “structure” as well as the fact that it is agnostic to which type of tensor approximation algorithm is employed. Our approach is flexible in that any tensor approximation method can be utilized in the second step, and we can provably relate the quality of our matrix approximation. As we discuss in section 7, the framework we provide here either subsumes or allows improvement upon some other methods in the literature that aim to provide optimal/near optimal Kronecker product approximations to structured matrices, such as those in [12, 21].

Organization and contents. The first part of this paper is devoted to a general formulation of structure and developing a matrix-to-tensor invertible mapping that can utilize this formulation. While we describe this in detail for two level matrices having structure on the outermost layer, we show that our method is applicable for higher level structure as well. Since the matrix to tensor mapping is invertible, if we are given an approximation to the tensor representation of our matrix, we can map this approximation back to a matrix approximation. We give a result that relates the quality of the tensor approximation to the quality of the matrix approximation. Next, we investigate the specific structure that the matrix approximation can inherit depending on the type of tensor approximation used. For this, we focus on a CP formulation of our tensor approximation, and on a truncated higher order SVD (HOSVD) based approximation.

This paper is organized as follows. Section 2 describes the notation and terminology, including a brief description of the CP and Tucker decompositions. In section 3, we describe the block matrix-to-tensor mapping, with examples for clarity. We also give a theoretical result that relates the tensor approximation problem to the matrix approximation problem. Section 4 describes how we obtain Kronecker sum approximations to the operator from CP or Tucker approximations to the tensor. The subject of section 5 is how to obtain matrix approximations that are block-structured factorization from the CP or Tucker approximations to the original tensor. To this end, we also discuss how to preserve symmetric positive definiteness. An application in subspace system identification and in space-time covariance matrices is given in section 6. We also process several matrices from the SparseSuite in section 6. Our framework is generalized to capture higher level structure in section 7. As an illustration of the utility of our approach, we highlight how our framework can be used to capture or improve on a few previous results in the literature. Conclusions and future work are addressed in section 8.

2. Background and notation. We briefly clarify the notation that we use in this paper and some background material on Kronecker products and tensors. In this brief review, we mostly focus on third order tensors; however, they can be readily extended to higher order tensors and we refer to [14, 9] for the details.

2.1. Notation and terminology. Throughout this paper, matrices are defined in bold uppercase, vectors are in bold lowercase, and third order tensors are in calligraphic boldface script.

Let $\mathcal{A} \in \mathbb{R}^{m \times p \times n}$. We denote the frontal slices as $\mathcal{A}_{:, :, j}$, lateral slices as $\mathcal{A}_{:, j, :}$, and horizontal slices as $\mathcal{A}_{j, :, :}$. Following [13], we use invertible mappings between $m \times n$ matrices and $m \times 1 \times n$ tensors by twisting and squeezing,¹ i.e., $\mathbf{X} \in \mathbb{C}^{m \times n}$ is related to $\mathcal{X}_{:, j, :}$ via

$$\mathcal{X} = \text{twist}(\mathbf{X}) \text{ and } \mathbf{X} = \text{sq}(\mathcal{X}_{:, j, :}).$$

Tensor unfoldings refer to unwrapping a tensor into a matrix. For example, a third order tensor $\mathcal{A} \in \mathbb{R}^{m \times p \times n}$ admits three unfoldings [14]:

$$\begin{aligned} \mathbf{A}_{(1)} &= [\mathcal{A}_{:, :, 1}, \mathcal{A}_{:, :, 2}, \dots, \mathcal{A}_{:, :, n}] \in \mathbb{C}^{m \times np}, \\ \mathbf{A}_{(2)} &= [\mathcal{A}_{:, :, 1}^T, \mathcal{A}_{:, :, 2}^T, \dots, \mathcal{A}_{:, :, n}^T] \in \mathbb{C}^{p \times mn}, \end{aligned}$$

and

$$(1) \quad \mathbf{A}_{(3)} = [\text{sq}(\mathcal{A}_{:, 1, :})^T, \text{sq}(\mathcal{A}_{:, 2, :})^T, \dots, \text{sq}(\mathcal{A}_{:, p, :})^T] \in \mathbb{C}^{n \times mp}.$$

The unfoldings can be used to define modewise products between matrices and tensors. For example, let

$$\mathcal{A} \times_i \mathbf{U} \equiv \mathbf{U} \mathbf{A}_{(i)}, \quad i = 1, 2, 3,$$

where it is clear that the column dimension of \mathbf{U} must match the row dimension of $\mathbf{A}_{(i)}$ and the resulting tensor must have i th mode of dimension r , where r is the number of rows in \mathbf{U} .

If $\mathbf{a} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^\ell$, $\mathbf{c} \in \mathbb{R}^n$, then

$$\mathcal{A} := \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \in \mathbb{R}^{m \times \ell \times n}$$

is a third order, rank-1 tensor, with (i, j, k) entry of \mathcal{A} given by $\mathbf{a}_i \mathbf{b}_j \mathbf{c}_k$.

For convenience, we may also use the `vec` and `reshape` commands to define mappings between matrices and vectors by column unwrapping and reshaping:

$$\mathbf{a} = \text{vec}(\mathbf{A}) \in \mathbb{C}^{mn} \leftrightarrow \mathbf{A} = \text{reshape}(\mathbf{a}, [m, n]).$$

2.2. Kronecker products. As we will see, the recovered matrix approximations all involve one or more Kronecker products. Let $\mathbf{A} \in \mathbb{R}^{m \times p}$ and $\mathbf{B} \in \mathbb{R}^{n \times \ell}$. Then their Kronecker product $\mathbf{B} \otimes \mathbf{A} \in \mathbb{R}^{(mn) \times (p\ell)}$ is the block matrix

$$\mathbf{B} \otimes \mathbf{A} = \begin{bmatrix} b_{11}\mathbf{A} & b_{12}\mathbf{A} & \cdots & b_{1p}\mathbf{A} \\ b_{21}\mathbf{A} & b_{22}\mathbf{A} & \ddots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ b_{m1}\mathbf{A} & \cdots & \cdots & b_{mn}\mathbf{A} \end{bmatrix}.$$

Kronecker products satisfy some important properties [31] which can be used to great computational advantage in applications. We review a few of these properties:

1. $\text{vec}(\mathbf{A}\mathbf{X}\mathbf{B}^T) \equiv (\mathbf{B} \otimes \mathbf{A})\text{vec}(\mathbf{X})$, where $\text{vec}(\cdot)$ is a vectorization operator that converts a matrix to a column vector by stacking its columns.

¹If \mathcal{A} is $m \times 1 \times n$, the MATLAB command `squeeze(\mathcal{A})` returns the $m \times n$ matrix. Note, however, that if the first mode has dimension 1, the `squeeze` command in MATLAB behaves differently, depending on the size of the other modes. For example, the `squeeze` of a $1 \times 1 \times n$ tensor produces an length- n column vector, the `squeeze` of a $1 \times m \times n$ produces an $m \times n$ matrix.

2. For $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ of appropriate dimension, $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$.
3. Kronecker products of orthogonal matrices are orthogonal.
4. If \mathbf{A} is a structured matrix (e.g., diagonal, banded, upper triangular, Toeplitz, etc.), then $\mathbf{A} \otimes \mathbf{B}$ will inherit that structure on the block level.
5. The SVD of $\mathbf{A} \otimes \mathbf{B}$, is obtained from the SVDs $\mathbf{A} = \mathbf{WSQ}^\top$ and $\mathbf{B} = \mathbf{UDV}^\top$ since $(\mathbf{UDV}^\top) \otimes (\mathbf{WSQ}^\top) = (\mathbf{U} \otimes \mathbf{W})(\mathbf{D} \otimes \mathbf{S})(\mathbf{V} \otimes \mathbf{Q})^\top$. Note that though the singular values of $\mathbf{A} \otimes \mathbf{B}$ are contained in the diagonals of the diagonal matrix $\mathbf{D} \otimes \mathbf{S}$, they are likely not in order, so the ordered SVD would require an additional permutation.
6. As a consequence of the last point, $\|\mathbf{A} \otimes \mathbf{B}\|_F = \|\mathbf{A}\|_F \|\mathbf{B}\|_F$.

2.3. CP decomposition. Suppose the tensor $\mathcal{A} \in \mathbb{R}^{m \times p \times n}$ can be expressed as

$$\mathcal{A} = \sum_{i=1}^r \mathbf{X}_{:,i} \circ \mathbf{Y}_{:,i} \circ \mathbf{Z}_{:,i},$$

where $\mathbf{X} \in \mathbb{R}^{m \times r}$, $\mathbf{Y} \in \mathbb{R}^{p \times r}$, $\mathbf{Z} \in \mathbb{R}^{n \times r}$ are called the *factor matrices*. The representation is also expressed using the so-called Kruskal notation in terms of the factor matrices $[\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$. Such a decomposition is called the CP² decomposition of a tensor. If r is minimal, then r is called the rank of the tensor. Unfortunately, determining the rank of a tensor is in general an NP-hard problem. Moreover, the factor matrices do not need to have independent, let alone orthogonal, columns. For information about properties of the CP decomposition, see [14].

Independent of whether or not r is minimal, if there exists a triple of factor matrices such that $\mathcal{A} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$, then we can use the factor matrices to deduce relationships among the various slices of the tensor. For example, the k th lateral slice has entries corresponding to the triple matrix product

$$(2) \quad \text{sq}(\mathcal{A}_{:,k,:}) = \mathbf{X} \text{diag}(\mathbf{Y}_{k,:}) \mathbf{Z}^\top, \quad k = 1, \dots, p,$$

with similar formulations for the other slices. Thus, a CP decomposition of a third order tensor reveals a “joint diagonalization” of the matrices that comprise the slices of the tensor, although the matrices in the diagonalization need to be neither square nor invertible. A popular way of computing the CP decomposition is the alternating least squares approach; see, e.g., [14, section 3.4].

2.4. Tucker decomposition. The Tucker-3 factorization [30] of a third order tensor $\mathcal{A} \in \mathbb{R}^{m \times p \times n}$ is

$$\mathcal{A} = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} = \sum_{i=1}^{R_1} \sum_{j=1}^{R_2} \sum_{k=1}^{R_3} \mathcal{G}_{i,j,k} (\mathbf{U}_{:,i} \circ \mathbf{V}_{:,j} \circ \mathbf{W}_{:,k}),$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is the core tensor, $\mathbf{U} \in \mathbb{R}^{m \times R_1}$, $\mathbf{V} \in \mathbb{R}^{p \times R_2}$, $\mathbf{W} \in \mathbb{R}^{n \times R_3}$. The matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$ are also called factor matrices. We say that the tensor is rank- (R_1, R_2, R_3) , if $R_j = \text{rank}(\mathbf{A}_{(j)})$ for $j = 1, \dots, 3$. The core is not typically “diagonal” and the elements need not be nonnegative. If the core is diagonal, and $R_1 = R_2 = R_3$ then this decomposition reduces to a CP decomposition.

For later use, we highlight one important feature that relates the mode-3 unfolding of \mathcal{A} with the mode-3 unfolding of the core \mathcal{C} and the factor matrices [14]:

$$(3) \quad \mathcal{A}_{(3)} = \mathbf{W} \mathcal{G}_{(3)} (\mathbf{V}^\top \otimes \mathbf{U}^\top).$$

²Also goes by CANDECOMP/PARAFAC, canonical factors, etc.

Combining (3) with (1), we observe that the k th lateral slice of \mathcal{A} is given by

$$(4) \quad \text{sq}(\mathcal{A}_{:,k,:}) = \mathbf{U} \left(\sum_{j=1}^{r_2} v_{k,j} \text{sq}(\mathcal{G}_{:,j,:}) \right) \mathbf{W}^\top, \quad k = 1, \dots, p.$$

Of particular interest is the HOSVD [6] for obtaining a compressed representation in the Tucker format. Suppose we want to compress $\mathcal{A} \in \mathbb{R}^{m \times p \times n}$ and the target rank is (r_1, r_2, r_3) . The HOSVD first proceeds by computing the SVDs of the 3 unfoldings and letting $\mathbf{U}, \mathbf{V}, \mathbf{W}$ correspond to the top r_1, r_2, r_3 left singular vectors of each mode unfolding. Next, the core tensor is obtained as $\mathcal{G} = \mathcal{A} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top$, so that we have the approximation

$$\mathcal{A} \approx \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}.$$

For short, we write $\mathcal{A} \approx [\mathcal{G}; \mathbf{U}, \mathbf{V}, \mathbf{W}]$. The accuracy of the HOSVD can be established by the results in [9, Theorem 10.3]. Besides the HOSVD, there are other efficient algorithms for approximating a tensor in the Tucker format. For some recent work on randomized algorithms in the Tucker format, see [19, 29].

