Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Hierarchical Singular Value Decomposition of Tensors

(revised version: March 2010)

by

Lars Grasedyck

Preprint no.: 27 2009

HIERARCHICAL SINGULAR VALUE DECOMPOSITION OF TENSORS

LARS GRASEDYCK[†]

Abstract. We define the hierarchical singular value decomposition (SVD) for tensors of order $d \geq 2$. This hierarchical SVD has properties like the matrix SVD (and collapses to the SVD in $d = 2$), and we prove these. In particular, one can find low rank (almost) best approximations in a hierarchical format ($\mathcal{H}$-Tucker) which requires only $\mathcal{O}((d-1)k^3 + dnk)$ parameters, where d is the order of the tensor, n the size of the modes and k the (hierarchical) rank. The $\mathcal{H}$-Tucker format is a specialization of the Tucker format and it contains as a special case all (canonical) rank k tensors. Based on this new concept of a hierarchical SVD we present algorithms for hierarchical tensor calculations allowing for a rigorous error analysis. The complexity of the truncation (finding lower rank approximations to hierarchical rank k tensors) is in $\mathcal{O}((d-1)k^4 + dnk^2)$ and the attainable accuracy is just 2–3 digits less than machine precision.

Key words. Tensor, Tucker, hierarchical Tucker, high-dimensional, low rank, SVD

AMS subject classifications. 15A69, 90C06, 65K10

1. Introduction.

Several problems of practical interest in physical, chemical, biological or mathematical applications naturally lead to high-dimensional (multivariate) approximation problems and thus are essentially not tractable in a naive way when the dimension d grows beyond $d = 10$. Examples are partial differential equations with many stochastic parameters, computational chemistry computations, the multiparticle electronic Schrödinger equation etc. This is due to the fact that the computational complexity or error bounds *must* depend exponentially on the dimension parameter d, which is coined by Bellman the *curse of dimensionality*. In order to make the setting more concrete we consider a multivariate function

$$f : [0,1]^d \to \mathbb{R}$$

discretized by tensor basis functions

$$\phi_{(i_1,\dots,i_d)}(x_1,\dots,x_d) := \prod_{\mu=1}^{d} \phi_{i_\mu}(x_\mu), \quad \phi_{i_\mu} : [0,1] \to \mathbb{R}, \quad 1 \leq i_\mu \leq n_\mu, 1 \leq \mu \leq d :$$

$$f(x_1,\dots,x_d) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} c_{i_1,\dots,i_d} \phi_{(i_1,\dots,i_d)}(x_1,\dots,x_d).$$

The one-dimensional basis functions $\phi_{i_\mu}(x_\mu)$ could for example be higher order Lagrange polynomials, characteristic functions, higher order wavelets or any other set of basis functions for an n_μ-dimensional subspace of $\mathbb{R}^{[0,1]}$.

The total number N of basis functions scales exponentially in d as $N = \prod_{\mu=1}^{d} n_\mu$. One strategy to overcome this curse (in complexity) is to assume some sort of smoothness of the function or object to be approximated so that one can choose a subspace $\tilde{V}$ of

$$V = \mathrm{span}\{\phi_{(i_1,\dots,i_d)} \mid (i_1,\dots,i_d) \in \{1,\dots,n_1\} \times \cdots \times \{1,\dots,n_d\}\}.$$

[†]Max Planck Institute for Mathematics in the Sciences, Inselstr. 22-26, D-04103 Leipzig, Germany. Phone: ++49(0)3419959827, ++49(0)3419959752. FAX: ++49(0)3419959999. Email: lgr@mis.mpg.de

This leads to the sparse grids method [19, 8] which chooses (adaptively [9] or non-adaptively) combinations of basis functions. An alternative way to approximate the multivariate function f is to separate the variables, i.e. to seek for an approximation of the form

$$f(x_1,\dots,x_d) \approx \tilde f(x_1,\dots,x_d) = \sum_{i=1}^{k} \prod_{\mu=1}^{d} f_{\mu,i}(x_\mu)$$

where each of the univariate functions $f_{\mu,i}(x) : [0,1] \to \mathbb{R}$ is discretized by the full one-dimensional set of basis functions $\phi_{j_\mu}(x)$, $j_\mu = 1,\dots,n_\mu$. If the separation rank k is small compared to N, then this is an efficient data-sparse representation. However, whereas sparse grids define a linear space, the set of functions representable with separation rank k is not a linear space. In particular it is not closed with respect to addition (the rank increases) and thus a necessary basic operation is to *truncate* representations from larger to smaller rank:

$$\text{For given } f \in V \text{ find } \tilde f \in V \text{ of rank } k \text{ s.t.} \quad \|f - \tilde f\| \approx \inf_{v\in V, \mathrm{rank}(v)=k} \|f - v\|.$$

This approximation problem suffers from the following difficulties:

1. A minimizer $\tilde f$ does not necessarily exist (problem is ill-posed), cf. [5]. The corresponding minimizing sequence consists of factors with increasing norm (and leads to severe cancellation effects). This can easily be overcome by Lagrange multipliers or penalty terms involving the norm of the factors.
2. There are no known algorithms allowing for an a priori estimate of the truncation error, see e.g. [12] for an overview on tensor algorithms. This is a severe bottleneck, because even for model problems one cannot be sure to find approximations of almost optimal rank — despite the fact that one might be able to prove that such a low rank approximation exists.
3. The approximation problem is rather difficult to solve if one wants to obtain an accuracy suitable for standard numerical applications, see e.g. [2, 7, 15] for the state of the art of efficient algorithms.