3. Block matrix-to-tensor mapping. In this section, we consider a block matrix $\mathbf{A} \in \mathbb{R}^{(\ell m) \times (qn)}$ with $\ell \times q$ blocks of size $m \times n$ each. The basic idea is to first represent the matrix \mathbf{A} using a sum of Kronecker products. Let $\mathbf{A}_k \in \mathbb{R}^{m \times n}$ be a *distinct* subblock of a matrix which repeats $\eta_k > 0$ times for $k = 1, \dots, p$. Here $1 \leq p \leq \ell q$ is the number of distinct subblocks. Define the tuple of matrices³

$$\mathcal{A} = (\mathbf{A}_1, \dots, \mathbf{A}_p), \quad \mathcal{E} = (\mathbf{E}_1, \dots, \mathbf{E}_p),$$

where $\mathbf{E}_k \in \mathbb{R}^{\ell \times q}$ with entries

$$[\mathbf{E}_k]_{ij} = \begin{cases} \frac{1}{\sqrt{\eta_k}} & \text{if } \mathbf{A}_k \text{ occurs in } (i, j)\text{th block of } \mathbf{A}, \\ 0 & \text{otherwise.} \end{cases}$$

The matrices \mathbf{E}_k both account for the position and repetition of the blocks \mathbf{A}_k and satisfy

$$(5) \quad \|\mathbf{E}_k\|_F = 1, \quad k = 1, \dots, p.$$

To express the original matrix \mathbf{A} as the sum of Kronecker products, we define

$$(6) \quad \text{struct}_{\mathcal{E}}(\mathcal{A}) = \sum_{k=1}^p \mathbf{E}_k \otimes (\sqrt{\eta_k} \mathbf{A}_k).$$

Then it is verified that $\text{struct}_{\mathcal{E}}(\mathcal{A}) = \mathbf{A}$.

Remark 3.1. Consider the trace inner product $\langle \mathbf{C}, \mathbf{D} \rangle = \text{trace}(\mathbf{C}^\top \mathbf{D})$ for matrices \mathbf{C}, \mathbf{D} of the same size. If we let $\mathbf{C} = \mathbf{E}_j \otimes \mathbf{K}$ and $\mathbf{D} = \mathbf{E}_k \otimes \mathbf{L}$ for $\mathbf{K}, \mathbf{L} \in \mathbb{R}^{m \times n}$ and $j \neq k$, then \mathbf{C} and \mathbf{D} are orthogonal with respect to this inner product, since

$$\langle \mathbf{C}, \mathbf{D} \rangle = \text{trace}(\mathbf{E}_j^\top \mathbf{E}_k) \text{trace}(\mathbf{K}^\top \mathbf{L}) = 0.$$

This means that in (6) we have decomposed \mathbf{A} into a sum of orthogonal matrices with respect to the trace inner product.

³This tuple is unique up to permutation.

We illustrate this data structure through several commonly occurring examples. For simplicity, we take the number of blocks $\ell = q$, but the general case can be handled easily.

Example 1 (block-diagonal matrix). We consider a block-diagonal matrix of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & & & \\ & \mathbf{A}_2 & & \\ & & \ddots & \\ & & & \mathbf{A}_p \end{bmatrix} \in \mathbb{R}^{m\ell \times n\ell}.$$

In this case we have $p = \ell = q$ and

$$\mathcal{A} = (\mathbf{A}_1, \dots, \mathbf{A}_p), \quad \mathcal{E} = (\mathbf{e}_1 \mathbf{e}_1^\top, \dots, \mathbf{e}_p \mathbf{e}_p^\top),$$

where \mathbf{e}_j is the j th column of a $p \times p$ identity matrix.

Example 2 (block-banded matrix). If the matrix is block tridiagonal with $\ell \times \ell$ blocks, then $p = 3\ell - 2$ (block nonsymmetric) and $p = 2\ell - 1$ (block symmetric). More generally, if the matrix is block banded and we denote by b the block semibandwidth, then $p = m(2b + 1) - b(b + 1)$ if the matrix is block non symmetric, and $p = m(b + 1) - b(b + 1)/2$ if block symmetric. For example, consider a block-tridiagonal matrix of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & 0 & 0 \\ \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 & 0 \\ 0 & \mathbf{A}_4 & \mathbf{A}_5 & \mathbf{A}_6 \\ 0 & 0 & \mathbf{A}_6 & \mathbf{A}_7 \end{bmatrix} \in \mathbb{R}^{4m \times 4n}.$$

In this case we have $p = 7$,

$$\mathcal{A} = (\mathbf{A}_1, \dots, \mathbf{A}_7), \quad \mathcal{E} = (\mathbf{e}_1 \mathbf{e}_1^\top, \mathbf{e}_2 \mathbf{e}_1^\top, \dots, \mathbf{e}_3 \mathbf{e}_4^\top, \mathbf{e}_4 \mathbf{e}_4^\top),$$

where \mathbf{e}_j is the j th column of a 4×4 identity matrix.

Example 3 (block-Toeplitz). Consider the block Toeplitz matrix of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_{\ell+1} & \dots & \mathbf{A}_{2\ell-1} \\ \mathbf{A}_2 & \mathbf{A}_1 & \dots & \mathbf{A}_{2\ell-2} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{A}_\ell & \dots & \mathbf{A}_2 & \mathbf{A}_1 \end{bmatrix} \in \mathbb{R}^{\ell m \times \ell n}.$$

Define the shift matrices

$$\mathbf{E}_d = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbb{R}^{\ell \times \ell}$$

and $\mathbf{E}_u = \mathbf{E}_d^\top$. Then

$$\mathcal{A} = (\mathbf{A}_1, \dots, \mathbf{A}_{2\ell-1})$$

and

$$\mathcal{E} = \left(\frac{1}{\sqrt{\ell}} \mathbf{E}_d^0, \frac{1}{\sqrt{\ell-1}} \mathbf{E}_d, \dots, \mathbf{E}_d^{\ell-1}, \frac{1}{\sqrt{\ell-1}} \mathbf{E}_u, \frac{1}{\sqrt{\ell-2}} \mathbf{E}_u, \dots, \mathbf{E}_u^{\ell-1} \right).$$

If the matrix is block symmetric then additional savings are possible since $p = \ell$. Similarly, if the matrix is additionally block banded, then further savings are possible. A similar approach is possible if the matrix is block Hankel instead of block Toeplitz.

In each case above, the reader is invited to verify that (6) holds. This representation is unique up to reordering of the terms in the summation.

Usually, knowledge of the problem structure can be used to find this representation. If no knowledge is available, and a purely algebraic technique is desired, this representation can be determined by a greedy search. Assuming that it takes $\mathcal{O}(1)$ operations to verify if the blocks are equal, this representation can be computed in $\mathcal{O}((\ell q)^2)$ operations (worst case complexity), but the complexity is much lower if there is latent structure (e.g., block Toeplitz, block diagonal, etc.) to be discovered. Furthermore, if the block sizes are large it may be advantageous to store the matrices in \mathcal{E} in compressed sparse column representation.

With this matrix $\mathbf{A} \in \mathbb{R}^{(\ell m) \times (qn)}$, and associated data structures \mathcal{A} and \mathcal{E} , we associate a third order tensor mapping $\mathcal{T}_{\mathcal{E}} : \mathbb{R}^{\ell m \times qn} \rightarrow \mathbb{R}^{m \times p \times n}$ defined in terms of its lateral slices as

$$\mathcal{T}_{\mathcal{E}}[\mathbf{A}]_{:,k,:} = \sqrt{\eta_k} \text{twist}(\mathbf{A}_k), \quad k = 1, \dots, p.$$

The tensor-to-matrix mapping $\mathcal{M}_{\mathcal{E}} : \mathbb{R}^{m \times p \times n} \rightarrow \mathbb{R}^{\ell m \times qn}$ is defined as

$$\mathcal{M}_{\mathcal{E}}[\mathcal{X}] := \text{struct}_{\mathcal{E}}(\text{sq}(\mathcal{X}_{:,k,:}))_{k=1}^p = \sum_{k=1}^p \mathbf{E}_k \otimes \text{sq}(\mathcal{X}_{:,k,:}).$$

It is easy to verify that $\mathcal{M}_{\mathcal{E}}[\mathcal{T}_{\mathcal{E}}[\mathbf{A}]] = \mathbf{A}$ since

$$\mathcal{M}_{\mathcal{E}}[\mathcal{T}_{\mathcal{E}}[\mathbf{A}]] = \sum_{k=1}^p \mathbf{E}_k \otimes \text{sq}(\sqrt{\eta_k} \text{twist}(\mathbf{A}_k)) = \sum_{k=1}^p \mathbf{E}_k \otimes (\sqrt{\eta_k} \mathbf{A}_k) = \mathbf{A}.$$

3.1. Strategy. Given \mathbf{A} and the new mapping scheme, we are now ready to present our approach to generating our matrix approximations. The steps are outlined in Algorithm 1. It is important to note that this approach is applicable for any choice of tensor decomposition and corresponding tensor approximation strategy. However, the practical utility is tied to the structure obtained after applying the tensor-to-matrix mapping in the last step. Thus, in section 4, we consider some specific tensor approximation strategies and delve into the resulting matrix structure.

Algorithm 1 Tensor-based matrix approximation.

1. Determine \mathcal{E}, \mathcal{A} from \mathbf{A} .
 2. Define $\mathcal{X} := \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$.
 3. Compute a tensor approximation to \mathcal{X} , $\mathcal{T} := \hat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]$.
 4. Define the matrix approximation $\hat{\mathbf{A}} := \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = \mathcal{M}_{\mathcal{E}}[\hat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]]$.
-

For our approach to be successful, we need to know how to control the error in the matrix approximation. Fortunately, we can directly connect the error in the tensor

approximation with the error in the matrix approximation, as we now show. This result is general in that it does not depend on the specific tensor structure that is used, or the algorithm used for low tensor rank approximation.

LEMMA 4. Let $\mathbf{A} \in \mathbb{R}^{(\ell m) \times (qn)}$ and let $\mathcal{T}_{\mathcal{E}}[\cdot]$ and $\mathcal{M}_{\mathcal{E}}[\cdot]$ be the tensor-to-matrix and matrix-to-tensor mappings, respectively. Let $\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}] \approx \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ be a tensor approximation computed using any appropriate method. Then the error in the matrix approximation satisfies

$$\|\mathbf{A} - \mathcal{M}_{\mathcal{E}}[\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]]\|_F = \|\mathcal{T}_{\mathcal{E}}[\mathbf{A}] - \widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]\|_F.$$

Proof. For simplicity, write $\mathcal{X} = \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ and $\mathcal{J} = \widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]$. By the assumptions made earlier, there are p distinct, and nonoverlapping blocks and each block is repeated η_k times. Combining this insight with the properties of Kronecker products and Frobenius norms,

$$\begin{aligned} \|\mathbf{A} - \mathcal{M}_{\mathcal{E}}[\mathcal{J}]\|_F^2 &= \left\| \sum_{k=1}^p \mathbf{E}_k \otimes \text{sq}(\mathcal{X}_{:,k,:}) - \sum_{k=1}^p \mathbf{E}_k \otimes \text{sq}(\mathcal{J}_{:,k,:}) \right\|_F^2 \\ &= \sum_{k=1}^p \|\mathbf{E}_k \otimes (\text{sq}(\mathcal{X}_{:,k,:}) - \text{sq}(\mathcal{J}_{:,k,:}))\|_F^2 \\ &= \sum_{k=1}^p \|\mathbf{E}_k\|_F^2 \|\mathcal{X}_{:,k,:} - \mathcal{J}_{:,k,:}\|_F^2. \end{aligned}$$

Taking square roots, we have the desired result. \square

Remark 3.2. In the worst case, if there is no block structure, we have $p = \ell q$. In this case, we can write

$$\mathbf{A} = \sum_{j=1}^{\ell} \sum_{k=1}^q \mathbf{E}_{jk} \otimes \mathbf{A}_{jk},$$

where $\mathbf{E}_{jk} = \mathbf{e}_j \mathbf{e}_k^T \in \mathbb{R}^{\ell \times q}$, and $\mathbf{A}_{jk} \in \mathbb{R}^{m \times n}$ is the (j, k) th block matrix of \mathbf{A} . In this case, it may be more appropriate to define the tensor mapping $\mathcal{T}_{\mathcal{E}}[\mathbf{A}] \in \mathbb{R}^{m \times \ell \times q \times n}$ with lateral slices

$$\mathcal{T}_{\mathcal{E}}[\mathbf{A}]_{:,j,k,:} = \text{twist}(\mathbf{A}_{jk}).$$

The important point is that it may be more appropriate to represent it as a fourth order tensor, rather than a third order tensor. We will not consider this case here and leave it for future work.

Remark 3.3. The result in Lemma 4 is independent of the tensor approximation method that is used. In the next sections, we describe the specific structure that can be obtained when using CP or truncated-HOSVD expansions, but it is important to note that other options are possible, though we do not cover them all here.

4. Kronecker sum approximations using tensor decompositions. In the notation of the previous section, suppose we have a tensor $\mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ that is the result of mapping a structured matrix \mathbf{A} . We decompose (approximately) the tensor to obtain $\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]$, then we use the inverse mapping to get our matrix approximation. For simplicity, we write $\mathcal{X} := \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ and $\mathcal{J} := \widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]$. We will show that when the tensor approximations are available in either a CP or truncated HOSVD format, we obtain matrix approximations that involve a level of Kronecker structure. Indeed, using the right lens, we can express our matrix approximation as a sum of (structured) Kronecker products of matrices.

4.1. Matrix structure using CP. Suppose we are given the approximation to $\mathcal{X} \in \mathbb{R}^{m \times p \times n}$ in the Kruskal form for a third order tensor $\mathcal{T} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$. We observe the following for the corresponding tensor-to-matrix mapping.

THEOREM 5. *Let $\mathbf{A} \in \mathbb{R}^{(\ell m) \times (qn)}$ and let $\mathcal{X} := \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ be the mapped tensor. Suppose we approximate \mathcal{X} using $\mathcal{T} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$, where $\mathbf{X} \in \mathbb{R}^{m \times r}$, $\mathbf{Y} \in \mathbb{R}^{p \times r}$, $\mathbf{Z} \in \mathbb{R}^{n \times r}$. Then*

$$(7) \quad \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = (\mathbf{I}_{\ell} \otimes \mathbf{X}) \text{struct}_{\mathcal{E}}(\text{diag}(\mathbf{Y}_{k,:}))_{k=1}^p (\mathbf{I}_q \otimes \mathbf{Z}^{\top}).$$

The term in the middle of the factorization is block $\ell \times q$ with $r \times r$ diagonal blocks.

Proof. The Kruskal form of the tensor means that each lateral slice of \mathcal{T} must have the form $\mathbf{X} \text{diag}(\mathbf{Y}_{k,:}) \mathbf{Z}^{\top}$. Therefore,

$$\begin{aligned} \mathcal{M}_{\mathcal{E}}[\mathcal{T}] &= \sum_{k=1}^p \mathbf{E}_k \otimes \mathbf{X} \text{diag}(\mathbf{Y}_{k,:}) \mathbf{Z}^{\top} \\ &= \sum_{k=1}^p (\mathbf{I}_{\ell} \otimes \mathbf{X}) (\mathbf{E}_k \otimes \text{diag}(\mathbf{Y}_{k,:})) (\mathbf{I}_q \otimes \mathbf{Z}^{\top}) \\ &= (\mathbf{I}_{\ell} \otimes \mathbf{X}) \left(\sum_{k=1}^p \mathbf{E}_k \otimes \text{diag}(\mathbf{Y}_{k,:}) \right) (\mathbf{I}_q \otimes \mathbf{Z}^{\top}). \end{aligned}$$

The proof is completed by recognizing the middle term as a block-structured matrix. \square

Suppose the rank of \mathbf{Y} is $\rho \leq \min(p, r)$, and $\mathbf{Y} = \mathbf{F} \mathbf{G}^{\top}$ is a rank-revealing factorization with \mathbf{F} of size $p \times \rho$ and \mathbf{G}^{\top} of size $\rho \times r$. Then it follows we can also represent our approximation as a sum of structured Kronecker products, as the following result shows.

COROLLARY 6. *Given the assumptions and notation in Theorem 5, suppose we can factorize $\mathbf{Y} = \mathbf{F} \mathbf{G}^{\top}$ with $\mathbf{F} \in \mathbb{R}^{p \times \rho}$ and $\mathbf{G} \in \mathbb{R}^{r \times \rho}$, where $\rho := \text{rank}(\mathbf{Y}) \leq \min\{p, r\}$, then*

$$(8) \quad \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = \sum_{j=1}^{\rho} \text{struct}_{\mathcal{E}}(\mathbf{F}_{:,j})_{k=1}^p \otimes (\mathbf{X} \text{diag}(\mathbf{G}_{:,j}) \mathbf{Z}^{\top}),$$

is a representation as a sum of Kronecker products of matrices.

Proof. We have $\text{diag}(\mathbf{Y}_{k,:}) = \sum_{j=1}^{\rho} f_{kj} \text{diag}(\mathbf{G}_{:,j})$. Hence,

$$\text{struct}_{\mathcal{E}}(\text{diag}(\mathbf{Y}_{k,:}))_{k=1}^p = \sum_{j=1}^{\rho} \text{struct}_{\mathcal{E}}(\mathbf{F}_{:,j})_{k=1}^p \otimes \text{diag}(\mathbf{G}_{:,j}).$$

Putting this last line together with (7) gives (8). \square

The important point about (8) is that we now have a matrix approximation $\hat{\mathbf{A}} \approx \mathbf{A}$ as follows,

$$\hat{\mathbf{A}} = \sum_{j=1}^{\rho} \mathbf{C}_j \otimes \mathbf{D}_j,$$

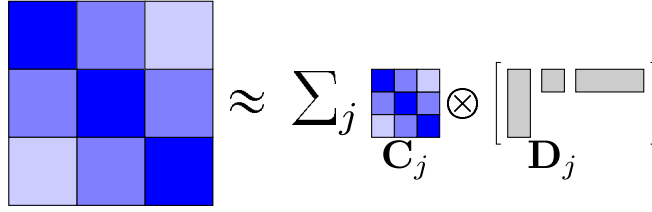


FIG. 1. A visualization of the Kronecker sum approximation of the structured matrix.

where $\mathbf{C}_j = \text{struct}_{\mathcal{E}}(\mathbf{F}_{:,j})_{j=1}^p$ and $\mathbf{D}_j = \mathbf{X} \text{diag}(\mathbf{G}_{:,j}) \mathbf{Z}^\top$ for $j = 1, \dots, \rho$. This decomposition into a sum of Kronecker products of matrices, is all the more noteworthy since the \mathbf{C}_j have the same *block structure* as the original matrix. See Figure 1 for an illustration.

Remark 4.1. If additional structure is imposed on the CP decomposition, this could result in additional structure for the corresponding matrix approximation. For example, if \mathbf{A} is nonnegative, so that the tensor is nonnegative, the CP factors should be nonnegative, implying the approximation will also have nonnegative entries.

4.2. Matrix structure using Tucker. Recall that we denote tensor $\mathfrak{X} := \mathcal{T}_{\mathcal{E}}[\mathbf{A}] \in \mathbb{R}^{m \times p \times n}$ is approximated by the tensor $\mathfrak{T} := \hat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]$ such that

$$\mathfrak{T} = \mathfrak{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

with core tensor $\mathfrak{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and the factor matrices $\mathbf{U} \in \mathbb{R}^{m \times r_1}$, $\mathbf{V} \in \mathbb{R}^{p \times r_2}$, and $\mathbf{W} \in \mathbb{R}^{n \times r_3}$ typically have orthonormal columns. There are several methods for computing low-rank approximations in the Tucker format. Some options include HOSVD, sequentially truncated HOSVD, and higher order orthogonal iteration. See [14, 9] for detailed reviews of the various techniques. For large-scale tensors, many randomized variants have been devised in recent years [4, 19, 1] to reduce the computational cost. In this next result, we show how to convert the approximation in the Tucker form to obtain an approximation as the sum of Kronecker products.

THEOREM 7. Consider the setting as in Theorem 5 but with the approximation in the Tucker format $\mathfrak{T} = \hat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}] = \mathfrak{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$, as defined above. Then

$$(9) \quad \mathcal{M}_{\mathcal{E}}[\mathfrak{T}] = \sum_{j=1}^{r_2} \text{struct}_{\mathcal{E}}(v_{kj})_{k=1}^p \otimes (\mathbf{U} \text{sq}(\mathfrak{G}_{:,j,:}) \mathbf{W}^\top).$$

Proof. From (3), the mode-3 unfolding of \mathfrak{T} has the expression $\mathfrak{T}_{(3)} = \mathbf{U} \mathfrak{G}_{(3)} (\mathbf{V}^\top \otimes \mathbf{W}^\top)$. It follows that each lateral slice of \mathfrak{T} is (see also (4))

$$\text{sq}(\mathfrak{T}_{:,k,:}) = \mathbf{U} \left(\sum_{j=1}^{r_2} v_{kj} \text{sq}(\mathfrak{G}_{:,j,:}) \right) \mathbf{W}^\top.$$

Applying the definition of the matrix-to-tensor mapping,

$$\begin{aligned} \mathcal{M}_{\mathcal{E}}[\hat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]] &= \text{struct}_{\mathcal{E}}(\text{sq}(\mathfrak{T}_{:,k,:}))_{k=1}^p = \sum_{k=1}^p \mathbf{E}_k \otimes \text{sq}(\mathfrak{T}_{:,k,:}) \\ &= \sum_{k=1}^p \mathbf{E}_k \otimes \left(\sum_{j=1}^{r_2} \text{struct}_{\mathcal{E}}(v_{kj})_{k=1}^p \otimes (\mathbf{U} \text{sq}(\mathfrak{G}_{:,j,:}) \mathbf{W}^\top) \right). \end{aligned}$$

Interchanging the order of the summations, we have

$$\begin{aligned}\mathcal{M}_{\mathcal{E}}[\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]] &= \sum_{j=1}^{r_2} \sum_{k=1}^p \mathbf{E}_k \otimes v_{kj} (\mathbf{U} \mathbf{s} \mathbf{q}(\mathcal{G}_{:,j,:}) \mathbf{W}^{\top}) \\ &= \sum_{j=1}^{r_2} \left(\sum_{k=1}^p \mathbf{E}_k \otimes v_{kj} \right) \otimes (\mathbf{U} \mathbf{s} \mathbf{q}(\mathcal{G}_{:,j,:}) \mathbf{W}^{\top}).\end{aligned}$$

The proof is complete by identifying $\sum_{k=1}^p \mathbf{E}_k \otimes v_{kj} = \mathbf{struct}_{\mathcal{E}}(v_{kj})_{k=1}^p$. \square

Using this theorem, we can compute the Kronecker product representation

$$\mathbf{A} \approx \sum_{j=1}^{r_2} \mathbf{C}_j \otimes \mathbf{D}_j,$$

where $\mathbf{C}_j = \mathbf{struct}_{\mathcal{E}}(v_{kj})_{k=1}^p$ and $\mathbf{D}_j = \mathbf{U} \mathbf{s} \mathbf{q}(\mathcal{G}_{:,j,:}) \mathbf{W}^{\top}$ for $j = 1, \dots, r_2$. It is interesting to note that the matrices \mathbf{C}_j preserve the original block structure of the matrix; for example, if \mathbf{A} is block Toeplitz, then the matrices $\{\mathbf{C}_j\}_{j=1}^{r_2}$ are Toeplitz. This is similar to what we observed in the CP case (see Figure 1).

We can take this analogy a step further. The CP decomposition $\mathcal{T} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$ can be written in Tucker format as $\mathcal{T} = [\mathcal{I}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]$, where \mathcal{I} is the identity tensor with $\mathcal{I}_{ijk} = 1$ if $i = j = k$ else 0. Note that in writing in the Tucker format we are not enforcing that the factor matrices have orthonormal columns. Applying Theorem 7 with the Tucker form of \mathcal{T} we obtain a result similar to Corollary 6.

4.3. Costs. We analyze the costs of storage and matvecs in the Kronecker sum format. In the discussion below, we assume that all the submatrices involved are dense. Additional savings may be possible if the submatrices are sparse. Furthermore, the costs involving matvecs may be further lowered if specific structure (Toeplitz, Hankel) can be leveraged; for example, one may be able to use Fast Fourier Transforms. In the naive case, we require $\mathcal{O}(mnp)$ units of storage and $\mathcal{O}(\ell mnq)$ flops for computing a matrix-vector product (matvec).

We assume that for the Tucker case, a rank (r_1, r_2, r_3) Tucker approximation has been computed. Similarly, for the CP decomposition, we assume a rank- r CP decomposition is computed; note that in this case r is not constrained to be smaller than $\min\{m, p, n\}$. Then the relevant costs are below:

1. **Storage:** This is split into two cases:

- (a) **CP:** This form requires $r(p + mr + nr + r)$ units of storage.
- (b) **Tucker:** Each \mathbf{C}_j requires p units of storage and each \mathbf{D}_j requires $mr_1 + nr_3 + r_1 r_3$ units of storage. Altogether this requires $r_2(p + mr_1 + nr_3 + r_1 r_3)$ units of storage.

2. **Matvec:** Once again, the analysis is split into two cases.

- (a) **CP:** Using the properties of Kronecker products, the cost of a matvec is $\mathcal{O}(r((nr + mr + r^2) \min\{\ell, q\} + nq\ell))$ flops.
- (b) **Tucker:** Using the properties of Kronecker products, the cost of a matvec is $\mathcal{O}(r_2((nr_3 + mr_1 + r_1 r_3) \min\{\ell, q\} + nq\ell))$ flops.