Thus, for some cases it is known how to construct a low separation rank approximation with high accuracy and stable representation but in order to use this low rank format as a basic format in numerical algorithms one needs a reliable truncation procedure that can be used universally without tuning parameters.

A new kind of separation scheme was introduced by Hackbusch and Kühn [10] and is coined *hierarchical* low rank tensor format. This new format allows the representation of order d tensors with $(d-1)k^3 + k\sum_{\mu=1}^{d} n_\mu$ data, where k is the involved — implicitly defined — representation rank. A similar format has been presented by other groups: the tree Tucker and tensor train format [14, 13] as well as the sequential unfolding SVD [16]. To our best knowledge the first successful approach to a hierarchical format has been developed by Beck & Jäckle & Worth & Meyer [1] and Wang & Thoss[18] (these references were kindly pointed out to us by Christian Lubich and Michael Griebel). We refer to Section 5 for a more detailed comparison.

In this article we will define the hierarchical rank of a tensor by singular value decompositions (SVD). The hierarchical format is then characterized by a nestedness of subspaces that stem from the SVDs. We present a corresponding hierarchical SVD which has a similar property as the higher order SVD (HOSVD) by De Lathauwer et al. [4], namely that the best approximation up to a factor of $\sqrt{2d-3}$ is obtained via cutting off the hierarchical singular values. We then derive a truncation procedure for (1.) dense or unstructured tensors as well as (2.) those already given in hierarchical format. In both cases almost linear (optimal) complexity with respect to the number of input data is achieved, in the latter case the truncation is of complexity $\mathcal{O}((d-1)k^4 + k^2 \sum_{\mu=1}^{d} n_\mu)$. Finally, we present numerical examples that underline the attainable accuracy which is close to machine precision (roughly 10^{-13} in double precision arithmetic) and apply the truncation for hierarchical tensors of order $d = 1,000,000$.

2. Tucker Format. NOTATION 2.1 (Index set). *Let $d \in \mathbb{N}$ and $n_1, \ldots, n_d \in \mathbb{N}$. We consider tensors as vectors over product index sets. For this purpose we introduce the d-fold product index set*

$$\mathcal{I} := \mathcal{I}_1 \times \cdots \times \mathcal{I}_d, \qquad \mathcal{I}_\mu := \{1, \ldots, n_\mu\}, \quad (\mu \in \{1, \ldots, d\}).$$

The order of the index sets can be important, but since it will always be clear which index belongs to which index set we will treat them without specifying the order. If the ordering becomes important it will be mentioned.

DEFINITION 2.2 (Mode, matricization, fibre). *Let $A \in \mathbb{R}^{\mathcal{I}}$. The dimension directions $\mu = 1, \ldots, d$ are called the* modes*. Let $\mu \in \{1, \ldots, d\}$. We define the index set*

$$\mathcal{I}^{(\mu)} := \mathcal{I}_1 \times \cdots \times \mathcal{I}_{\mu-1} \times \mathcal{I}_{\mu+1} \times \cdots \times \mathcal{I}_d$$

and the corresponding μ-mode matricization *by*

$$\mathcal{M}_\mu : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{I}_\mu \times \mathcal{I}^{(\mu)}}, \qquad (\mathcal{M}_\mu(A))_{i_\mu,(i_1,\ldots,i_{\mu-1},i_{\mu+1},\ldots,i_d)} := A_{(i_1,\ldots,i_d)}.$$

We use the short notation

$$A^{(\mu)} := \mathcal{M}_\mu(A)$$

and call this the μ-mode matricization *of A. The columns of $A^{(\mu)}$ define the μ-mode* fibres *of A.*

The μ-mode matricization $A^{(\mu)}$ is in one-to-one correspondence with the tensor A. The vector 2-norm $\|A\|_2$ corresponds to the matrix Frobenius norm: $\|A^{(\mu)}\|_F = \|A\|_2$.