It is clear that the structure obtained using CP is efficient if $r \ll \min\{m, n\}$. To simplify the bounds for Tucker, let $r_t = \max_{1 \leq i \leq 3} (r_i)$, $m = n$, and $\ell = q$. Then the cost of the naive method is $\mathcal{O}(\ell^2 n^2)$ flops. The cost using Tucker is $\mathcal{O}(\ell r_t^2 m + m r_t \ell^2)$ flops. This is advantageous compared to the naive method if $r_t \ll m$.

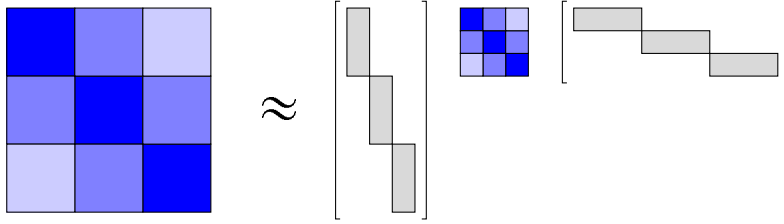


FIG. 2. Visualization of the block-structured matrix approximation.

5. Block-structured factorizations. In the previous section, we approximated \mathbf{A} as a sum of structured Kronecker products. In this subsection, we show how the tensor structure can be used to derive block-structured low-rank factorizations. Previous work in [27, 28, 33, 2] also considered block low-rank factorizations but our work is novel in two different ways: we consider the additional structure present in the matrix and we leverage tensor-based algorithms to exploit this structure.

CP decomposition. First, we show how to express the tensor approximation computed in Kruskal format as a block-structured factorization. We assume, once again that we have the CP decomposition $\mathcal{T} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$. Here, we are assuming that the number of terms in the CP approximation, r , is small relative to the problem dimension, in which case the following theorem can have important practical consequences.

THEOREM 8. *Consider the notation and assumptions in Theorem 5. Furthermore, let $\mathbf{X} = \mathbf{Q}_X \mathbf{R}_X$ and $\mathbf{Z} = \mathbf{Q}_Z \mathbf{R}_Z$ be the respective thin QR factorizations. Then,*

$$(10) \quad \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = (\mathbf{I}_{\ell} \otimes \mathbf{Q}_X) \text{struct}_{\mathcal{E}}(\mathbf{F}_k)_{k=1}^p (\mathbf{I}_q \otimes \mathbf{Q}_Z^{\top}),$$

where $\mathbf{F}_k = \mathbf{R}_X \text{diag}(\mathbf{Y}_{k,:}) \mathbf{R}_Z^{\top} \in \mathbb{R}^{r \times r}$ for $k = 1, \dots, p$.

Proof. Each lateral slice of \mathcal{T} has the form $\mathbf{X} \text{diag}(\mathbf{Y}_{k,:}) \mathbf{Z}^{\top} = \mathbf{Q}_X \mathbf{F}_k \mathbf{Q}_Z^{\top}$. The rest of the proof follows the proof of Theorem 5. \square

In this representation, the outer terms have orthonormal columns of sizes $(\ell m) \times (\ell r)$ and $(r q) \times (n q)$, whereas the middle term has block rank structure of size $(\ell r) \times (q r)$. See Figure 2 for an illustration, assuming that r is relatively small.

Tucker format. Suppose we have the tensor approximation in the Tucker format $\mathcal{T} = [\mathcal{G}; \mathbf{U}, \mathbf{V}, \mathbf{W}]$. The block-structured factorization is evident from observing that the expression in Theorem 7 can be equivalently expressed as

$$(11) \quad \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = (\mathbf{I}_{\ell} \otimes \mathbf{U}) \text{struct}_{\mathcal{E}} \left(\sum_{j=1}^{\ell} v_{kj} \text{sq}(\mathcal{G}_{:,j,:}) \right)_{k=1}^p (\mathbf{I}_q \otimes \mathbf{W}^{\top}).$$

In this representation, the outer terms have orthonormal columns and have sizes $(\ell m) \times (\ell r_1)$ and $(r_3 q) \times (n q)$, whereas the middle term has block rank structure of size $(\ell r_1) \times (q r_3)$.

Supposing the block-structured matrix \mathbf{A} is symmetric positive definite (SPD) or symmetric positive semidefinite (SPSD), we want the approximation obtained using the tensor compression to also be SPD or SPSPD, respectively. We now show how to do this in the context of block-structured factorizations.

5.1. Preserving positive semidefiniteness. Let $\mathbf{A} \in \mathbb{R}^{n\ell \times n\ell}$ be SPSPD and let \mathcal{A} , \mathcal{W} , and $\mathcal{T}_{\mathcal{E}}[\mathbf{A}] = \mathbf{X} \in \mathbb{R}^{n \times \ell \times n}$ be defined as before. Note that since \mathbf{A} is symmetric we have $\ell = q$ and $m = n$, and its diagonal blocks are symmetric. Furthermore, each matrix in \mathcal{A} (i.e., the subblocks of \mathbf{A}) is either symmetric or its transpose is also contained in \mathcal{A} . For example, consider the block Toeplitz matrix from Example 3 in section 3; if the block Toeplitz matrix is symmetric, then this implies \mathbf{A}_1 is symmetric and $\mathbf{A}_{\ell+k-1} = \mathbf{A}_k^{\top}$ for $k = 2, \dots, \ell$. This simple observation has the following important consequence: the mode-1 and mode-3 unfoldings of $\mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ have the same range (see (1) and the discussion preceding it), and therefore, we can use the same factor matrix to compress across each of these modes. That is, we compute a two-sided compression of the form

$$\mathcal{T}_{\mathcal{E}}[\mathbf{A}] \approx \mathcal{T} = \mathcal{G} \times_1 \mathbf{U} \times_3 \mathbf{U}, \quad \mathcal{G} = \mathcal{T} \times_1 \mathbf{U}^{\top} \times_3 \mathbf{U}^{\top},$$

where $\mathbf{U} \in \mathbb{R}^{n \times r}$ has orthonormal columns. We can compute \mathbf{U} , for example, from the dominant r left singular vectors of either the mode-1 or mode-3 unfolding of $\mathcal{T}_{\mathcal{E}}[\mathbf{A}]$. The corresponding block-structured approximation to \mathbf{A} is

$$\mathbf{A} \approx \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = (\mathbf{I} \otimes \mathbf{U}) \text{struct}_{\mathcal{E}} (\mathbf{U}^{\top} \mathbf{A}_k \mathbf{U})_{k=1}^p (\mathbf{I} \otimes \mathbf{U}^{\top}).$$

A straightforward rearrangement shows that

$$\mathcal{M}_{\mathcal{E}}[\mathcal{T}] = (\mathbf{I} \otimes \mathbf{P}) \mathbf{A} (\mathbf{I} \otimes \mathbf{P}),$$

where $\mathbf{P} = \mathbf{U} \mathbf{U}^{\top}$ is an orthogonal projector, since \mathbf{U} has orthonormal columns. Since \mathbf{A} is SPSPD, it is easy to verify that its approximation $\mathcal{M}_{\mathcal{E}}[\mathcal{T}]$ is also SPSPD. Note that if \mathbf{A} is symmetric (but not necessarily SPSPD), we can use a Tucker compression of the form

$$\mathcal{T} = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{U}, \quad \mathcal{G} = \mathcal{T} \times_1 \mathbf{U}^{\top} \times_2 \mathbf{V}^{\top} \times_3 \mathbf{U}^{\top},$$

where \mathbf{V} has orthonormal columns. The resulting approximation $\mathcal{M}_{\mathcal{E}}[\mathcal{T}]$ is symmetric but does not preserve definiteness (if present in the original block matrix).

5.2. Costs. As in subsection 4.3, we analyze the costs of storing and computing matvecs using the block-structured format. We make the same assumptions as before. The relevant costs are below:

1. **Storage:** This is split into two cases:

- (a) **CP:** This form requires $mr + nr + pr^2$ units of storage.
- (b) **Tucker:** It costs mr_1 and nr_3 units of storage for \mathbf{U} and \mathbf{W} , respectively. It requires storing an additional p matrices of size $r_1 \times r_3$. Altogether the storage cost is $mr_1 + nr_3 + pr_1r_3$ units of storage.

2. **Matvec:** Once again, the analysis is split into two cases.

- (a) **CP:** The cost of a matvec is $\mathcal{O}(\ell mr + qnr + r^2 \ell q)$ flops.
- (b) **Tucker:** The cost of a matvec is $\mathcal{O}(\ell mr_1 + qnr_3 + r_1 r_3 \ell q)$ flops.

As before, it is clear that the structure obtained using CP is efficient if $r \ll \min\{m, n\}$. With the simplifying assumptions as before for the Tucker format ($r_t = \max_i(r_i)$, $m = n$, and $\ell = q$), the cost of block structure using the Tucker format is $\mathcal{O}(\ell m r_t + r_t^2 \ell^2)$ flops. This is efficient compared to the naive method if $r_t \ll m$.

5.3. Preserving positive definiteness. Suppose $\mathbf{T} \in \mathbb{R}^{n\ell \times n\ell}$ is SPD, and we want our block-structured approximation to also be SPD. Our approach is inspired by Method 2 in [34], which treats the diagonal blocks in a special way. Assume we can decompose \mathbf{T} as

$$\mathbf{T} = \mathbf{I}_\ell \otimes \mathbf{T}_0 + \sum_{k=1}^p \mathbf{E}_k \otimes \mathbf{T}_k,$$

where \mathbf{T}_0 is SPD. Such a decomposition is guaranteed since at least one diagonal block of \mathbf{T} is SPD. Let $\mathbf{T}_0 = \mathbf{L}\mathbf{L}^\top$ be its Cholesky factorization. By factoring out \mathbf{T}_0 from the diagonal blocks, we get

$$\mathbf{T} = (\mathbf{I} \otimes \mathbf{L}) (\mathbf{I} + \mathbf{A}) (\mathbf{I} \otimes \mathbf{L}^\top), \quad \mathbf{A} = \sum_{k=1}^p \mathbf{E}_k \otimes \mathbf{A}_k,$$

where $\mathbf{A}_k = \mathbf{L}^{-1} \mathbf{T}_k \mathbf{L}^{-\top}$ for $k = 1, \dots, p$. Since \mathbf{T} is SPD, so is $\mathbf{I} + \mathbf{A}$.

Assume that we define \mathcal{A} , \mathcal{E} , and $\mathcal{T}_\mathcal{E}[\mathbf{A}] = \mathcal{X} \in \mathbb{R}^{n \times \ell \times n}$ as before. Once again, we consider the tensor approximation along modes 1 and 3:

$$\widehat{\mathcal{T}}_\mathcal{E}[\mathbf{A}] = \mathcal{T} = \mathcal{G} \times_1 \mathbf{U} \times_3 \mathbf{U}, \quad \mathcal{G} = \mathcal{T} \times_1 \mathbf{U}^\top \times_3 \mathbf{U}^\top.$$

Therefore, this gives the matrix approximation

$$\mathcal{M}_\mathcal{E}[\mathcal{T}] = (\mathbf{I} \otimes \mathbf{U}) \text{struct}_\mathcal{E} (\mathbf{U}^\top \mathbf{A}_k \mathbf{U})_{k=1}^p (\mathbf{I} \otimes \mathbf{U}^\top).$$

Finally, we can then approximate \mathbf{T} as

$$\mathbf{T} \approx \widehat{\mathbf{T}} := (\mathbf{I} \otimes \mathbf{L}) (\mathbf{I} + \mathcal{M}_\mathcal{E}[\mathcal{T}]) (\mathbf{I} \otimes \mathbf{L}^\top).$$

In the next theorem, we show that the approximation as constructed above is also SPD.

THEOREM 9. *Let $\mathbf{T} \in \mathbb{R}^{n\ell \times n\ell}$ be SPD. The approximation $\widehat{\mathbf{T}} \approx \mathbf{T}$ constructed, as before, is SPD.*

Proof. It suffices to show that $\mathbf{I} + \mathcal{M}_\mathcal{E}[\mathcal{T}]$ is SPD. Define the orthogonal projectors $\mathbf{\Pi} = \mathbf{I} \otimes \mathbf{P}$, where $\mathbf{P} = \mathbf{U}\mathbf{U}^\top$, and $\mathbf{\Pi}_\perp = \mathbf{I} \otimes \mathbf{P}_\perp$, where $\mathbf{P}_\perp = \mathbf{I} - \mathbf{P} = \mathbf{I} - \mathbf{U}\mathbf{U}^\top$. With this notation, we can write

$$\mathbf{I} + \mathcal{M}_\mathcal{E}[\mathcal{T}] = \mathbf{I} + \mathbf{\Pi} \mathbf{A} \mathbf{\Pi} = \mathbf{\Pi}_\perp + \mathbf{\Pi} (\mathbf{I} + \mathbf{A}) \mathbf{\Pi}.$$

Consider a nonzero vector $\mathbf{x} \in \mathbb{R}^{n\ell}$ and decompose it uniquely as $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, where $\mathbf{x}_1 \in \mathcal{R}(\mathbf{\Pi})$ and $\mathbf{x}_2 \in \mathcal{R}(\mathbf{\Pi}_\perp)$. The quadratic form simplifies as

$$\mathbf{x}^\top (\mathbf{I} + \mathcal{M}_\mathcal{E}[\mathcal{T}]) \mathbf{x} = \mathbf{x}_2^\top \mathbf{x}_2 + \mathbf{x}_1^\top (\mathbf{I} + \mathbf{A}) \mathbf{x}_1.$$

This is clearly nonnegative; furthermore, since $(\mathbf{I} + \mathbf{A})$ is SPD, at least one of the two terms in the summation is positive. Therefore, $\mathbf{I} + \mathcal{M}_\mathcal{E}[\mathcal{T}]$ is SPD. \square

It is interesting to note that we can rearrange the expression for $\widehat{\mathbf{T}}$ to obtain

$$\widehat{\mathbf{T}} = \mathbf{I} \otimes \mathbf{T}_0 + (\mathbf{I} \otimes \mathbf{L} \mathbf{P} \mathbf{L}^{-1}) \text{struct}_\mathcal{E} (\mathbf{T}_k)_{k=1}^p (\mathbf{I} \otimes \mathbf{L}^\top \mathbf{P} \mathbf{L}^{-\top}),$$

where $\mathbf{L} \mathbf{P} \mathbf{L}^{-1} = \mathbf{L} \mathbf{U} \mathbf{U}^\top \mathbf{L}^{-1}$ is an orthogonal projector with respect to the $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{T}_0^{-1}} = \mathbf{x}^\top \mathbf{T}_0^{-1} \mathbf{y}$ inner product.

6. Applications and numerical experiments. In this section, we demonstrate the utility of our formulation and the performance of the various algorithms proposed in the previous sections. The first application (subsection 6.1) we consider is accelerating the computation of the eigensystem realization algorithm, which involves large block Hankel matrices. The second application (subsection 6.2) arises from space-time covariance matrices and involves block Toeplitz matrices. In subsection 6.3 we obtain Kronecker sum approximations to several sparse test matrices from the SuiteSparse matrix collection.

6.1. Subspace system identification. Consider the linear time-invariant discrete dynamical system

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \\ \mathbf{y}_{k+1} &= \mathbf{C}\mathbf{x}_{k+1} + \mathbf{D}\mathbf{u}_{k+1}.\end{aligned}$$

Here $\mathbf{A} \in \mathbb{R}^{d \times d}$ is the state transition matrix, $\mathbf{B} \in \mathbb{R}^{d \times n}$ is the input matrix, $\mathbf{C} \in \mathbb{R}^{m \times d}$ is the output matrix, and $\mathbf{D} \in \mathbb{R}^{m \times n}$ is the feedthrough matrix. The goal in system identification is to recover the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ (up to a certain similarity transformation) governing the dynamics of the system from the inputs $\{\mathbf{u}_k\}$ and outputs $\{\mathbf{y}_k\}$. When the inputs are of the impulse type, a special algorithm known as the eigensystem realization algorithm (ERA) can be used for system identification [17, 11].

In this setting, we are given, as data, the *Markov parameters*

$$(12) \quad \mathbf{h}_k = \begin{cases} \mathbf{D}, & k = 0, \\ \mathbf{C}\mathbf{A}^{k-1}\mathbf{B}, & k > 0. \end{cases}$$

Recovering the system matrix \mathbf{D} is straightforward since it is the first Markov parameter. In what follows, we focus on recovering \mathbf{A} , \mathbf{B} , and \mathbf{C} . Then, a block Hankel matrix is formed with the Markov parameters such that there are s block rows and columns of size $m \times n$ each:

$$\mathbf{H}_s = \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \cdots & \mathbf{CA}^{s-1}\mathbf{B} \\ \mathbf{CAB} & \mathbf{CA}^2\mathbf{B} & \cdots & \mathbf{CA}^s\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{s-1}\mathbf{B} & \mathbf{CA}^s\mathbf{B} & \cdots & \mathbf{CA}^{2s-2}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{(ms) \times (ns)}.$$

The value of s might also be large in practice, with $s \approx \mathcal{O}(10^3 - 10^5)$. A review of the computational challenges involved in storing and factorizing \mathbf{H}_s are reviewed in [19].

We offer a brief derivation of ERA here. Assuming that the system is observable and controllable (see [32, Lemma 3.5] for more details), we have $\text{rank}(\mathbf{H}_s) = d$ and we can factorize \mathbf{H}_s as

$$\mathbf{H}_s = \underbrace{\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{s-1} \end{bmatrix}}_{\mathbf{O}_s} \underbrace{[\mathbf{B}, \mathbf{AB}, \dots, \mathbf{A}^{s-1}\mathbf{B}]}_{\mathbf{C}_s},$$

where \mathbf{O}_s is the observability matrix and \mathbf{C}_s is the controllability matrix, which both have rank n . The trick is to partition the observability matrix \mathbf{O}_s ,

$$\mathbf{O}_s^{(f)} \mathbf{A} = \mathbf{O}_s^{(b)},$$

where $\mathbf{O}_s^{(f)}$ is the leading $ms - m$ rows of \mathbf{O}_s and $\mathbf{O}_s^{(b)}$ contains rows $m + 1$ to ms of \mathbf{O}_s . We can recover the matrix \mathbf{A} (up to a similarity transformation) by solving this problem in the least squares sense. That is, we write

$$\tilde{\mathbf{A}} = [\mathbf{O}_s^{(f)}]^\dagger \mathbf{O}_s^{(b)}.$$

In practice, the observability and controllability matrices are not explicitly available, but we can instead work with a factorization of the form $\mathbf{H}_s = \mathbf{\Theta}_s \mathbf{\Gamma}_s^\top$, where $\mathbf{\Theta}_s$ has the same dimensions as \mathbf{O}_s . For example, we can construct such a factorization using the reduced SVD. Furthermore, if we desire model reduction in addition to system identification, we work with the SVD of \mathbf{H}_s truncated to target rank $r < d$.

Once the state transition matrix \mathbf{A} is recovered, we can also recover the matrices \mathbf{B} and \mathbf{C} . The details are available in other sources (e.g., [32, 19]). The step involving the computation of the SVD of \mathbf{H}_s dominates the overall computational cost, since it is cubic in the number of time points s . Other methods to mitigate this computational cost are available in [19, 16], etc. In this section, we exploit the tensor-based computational framework for computing structured matrix approximations developed in section 4.

Matrix-to-tensor mapping. We now show how to map the structured block matrix \mathbf{H}_s into tensor form for compression using the tensor techniques. Let η_k be defined as

$$\eta_k = \begin{cases} k, & 1 \leq k \leq s, \\ 2s - k, & s < k < 2s - 1. \end{cases}$$

Then each \mathbf{E}_k is a Hankel matrix with 0's everywhere except for indices where $i + j - 1 = k$, where it takes the value $\sqrt{\eta_k}$ and

$$\mathcal{E} = (\mathbf{E}_1, \dots, \mathbf{E}_{2s-1}), \quad \mathcal{A} = (\mathbf{h}_1, \dots, \mathbf{h}_{2s-1}).$$

Therefore, the tensor $\mathcal{T}_{\mathcal{E}}[\mathbf{H}_s]_{:k:} = \sqrt{\eta_k} \text{twist}(\mathbf{h}_k)$ for $k = 1, \dots, 2s - 1$.

We compute a Tucker decomposition of multirank (r_1, r_2, r_3) of the form

$$\mathcal{T}_{\mathcal{E}}[\mathbf{H}_s] \approx \mathcal{T} = \hat{\mathcal{T}}_{\mathcal{E}}[\mathbf{H}_s] = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W},$$

where $\mathcal{G} = \mathcal{T} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top$. This low-rank decomposition is computed using the HOSVD algorithm. In this example problem the number of outputs and inputs are $m = 155$, $n = 50$, respectively, and we take $s = 100$. The size of the tensor $\mathcal{T}_{\mathcal{W}}[\mathbf{H}_s]$ is, therefore, $155 \times 199 \times 50$. In Figure 3, we plot the singular value decay in each mode unfolding. We observe that the singular values decay in all three mode unfoldings, with a very sharp drop in mode-2. We can then write the approximate block-structured matrix as

$$\hat{\mathbf{H}}_s = (\mathbf{I} \otimes \mathbf{U}) \text{struct}_{\mathcal{E}} \left\{ \sum_{j=1}^{r_2} v_{kj} \text{sq}(\mathcal{G}_{:,j,:}) \right\}_{k=1}^{2s-1} (\mathbf{I} \otimes \mathbf{W}^\top).$$

To perform system identification, it is sufficient to work with the SVD of

$$\hat{\mathbf{H}}_s^{\text{Tucker}} = \text{struct}_{\mathcal{E}} \left(\sum_{j=1}^{r_2} v_{kj} \text{sq}(\mathcal{G}_{:,j,:}) \right)_{k=1}^{2s-1},$$

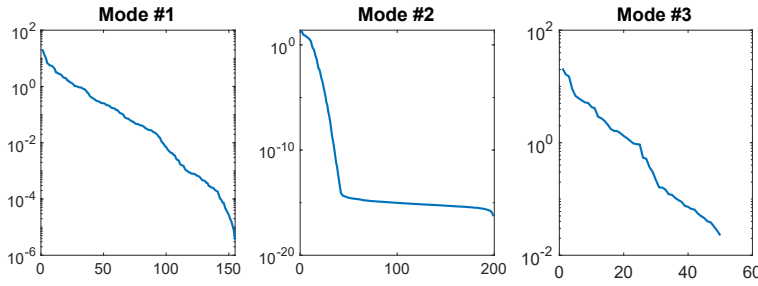


FIG. 3. This plot shows the singular value decay of each mode unfolding of $\mathcal{T}[\mathbf{H}_s]$, corresponding to the power system application problem.

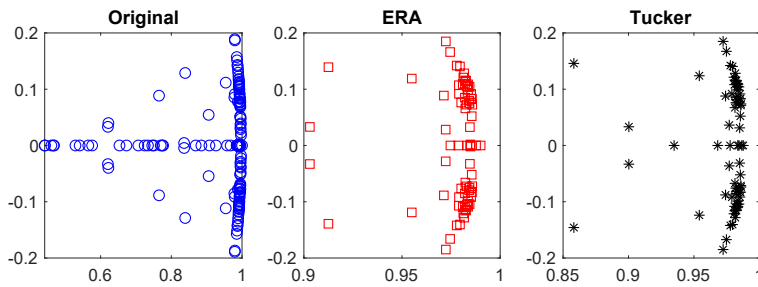


FIG. 4. Left displays the eigenvalues of the original system; middle displays the eigenvalues using ERA; and right displays the realized eigenvalues computed using our TuckerRandERA approach.

since the SVD of $\hat{\mathbf{H}}_s$ follows in a straightforward manner. Since $\hat{\mathbf{H}}_s^{\text{Tucker}}$ is still in the block Hankel format, we can take advantage of its structure to efficiently compute its dominant singular values. In particular, we apply the randomized ERA algorithm [19, section 3.1] to $\hat{\mathbf{H}}_s^{\text{Tucker}}$. We call this the ‘TuckerRandERA’ approach. This has a computational cost $\mathcal{O}(r_1 r_3 s \log_2 s)$ compared to $\mathcal{O}(mn \min\{m, n\} s^3)$ using the naive approach. Similarly, the storage cost of the Tucker approach is $\mathcal{O}(r_1 r_3 s)$ but for the naive approach this is $\mathcal{O}(mns^2)$ (since the entire matrix \mathbf{H}_s needs to be stored).

In the following computational experiment, we pick the target rank $(56, 30, 30)$ based on the decay of the singular values of the mode unfolding (see Figure 3). This is a significant compression considering the size of the original tensor is $155 \times 199 \times 50$. Based on [19] the target rank for the ERA was chosen to be $r = 75$. In Figure 4, we plot the eigenvalues of \mathbf{A} . The left panel displays the eigenvalues of the original system, the middle panel displays the eigenvalues using ERA, and the right panel displays the realized eigenvalues computed using the TuckerRandERA approach. As is seen from the plot, the eigenvalues computed both using the ERA and the TuckerRandERA approach are in good agreement, from a visual perspective. The Hausdorff distance between the eigenvalues identified using ERA and TuckerRandERA is approximately 5.47×10^{-2} (see section 5 in [19] for an explanation of this measure of accuracy). The accuracy improves significantly if a larger value of s is used. The run time for ERA was approximately 68 seconds, and the time for the TuckerRandERA approach is approximately 0.79 seconds (including precomputation of the Tucker decomposition and averaged over three runs). Both the accuracy and the computational benefits improve significantly if a larger s is used.

Relation to TERA. A related approach proposed in the literature is tangential interpolation ERA (TERA) [16]. In our notation, the TERA approximation to \mathbf{H}_s is

$$\mathbf{H}_s^{\text{TERA}} = (\mathbf{I} \otimes \mathbf{U}) \text{struct}_{\mathcal{E}} \left(\tilde{\mathbf{h}}_k \right)_{k=1}^{2s-1} (\mathbf{I} \otimes \mathbf{W}^\top),$$

where $\tilde{\mathbf{h}}_k = \mathbf{U}^\top \mathbf{h}_k \mathbf{W} = \mathbf{U}^\top \mathbf{C} \mathbf{A}^{k-1} \mathbf{B} \mathbf{W}$ are the projected Markov parameters. We construct the tensor $\mathcal{T}[\mathbf{H}_s]_{:k:} = \text{twist}(\mathbf{h}_k)$ with the Markov parameters as the lateral slices. Note that this tensor is constructed in a slightly different way than in section 3, where the lateral slices are weighted by the frequency with which they appear in the block matrix. It is relatively straightforward to show that this corresponds to a Tucker-2 approximation of $\mathcal{T}[\mathbf{H}_s]$ of the form

$$\mathcal{T}[\mathbf{H}_s] \approx \hat{\mathcal{T}}[\mathbf{H}_s] = \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{I} \times_3 \mathbf{W},$$

where $\mathcal{G} = \mathcal{T}[\mathbf{H}_s] \times_1 \mathbf{U}^\top \times_3 \mathbf{W}^\top$.

The following facts are worth pointing out regarding this connection to TERA. First, the structure of TERA is similar to that obtained using the Tucker approach described in this section if $\mathbf{V} = \mathbf{I}$; that is, we compress along modes 1 and 3 only. Second, the computational cost of TERA and using the Tucker approach is comparable; since both algorithms preserve the block Hankel structure, they can be accelerated using the randomized ERA approach. Third, using the Tucker decomposition across all three modes enables us to use additional structure that the TERA is presumably, neglecting. Finally, the TERA approximation treats all the Markov parameters “equally” and does not consider the frequency of occurrence in the block matrix \mathbf{H}_s .

6.2. Space-time covariance matrices. In geostatistical applications, it is important to represent the unknown quantities that vary over space and time by random processes [8]. Gaussian random processes are often used in this context and are completely characterized by the mean and covariance functions. Consider the Gaussian random field $Z(\mathbf{x}, t)$; assuming that it has a zero mean and finite second moment, we can represent the covariance function between two points in space-time as

$$C((\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2)) = \text{Cov}(Z(\mathbf{x}_1, t_1), Z(\mathbf{x}_2, t_2)).$$

Depending on the the type of covariance function used and the number of points in space and time (e.g., nonseparable covariance function), the resulting space-time covariance matrices can be large, dense, and infeasible both from a storage and computational perspective. An accurate and efficient approximation of the resulting space-time covariance matrix is highly desirable in many applications. We consider the class of covariance matrices that are weakly stationary and isotropic, so that

$$C((\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2)) = \varphi(\|\mathbf{x}_1 - \mathbf{x}_2\|_2, |t_1 - t_2|).$$

Let $\{\mathbf{x}_j\}_{j=1}^N$ be a set of points in space and $\{t_j\}_{j=1}^T$ be a set of points in time. The points in space are allowed to be unstructured, but the points in time are considered to be equidistant. Consider the space-time covariance matrix $\mathbf{C} \in \mathbb{R}^{(NT) \times (NT)}$ which is block Toeplitz,

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_T \\ \mathbf{C}_2 & \mathbf{C}_1 & \cdots & \mathbf{C}_{T-1} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{C}_T & \cdots & \mathbf{C}_2 & \mathbf{C}_1 \end{bmatrix} \in \mathbb{R}^{(NT) \times (NT)},$$

where the blocks $\mathbf{C}_i \in \mathbb{R}^{N \times N}$ have the entries

$$[\mathbf{C}_i]_{jk} = \varphi(\|\mathbf{x}_j - \mathbf{x}_k\|_2, |t_1 - t_i|), \quad j, k = 1, \dots, N.$$

We consider the block low-rank approximation as proposed in section 5; however, since covariance matrices are SP(S)D, when constructing an approximation it is important to ensure that it preserves definiteness.

The mapping to tensors has already been described in Example 3 of section 3 and we do not repeat it here. Note that $\ell = q = T$ and $m = n = N$. To construct the low-rank approximation, we follow the approach in section 5.1, and compute the rank- (r, T, r) Tucker decomposition of $\mathcal{T}_{\mathcal{E}}[\mathbf{C}] \in \mathbb{R}^{N \times T \times N}$ to obtain

$$\mathcal{T}_{\mathcal{E}}[\mathbf{C}] \approx \mathcal{T} := \mathcal{G} \times_1 \mathbf{U} \times_3 \mathbf{U}.$$

To compute this decomposition, because of the symmetry of the problem, we only need to compute the left singular vectors $\mathbf{U} \in \mathbb{R}^{N \times r}$ of the mode-1 unfolding of $\mathcal{T}_{\mathcal{E}}[\mathbf{C}]$ (corresponding to the r dominant singular values) and then compute the core tensor $\mathcal{G} := \mathcal{T}_{\mathcal{E}}[\mathbf{C}] \times_1 \mathbf{U}^\top \times_3 \mathbf{U}^\top$. If the SVD is used to compute \mathbf{U} , then the computational cost is $\mathcal{O}(TN^3)$, but can be lowered to $\mathcal{O}(rN^2T)$ using randomized SVD [10]. This cost can be further lowered to $\mathcal{O}(rNT \log N)$ using the \mathcal{H} -matrix approach [18], but we do not pursue this here. We then obtain the block-structured approximation to \mathbf{C} as

$$\hat{\mathbf{C}} = \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = (\mathbf{I} \otimes \mathbf{U}) \text{struct}_{\mathcal{E}} (\mathbf{U}^\top \mathbf{C}_k \mathbf{U})_{k=1}^T (\mathbf{I} \otimes \mathbf{U}^\top).$$

The block-structured approximation $\hat{\mathbf{C}}$ is efficient both in terms of storage and computation. The cost of storing $\hat{\mathbf{C}}$ implicitly is $\mathcal{O}(rN + r^2T)$. In contrast, the original matrix has the storage cost $\mathcal{O}(N^2T)$ if only the blocks \mathbf{C}_j are stored. The cost of a matvec with $\hat{\mathbf{C}}$ is $\mathcal{O}(rNT + r^2T^2)$, whereas a matvec with \mathbf{C} can be computed in $\mathcal{O}(N^2T^2)$. Both of these costs can be lowered, if we exploit the block Toeplitz structure; in particular, we can reduce the factor T^2 to $T \log T$.

In our first experiment, we check the accuracy of the block-structured approximation. We pick the covariance kernel corresponding to the function

$$\varphi(r, \tau) = \exp\left(-((r/90)^2 + (\tau/0.5)^2)\right).$$

We divide the world map into a $4^\circ \times 4^\circ$ grid, the points \mathbf{x}_j correspond to the land masses. In this particular instance, we have $N = 1351$ points and we choose $T = 30$ evenly spaced time points in the interval $[0, 1]$. To check the accuracy of the compression, we use the following relative error:

$$\text{relerr} = \frac{|\text{trace}(\mathbf{C}) - \text{trace}(\hat{\mathbf{C}})|}{\text{trace}(\mathbf{C})}.$$

Since \mathbf{C} cannot be formed explicitly, this metric is beneficial since it is easy to compute. Observe that $\text{trace}(\mathbf{C}) = NT$ and $\text{trace}(\hat{\mathbf{C}}) = T \text{trace}(\mathbf{U}^\top \mathbf{C}_1 \mathbf{U})$. When $r = 30$, we obtain the relative error $\text{relerr} \approx 3.35 \times 10^{-4}$, but the compression ratio

$$\text{ratio}_{\text{storage}} = \frac{r^2T + Nr}{N^2T} \approx 0.002.$$

In other words, at only a fraction of the storage costs, we obtain an approximation to \mathbf{C} that is reasonably accurate. In Figure 5, we plot samples from $\mathcal{N}(\mathbf{0}, \hat{\mathbf{C}} + \delta \mathbf{I})$, where $\delta = 10^{-8}$ is known as the nugget parameter added to ensure positive definiteness. The samples were computed using the Lanczos approach [5].

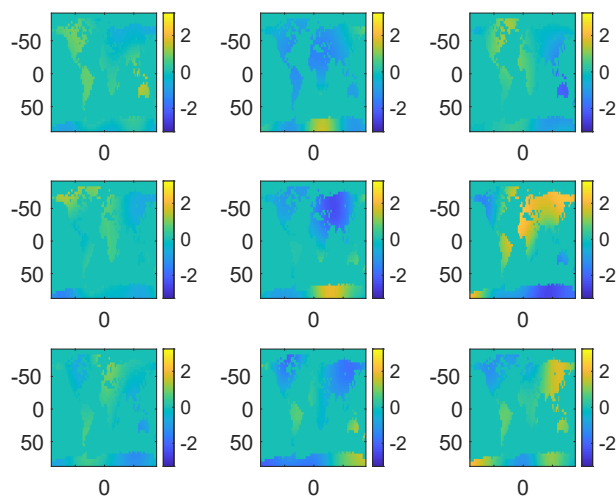


FIG. 5. Samples from the distribution $\mathcal{N}(\mathbf{0}, \hat{\mathbf{C}} + \delta \mathbf{I})$. The rows correspond to the time indices 1, 5, 15 and the columns correspond to three different samples.

6.3. Test matrices from SuiteSparse collection. In this example, we consider a set of sparse block-tridiagonal matrices from the SuiteSparse collection of matrices [15]. The tuples \mathcal{A}, \mathcal{E} are described in Example 2 of section 3, so we will not repeat the matrix-to-tensor mappings. Note that $\ell = q$, $m = n$, and $p = 3\ell - 2$. For each matrix \mathbf{A} , we produce a Kronecker sum approximation by first computing a compressed representation of the form

$$\mathcal{T} = \mathcal{G} \times_2 \mathbf{V}, \quad \mathcal{G} = \mathcal{X} \times_2 \mathbf{V}^\top,$$

where $\mathcal{X} = \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ and $\mathbf{V} \in \mathbb{R}^{p \times r}$ so that $\mathcal{G} \in \mathbb{R}^{n \times r \times n}$. Here r is the target rank of the compression. We chose to compress along mode-2 since for each example, the singular values of the mode-2 unfolding decayed rapidly. The matrix \mathbf{V} is computed using the first r left singular vectors of $\mathcal{X}_{(2)}$.

The compressed representation is then converted to a Kronecker sum approximation of the form $\mathbf{A} \approx \hat{\mathbf{A}} = \sum_{j=1}^r \mathbf{C}_j \otimes \mathbf{D}_j$, where

$$\mathbf{C}_j = \text{struct}_{\mathcal{E}}(v_{kj})_{k=1}^p \in \mathbb{R}^{\ell \times \ell}, \quad \mathbf{D}_j = \text{sq}(\mathcal{G}_{:,j,:}) \in \mathbb{R}^{n \times n}.$$

Note here that \mathbf{C}_j is a tridiagonal matrix, so the resulting approximation $\hat{\mathbf{A}}$ is also block tridiagonal. Furthermore, note that for the chosen test problems, since the matrices \mathbf{A}_j for $j = 1, \dots, p$ are tridiagonal, the matrices \mathbf{D}_j for $j = 1, \dots, r$ are also tridiagonal matrices as such a structure will be preserved in the lateral slices of \mathcal{G} . This makes the resulting Kronecker sum approximation efficient to store, providing $r \leq p$.

In Table 2, we list the name of the matrix, the relevant problem dimensions, the target rank, and the resulting relative error in the Frobenius norm $\|\mathbf{A} - \hat{\mathbf{A}}\|_F / \|\mathbf{A}\|_F$. We also report the compression ratio

$$\text{ratio}_{\text{storage}} = \frac{pr + \text{nnz}(\mathcal{G})}{\text{nnz}(\mathbf{A})}.$$

TABLE 1
Summary of the notation used to map matrices to tensors.

Symbol	Dimensions	Description
$\mathcal{X} = \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$	$m \times p \times n$	Matrix mapped to tensor
$\mathbf{A} = \mathcal{M}_{\mathcal{E}}[\mathcal{X}]$	$(\ell m) \times (qn)$	Tensor mapped to matrix
$\mathcal{T} = \widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]$	$m \times p \times n$	Tensor approximation
$\widehat{\mathbf{A}} = \mathcal{M}_{\mathcal{E}}[\mathcal{T}]$	$(\ell m) \times (qn)$	Matrix approximation

TABLE 2
For each test problem, we report the name of the matrix, the number of block rows ℓ , the size of each block n , the target rank used, the relative error, and the compression ratio. For the *ecology1* test case, we used the leading principal submatrix of size 500000×500000 .

Name	ℓ	n	Target rank r	Relative error	Compression
pde2961	63	47	20	8.36×10^{-10}	0.4470
t2d`q4	99	99	5	2.93×10^{-15}	0.034
t2d`q9	99	99	5	2.93×10^{-15}	0.034
fv2	99	99	5	2.28×10^{-15}	0.034
chem`master1	201	201	5	1.79×10^{-15}	0.030
ecology1 (*)	500	1000	5	6.10×10^{-15}	0.009

For the test problem *ecology1*, we made two modifications to reduce the runtime of the problem: first we used the leading principal submatrix of size 500000×500000 ; second, to compute the matrix \mathbf{V} , we use a randomized approach [20]. That is, we compute \mathbf{V} to be the left singular vectors of the mode-2 unfolding of the tensor $\mathbf{Y} = \mathcal{X} \times_1 \mathbf{\Omega} \times_3 \mathbf{\Psi}$. Here $\mathbf{\Omega}, \mathbf{\Psi} \in \mathbb{R}^{n \times r}$ are chosen to be independent standard Gaussian random matrices. In Table 2, it is seen that in each case we obtain a Kronecker sum approximation that is accurate, efficient to compute with, and to store.

7. Multilevel block structure. The discovery of latent tensor structure in the block matrices can be extended to multiple levels of block structure. As a motivating example, consider a BTTB matrix. Under the current formulation, we would represent this as a three-dimensional tensor with Toeplitz matrices as lateral slices, but ignore the Toeplitz structure within each lateral slice. We now develop a formulation that is capable of handling structure on every level on which it exists.

7.1. Multilevel structured approximation. Let $\mathbf{A} \in \mathbb{R}^{M \times N}$ with L levels of block structure, where $M = m \left(\prod_{j=1}^L \ell_j \right)$ and $N = n \left(\prod_{k=1}^L q_k \right)$. That is, at level 1, the matrix \mathbf{A} has $\ell_1 \times q_1$ blocks of size $(M/\ell_1) \times (N/q_1)$; at level L , the matrices are of size $m \times n$. Note that in our model, the type of block structure (e.g., banded, Toeplitz, Hankel, etc.) at each level is fixed. However, the type of block structure can vary across levels.

Example 10. Consider a 4×4 block-pentadiagonal Toeplitz matrix block symmetry and each block is itself a 5×5 block-diagonal matrix such that each block on the block diagonal is an $m \times n$ matrix with no additional structure to exploit (see Figure 6). In this example, $L = 2$, $\ell_1 = q_1 = 4$, $\ell_2 = q_2 = 5$. The total dimension of the matrix is $m(4 \times 5) \times n(4 \times 5)$, but because of the outer block Toeplitz structure and symmetry, the matrix can be represented by storing a total of $15 = 3 \times 5$, $m \times n$ matrices. However, we can keep those $15 m \times n$ matrices in a fourth order, $m \times 3 \times 5 \times n$ tensor.

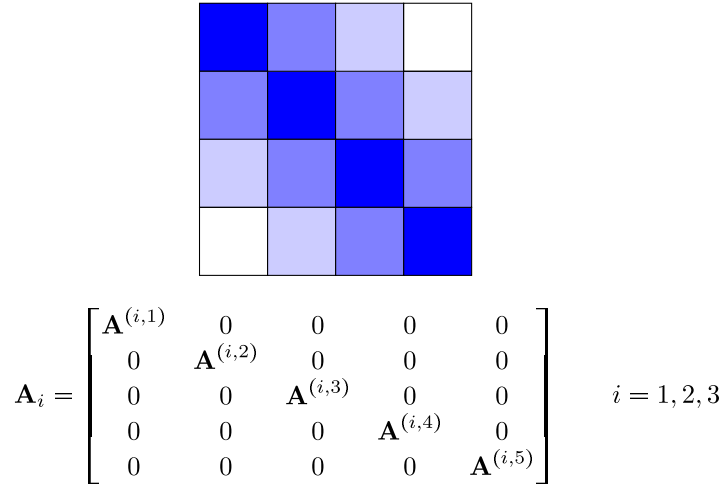


FIG. 6. Example where $L = 2$, $l_1 = q_1 = 4$, $l_2 = q_2 = 5$. The level 1 structure is block-symmetric Toeplitz (top row), the level 2 structure is block diagonal (bottom row). Each colored square on the left denotes a block diagonal matrix \mathbf{A}_i , where $i = 1, 2, 3$ represents the nonredundant information on the first level; each subblock $\mathbf{A}^{(i,j)}$, $j = 1, \dots, 5$ is $m \times n$.

7.2. Structured matrix-to-tensor mapping. In general, we can then write

$$\mathbf{A} = \sum_{i_1=1}^{p_1} \cdots \sum_{i_L=1}^{p_L} \mathbf{E}_{i_1}^{(1)} \otimes \cdots \otimes \mathbf{E}_{i_L}^{(L)} \otimes \sqrt{\eta_{i_1}^{(1)} \cdots \eta_{i_L}^{(L)}} \mathbf{A}^{(i_1, \dots, i_L)},$$

where $\mathbf{A}^{(i_1, \dots, i_L)}$ are the $m \times n$ nonredundant blocks at level L . A similar setup was also used in [26]. The matrices $\mathbf{E}_k^{(j)} \in \mathbb{R}^{\ell_j \times q_j}$, with entries in $\{0, \frac{1}{\sqrt{\eta_k^{(j)}}}\}$, represent the different mapping matrices at level j , where $1 \leq j \leq L$. At each level j , there are p_j such matrices, where $1 \leq p_j \leq \ell_j q_j$. Define the tuple of mapping matrices

$$\mathcal{E}^{(j)} = (\mathbf{E}_1^{(j)}, \dots, \mathbf{E}_{p_j}^{(j)}), \quad j = 1, \dots, L,$$

and $\mathcal{E} = (\mathcal{E}^{(1)}, \dots, \mathcal{E}^{(L)})$. For completeness, we note that in our running example, $p_1 = 3, p_2 = 5, \eta_1^{(1)} = 4, \eta_2^{(1)} = 3, \eta_3^{(1)} = 2$, and $\eta_{i_2}^{(2)} = 1, i_2 = 1, \dots, 5$.

We can express the blocks as lateral slices of a tensor $\mathcal{T}_{\mathcal{E}}[\mathbf{A}] \in \mathbb{R}^{m \times p_1 \times \dots \times p_L \times n}$ of order $(L+2)$, such that⁴

$$\mathcal{X}_{:,i_1, \dots, i_L, :} = \text{twist}(\mathbf{A}^{(i_1, \dots, i_L)}).$$

In our running example, this means that $\mathcal{X}_{i,j,k,\ell} = \mathbf{A}_{i,\ell}^{(j,k)}$. Consistent with the third order case, we express the mode- $(L+2)$ unfolding as⁵

$$\mathcal{X}_{(L+2)} = [\text{sq}(\mathcal{X}_{:,1, \dots, 1, :})^\top, \text{sq}(\mathcal{X}_{:,2, 1, \dots, 1, :})^\top, \dots, \text{sq}(\mathcal{X}_{:,p_1, \dots, p_L, :})^\top].$$

⁴We will have to choose an ordering for the indices. This description is independent of the ordering, as long as it is consistent. For example, one could use lexicographical, first running over i_1 , with all others fixed, then i_2 , etc.

⁵Consistent with ordering first in i_1 , then i_2 , etc.

The tensor-to-matrix mapping $\mathcal{M}_{\mathcal{E}} : \mathbb{R}^{m \times p_1 \times \cdots \times p_L \times n} \rightarrow \mathbb{R}^{M \times N}$ is

$$\mathcal{M}_{\mathcal{E}}[\mathcal{X}] = \sum_{i_1=1}^{p_1} \cdots \sum_{i_L=1}^{p_L} \mathbf{E}_{i_1}^{(1)} \otimes \cdots \otimes \mathbf{E}_{i_L}^{(L)} \otimes \text{sq}(\mathcal{X}_{:,i_1,\dots,i_L,:}).$$

It is easy to verify that $\mathcal{M}_{\mathcal{E}}[\mathcal{T}_{\mathcal{E}}[\mathbf{A}]] = \mathbf{A}$.

In the next section, we will show how this new framework encapsulates some special cases mentioned in the literature.

7.3. Approximation error. Suppose we construct a low-rank tensor approximation $\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]$. The next result generalizes Lemma 4 to the multilevel case and relates the error in the tensor approximation to the error in the block-matrix approximation in the Frobenius norm.

LEMMA 11. *Let \mathbf{A} , $\mathcal{M}_{\mathcal{E}}$, and $\mathcal{T}_{\mathcal{E}}$ be defined as in this section, and let $\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}] \approx \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$. The error in $\mathcal{M}_{\mathcal{E}}[\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]]$ satisfies*

$$\|\mathbf{A} - \mathcal{M}_{\mathcal{E}}[\widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]]\|_F = \|\mathcal{T}_{\mathcal{E}}[\mathbf{A}] - \widehat{\mathcal{T}}_{\mathcal{E}}[\mathbf{A}]\|_F.$$

Proof. The proof follows from an iterative application of the technique in the proof of Lemma 4. \square

Once again, we see that the error in the Frobenius norm in the matrix approximation is precisely equal to the error in the Frobenius norm in the tensor approximation and that the error is independent of the tensor compression technique or the format that is used. We now show how to convert the compressed representation in the Tucker format to obtain a sum of Kronecker products.

7.4. Structured matrix approximation. As in section 4, a low-rank approximation to $\mathcal{T}_{\mathcal{E}}[\mathbf{A}]$ can be computed either in the CP or Tucker format, using any appropriate low-rank approximation algorithm. Let us denote $\mathcal{X} = \mathcal{T}_{\mathcal{E}}[\mathbf{A}]$. Suppose we now use a Tucker approximation

$$(13) \quad \mathcal{X} \approx \mathcal{T} = \mathcal{G} \times_1 \mathbf{X} \begin{pmatrix} L+1 \\ \times \\ j=2 \end{pmatrix} \mathbf{Y}^{(j-1)} \times_{L+2} \mathbf{Z},$$

where the factor matrices $\mathbf{Y}^{(j)} \in \mathbb{R}^{p_j \times k_j}$, where $k_j \leq p_j$, the factor matrix $\mathbf{X} \in \mathbb{R}^{m \times k_x}$, and the factor matrix $\mathbf{Z} \in \mathbb{R}^{n \times k_z}$.

Then comparing the mode- $(L+2)$ unfoldings,

$$\mathcal{X}_{(L+2)} \approx \mathbf{Z} \mathcal{G}_{(L+2)} (\mathbf{Y}^{(L)})^\top \otimes \cdots \otimes (\mathbf{Y}^{(1)})^\top \otimes \mathbf{X}^\top.$$

It follows that

$$\text{sq}(\mathcal{X}_{:,i_1,\dots,i_L,:})^\top \approx \mathbf{Z} \sum_{j_1=1}^{k_1} \cdots \sum_{j_L=1}^{k_L} \mathbf{Y}_{i_L,j_L}^{(L)} \mathbf{Y}_{i_{L-1},j_{L-1}}^{(L-1)} \cdots \mathbf{Y}_{i_1,j_1}^{(1)} \text{sq}(\mathcal{G}_{:,j_1,\dots,j_L,:})^\top \mathbf{X}^\top.$$

This gives our matrix approximation as

$$(14) \quad \mathcal{M}_{\mathcal{E}}[\mathcal{T}] = (\mathbf{I} \otimes \mathbf{X}) \left(\sum_{i_1=1}^{p_1} \cdots \sum_{i_L=1}^{p_L} \bigotimes_{j=1}^L \mathbf{E}_{i_j}^{(j)} \otimes \mathbf{H}^{(i_1,\dots,i_L)} \right) (\mathbf{I} \otimes \mathbf{Z}^\top),$$

where

$$(15) \quad \mathbf{H}^{(i_1, \dots, i_L)} = \sum_{j_1=1}^{k_1} \cdots \sum_{j_L=1}^{k_L} \mathbf{Y}_{i_L, j_L}^{(L)} \mathbf{Y}_{i_{L-1}, j_{L-1}}^{(L-1)} \cdots \mathbf{Y}_{i_1, j_1}^{(1)} \mathbf{sq}(\mathcal{G}_{:, j_1, \dots, j_L, :}).$$

Inserting (15) into (14), and simplifying, we arrive at the approximation

$$\mathcal{M}_{\mathcal{E}}[\mathcal{T}] = \sum_{j_1=1}^{k_1} \cdots \sum_{j_L=1}^{k_L} \bigotimes_{t=1}^L \text{struct}_{\mathcal{E}(t)}(\mathbf{Y}_{:, j_t}^{(t)}) \otimes (\mathbf{X} \mathbf{sq}(\mathcal{G}_{:, j_1, \dots, j_L, :}) \mathbf{Z}^{\top}).$$

If the number of columns, k_x , in \mathbf{X} is less than m and/or the number of columns k_z , in \mathbf{Z} is less than n , this approximation obviously has lower rank than \mathbf{A} . Note that the slices $\mathbf{sq}(\mathcal{G}_{:, j_1, j_2, \dots, j_L, :})$ are of size $k_x \times k_z$.

7.5. Use case: Relation to previous literature. The framework we have described here subsumes, and in some cases enhances, some special cases that have been described elsewhere in the literature. For example, we can show that our method can be used to derive exactly the same optimal Kronecker decomposition as in [12] for BTTB matrices by taking $L = 2$, and $m = n = 1$ above.

In this section, we use our framework to improve on the results in [21]. In that paper, the authors consider a banded/block-banded triply Toeplitz matrix with Dirichlet boundary conditions on all three levels, whose “central” column is determined from a three-dimensional (3D) discrete point-spread function (PSF). Recognizing that the 3D PSF is a third order tensor, they use third order tensor decompositions to determine an approximate operator that is a (sum of) Kronecker products of Toeplitz matrices, and subsequently use the decomposition to find a preconditioner for the deblurring problem.

In their approach, the authors did not have a result that related the tensor approximation problem directly to the matrix approximation problem. We now show that using our framework, we can relate a (fifth order) tensor approximation problem comprised of entries from a 3D *weighted* PSF, and tie these approximation problems together in a natural way, thus improving on the approach in [21].

To simplify discussion, we will assume that the blurring operator is applied to a $K \times K \times K$ image, where K is odd. Thus the matrix is a $K^3 \times K^3$ blurring operator. The matrix is assumed Toeplitz at each level, and that the upper and lower bandwidth on all levels is $\frac{K-1}{2}$. Because of the 3-level banded Toeplitz structure, the central column is sufficient to specify every entry in the blurring operator. We can therefore consider the central column in the form of a third order tensor PSF, \mathcal{P} , such that the $(\frac{K+1}{2}, \frac{K+1}{2}, \frac{K+1}{2})$ entry corresponds to the entry of the central column on the main diagonal in \mathbf{A} . The authors of [21] propose to use a truncated HOSVD $\mathcal{P} \approx \sum_i \sum_j \sum_k \gamma_{i,j,k}(\mathbf{u}_i \circ \mathbf{v}_j \circ \mathbf{w}_k)$ to form $\mathbf{A} \approx \sum_{i,j,k} \mathbf{C}_k \otimes \mathbf{D}_j \otimes \mathbf{F}_i$, where each matrix is Toeplitz. For example, the Toeplitz matrix \mathbf{C}_k is formed using vector $\gamma_{:,j,k}^{1/3} \mathbf{w}_k$ as its central column, and so forth.

The tensor that is decomposed by the process above *is* the 3D PSF, with no weighting. This makes sense as a first choice since the entries in the PSF give the entries in the matrix, but it is not guaranteed that the matrix approximation error is equal to the tensor approximation error. Under our framework, we arrive at the optimally weighted PSF tensor naturally. In our framework applied to this problem

we have $p_1 = p_2 = p_3 = N$, $L = 3$, $m = n = 1$, so that our tensor is of size $1 \times N \times N \times N \times 1$. Specifically, our approach gives⁶

$$\mathcal{X} = [\mathcal{T}_{\mathcal{E}}[\mathbf{A}]]_{1,i_1,i_2,i_3,1} = \sqrt{\eta_{i_1}\eta_{i_2}\eta_{i_3}}\mathcal{P}_{i_3,i_2,i_1},$$

ensuring that our matrix approximation error is equal to the tensor approximation error in approximating \mathcal{X} , as given in Lemma 11. Since two modes are singleton dimensions, our fifth order tensor is effectively a third order tensor, and the matrices \mathbf{X}, \mathbf{Z} are just 1×1 (i.e., $\mathbf{X} = \mathbf{Z} = 1$) in (13). If $\mathcal{X} \approx \mathcal{T} = \mathcal{G} \times_1 \mathbf{1} \times_2 \mathbf{U} \times_3 \mathbf{V} \times_4 \mathbf{W} \times_5 \mathbf{1}$, then using (15), we have

$$\mathbf{H}^{(i_1,i_2,i_3)} = \sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \sum_{j_3=1}^{k_3} u_{i_1,j_1} v_{i_2,j_2} w_{i_3,j_3} \mathcal{G}_{1,j_1,j_2,j_3,1}.$$

We therefore arrive at the matrix approximation

$$\begin{aligned} \mathcal{M}_{\mathcal{E}}[\mathcal{T}] &= \sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \sum_{j_3=1}^{k_3} \mathcal{G}_{1,j_1,j_2,j_3,1} \mathbf{struct}_{\mathcal{E}^{(1)}}(\mathbf{U}_{:,j_1}) \\ &\quad \otimes \mathbf{struct}_{\mathcal{E}^{(2)}}(\mathbf{V}_{:,j_2}) \otimes \mathbf{struct}_{\mathcal{E}^{(3)}}(\mathbf{W}_{:,j_3}), \end{aligned}$$

and the scalars $\mathcal{G}_{1,j_1,j_2,j_3,1}$ can be distributed into the Kronecker factors as desired. Thus, while the method in [21] and our framework both give a sum of structured Kronecker products, our new approach has the optimal scaling baked into the process to ensure that the corresponding matrix approximation relates directly to the tensor approximation⁷.

8. Conclusions and future work. We have provided a powerful new tensor-based framework for computing structure-preserving matrix approximations, and demonstrated how the framework can be applied to great effect on some problems in scientific computing. Additionally, we showed that our framework is flexible enough that it can be used to describe optimal structured matrix approximation problems found in previous literature. The key design elements of our approach include the invertible matrix-to-tensor mapping followed by a suitable choice of tensor decomposition. For both the CP and Tucker formats, we showed how the features of the respective tensor approximations map back to the structure in the corresponding matrix approximation. We proved that the matrix approximations derived using our matrix-to-tensor mapping are as good as the corresponding tensor approximations, when the error is measured in the Frobenius norm. Moreover, we proved that it is possible to preserve symmetric positive definiteness in the matrix approximation.

Although we investigated the details given when using CP and Tucker decomposition, our framework permits the use of any suitable tensor decomposition—for example a tensor-train approach [23] or a nonnegativity constrained CP—in the intermediate stage. We are currently investigating the extension of the present approach to the case of dense matrices. Another possible area for investigation is whether or not the framework could be modified to handle the case of nonuniform block sizes on a given level.

⁶Note the swapping of the indices in the first and third modes.

⁷The difference in the index correspondence between the two methods has to do with the orientation of the tensor, where we reverse the first and third modes from their definition of \mathbf{P} .

REFERENCES

- [1] S. AHMADI-ASL, A. CICHOCKI, A. H. PHAN, I. OSELEDETS, S. ABUKHOVICH, AND T. TANAKA, *Randomized Algorithms for Computation of Tucker Decomposition and Higher Order SVD (HOSVD)*, IEEE Access, 9 (2021), pp. 28684–28706.
- [2] P. AMESTOY, A. BUTTARI, J.-Y. L'EXCELLENT, AND T. MARY, *On the complexity of the block low-rank multifrontal factorization*, SIAM J. Sci. Comput., 39 (2017), pp. A1710–A1740.
- [3] K. BATSELIER, W. YU, L. DANIEL, AND N. WONG, *Computing low-rank approximations of large-scale matrices with the tensor network randomized SVD*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1221–1244.
- [4] M. CHE AND Y. WEI, *Randomized algorithms for the approximations of Tucker and the Tensor Train decompositions*, Adv. Comput. Math., 45 (2019), pp. 395–428.
- [5] E. CHOW AND Y. SAAD, *Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions*, SIAM J. Sci. Comput., 36 (2014), pp. A588–A608.
- [6] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
- [7] C. GARVEY, C. MENG, AND J. G. NAGY, *Singular value decomposition approximation via Kronecker summations for imaging applications*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1836–1857.
- [8] T. GNEITING, M. G. GENTON, AND P. GUTTORP, *Geostatistical space-time models, stationarity, separability, and full symmetry*, in Statistical Methods for Spatio-Temporal Systems, Monogr. Statist. Appl. Probab. 107, CRC, Boca Raton, CA, 2006, pp. 151–176.
- [9] W. HACKBUSCH, *Tensor Spaces and Numerical Tensor Calculus*, Springer Ser. Comput. Math. 42, Springer, Heidelberg, 2012.
- [10] N. HALKO, P. G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, <https://doi.org/10.1137/090771806>.
- [11] J.-N. JUANG AND R. S. PAPPA, *An eigensystem realization algorithm for modal parameter identification and model reduction*, J. Guid. Control Dyn., 8 (1985), pp. 620–627.
- [12] J. KAMM AND J. G. NAGY, *Optimal Kronecker product approximation of block Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 155–172.
- [13] M. E. KILMER, K. BRAMAN, N. HAO, AND R. C. HOOVER, *Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 148–172.
- [14] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
- [15] S. P. KOLODZIEJ, M. AZNAVEH, M. BULLOCK, J. DAVID, T. A. DAVIS, M. HENDERSON, Y. HU, AND R. SANDSTROM, *The SuiteSparse matrix collection website interface*, J. Open Source Softw., 4 (2019), p. 1244.
- [16] B. KRAMER AND S. GUGERCIN, *Tangential interpolation-based eigensystem realization algorithm for MIMO systems*, Math. Comput. Model. Dyn. Syst., 22 (2016), pp. 282–306.
- [17] S.-Y. KUNG, *A new identification and model reduction algorithm via singular value decomposition*, in Proceedings 12th Asilomar Conf. on Circuits, Systems and Computers, Pacific Grove, CA, IEEE, New York, 1978, pp. 705–714.
- [18] A. LITVINENKO, R. KRIEMANN, M. G. GENTON, Y. SUN, AND D. E. KEYES, *HLIBCov: Parallel hierarchical matrix approximation of large covariance matrices and likelihoods with applications in parameter identification*, MethodsX, 7 (2020), 100600.
- [19] R. MINSTER, A. K. SAIBABA, J. KAR, AND A. CHAKRABORTY, *Efficient algorithms for eigensystem realization using randomized SVD*, SIAM J. Matrix Anal. Appl., 42 (2021), pp. 1045–1072, <https://doi.org/10.1137/20M1327616>.
- [20] R. MINSTER, A. K. SAIBABA, AND M. E. KILMER, *Randomized algorithms for low-rank tensor decompositions in the Tucker format*, SIAM J. Math. Data Sci., 2 (2020), pp. 189–215.
- [21] J. G. NAGY AND M. E. KILMER, *Kronecker product approximation for preconditioning in three-dimensional imaging applications*, IEEE Trans. Image Process., 15 (2006), pp. 604–613.
- [22] V. OLSHEVSKY, I. OSELEDETS, AND E. TYRTYSHNIKOV, *Tensor properties of multilevel Toeplitz and related matrices*, Linear Algebra Appl., 412 (2006), pp. 1–21.
- [23] I. OSELEDETS, E. TYRTYSHNIKOV, AND N. ZAMARASHKIN, *Tensor-train ranks for matrices and their inverses*, Comput. Methods Appl. Math., 11 (2011), pp. 394–403.
- [24] I. V. OSELEDETS, *Approximation of $2^d \times 2^d$ matrices using tensor decomposition*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2130–2145.
- [25] I. V. OSELEDETS AND S. V. DOLGOV, *Solution of linear systems and matrix inversion in the TT-format*, SIAM J. Sci. Comput., 34 (2012), pp. A2718–A2739.

- [26] J. PESTANA, *Preconditioners for symmetrized Toeplitz and multilevel Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 870–887.
- [27] B. SAVAS AND I. S. DHILLON, *Clustered low rank approximation of graphs in information science applications*, in Proceedings of the 2011 SIAM International Conference on Data Mining, SIAM, Philadelphia, 2011, pp. 164–175.
- [28] B. SAVAS AND I. S. DHILLON, *Clustered matrix approximation*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1531–1555.
- [29] Y. SUN, Y. GUO, C. LUO, J. TROPP, AND M. UDELL, *Low-rank Tucker approximation of a tensor from streaming data*, SIAM J. Math. Data Sci., 2 (2020), pp. 1123–1150.
- [30] L. R. TUCKER, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
- [31] C. F. VAN LOAN, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.
- [32] M. VERHAEGEN AND V. VERDULT, *Filtering and System Identification: A Least Squares Approach*, Cambridge University Press, Cambridge, 2007.
- [33] R. WANG, Y. LI, M. W. MAHONEY, AND E. DARVE, *Block basis factorization for scalable kernel evaluation*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1497–1526.
- [34] X. XING AND E. CHOW, *Preserving positive definiteness in hierarchically semiseparable matrix approximations*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 829–855.