DEFINITION 2.3 (Multilinear multiplication $\circ$). *Let $A \in \mathbb{R}^{\mathcal{I}}$, $\mu \in \{1, \ldots, d\}$ and $U_\mu \in \mathbb{R}^{J_\mu \times \mathcal{I}_\mu}$. Then the μ-mode multiplication $U_\mu \circ_\mu A$ is defined by the matricization*

$$(U_\mu \circ_\mu A)^{(\mu)} := U_\mu A^{(\mu)} \quad \in \mathbb{R}^{J_\mu \times \mathcal{I}^{(\mu)}},$$

with entries

$$(U_\mu \circ_\mu A)_{(i_1,\ldots,i_{\mu-1},j,i_{\mu+1},\ldots,i_d)} := \sum_{i_\mu=1}^{n_\mu} (U_\mu)_{j,i_\mu} A_{(i_1,\ldots,i_d)}.$$

The multilinear multiplication with matrices $U_\nu \in \mathbb{R}^{J_\nu \times I_\nu}$, $\nu = 1, \ldots, d$, is defined by

$$(U_1, \ldots, U_d) \circ A := U_1 \circ_1 \cdots U_d \circ_d A \quad \in \mathbb{R}^{J_1 \times \cdots \times J_d}.$$

The order of the mode multiplications is irrelevant for the multilinear multiplication.

DEFINITION 2.4 (Tucker rank, Tucker format, mode frames). *The* Tucker rank *of a tensor* $A \in \mathbb{R}^{\mathcal{I}}$ *is the tuple* $(k_1, \dots, k_d)$ *with (element-wise) minimal entries* $k_\mu \in \mathbb{N}_0$ *such that there exist (column-wise) orthonormal matrices* $U_\mu \in \mathbb{R}^{n_\mu \times k_\mu}$ *and a so-called* core tensor $C \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ *with*

$$A = (U_1, \dots, U_d) \circ C. \tag{2.1}$$

The representation of the form (2.1) is called the orthogonal Tucker format, or in short we say $A = (U_1, \dots, U_d) \circ C$ *is an orthogonal Tucker tensor. We call a representation of the form (2.1) with arbitrary* $U_\mu \in \mathbb{R}^{n_\mu \times \tilde{k}_\mu}$ *the Tucker format. The set of tensors of Tucker rank at most* $(k_1, \dots, k_d)$ *is denoted by* $\mathrm{Tucker}(k_1, \dots, k_d)$. *The matrices* U_μ *are called mode frames for the Tucker tensor representation.*

For fixed orthonormal mode frames $U_\mu \in \mathbb{R}^{n_\mu \times k_\mu}$ the unique core tensor C minimizing $\|A - (U_1, \dots, U_d) \circ C\|$ is

$$C = (U_1^T, \dots, U_d^T) \circ A.$$

The following definition is due to De Lathauwer et al. [4].

DEFINITION 2.5 (Tucker truncation). *Let* $A \in \mathbb{R}^{\mathcal{I}}$. *Let*

$$A^{(\mu)} = U_\mu \Sigma_\mu V_\mu^T, \qquad U_\mu \in \mathbb{R}^{n_\mu \times n_\mu},$$

be a singular value decomposition with diagonal matrix $\Sigma_\mu = \mathrm{diag}(\sigma_{\mu,1}, \dots, \sigma_{\mu,n_\mu})$. *Then the* truncation *of* A *to Tucker rank* $(k_1, \dots, k_d)$ *is defined by*

$$\mathcal{T}_{(k_1,\dots,k_d)}(A) := (\tilde{U}_1 \tilde{U}_1^T, \dots, \tilde{U}_d \tilde{U}_d^T) \circ A \quad = (\tilde{U}_1, \dots, \tilde{U}_d) \circ \left((\tilde{U}_1^T, \dots, \tilde{U}_d^T) \circ A \right),$$

where $\tilde{U}_\mu$ *is the matrix of the first* k_μ *columns of* U_μ.

The truncation $\mathcal{T}_{(k_1,\dots,k_d)}(A)$ yields an orthogonal Tucker tensor ($\tilde{U}_\mu$ is orthogonal). The exact representation $A = (\tilde{U}_1, \dots, \tilde{U}_d) \circ C$ is called the higher order SVD (HOSVD). Since the core tensor is uniquely defined by the orthonormal mode frames U_μ, the approximation of a tensor A in $\mathrm{Tucker}(k_1, \dots, k_d)$ is a minimization problem on a (product) Grassmann manifold. A best approximation A^{best} always exists. The geometry of the Grassmann manifold can be exploited to develop efficient Newton and quasi-Newton methods for a local optimization [6, 17]. As an initial guess one can use the Tucker truncation which allows for an explicit a priori error bound given next.

LEMMA 2.6 (Tucker approximation). *Let* $A \in \mathbb{R}^{\mathcal{I}}$. *We denote the best approximation of* A *in* $\mathrm{Tucker}(k_1, \dots, k_d)$ *by* A^{best}. *The error of the truncation is bounded by*

$$\|A - \mathcal{T}_{(k_1,\dots,k_d)}(A)\| \le \sqrt{\sum_{\mu=1}^{d} \sum_{i=k_\mu+1}^{n_\mu} \sigma_{\mu,i}^2} \le \sqrt{d}\|A - A^{\mathrm{best}}\|,$$

where the $\sigma_{\mu,i}$ *are the* μ*-mode singular values from Definition 2.5.*

Proof. Property 10 in [4]. ◻

The error bound stated in Lemma 2.6 is an a priori upper bound for the truncation error in terms of the best approximation error. The truncation is in general not a best approximation (but it may serve as an initial guess for a subsequent optimization). In the following section we will provide an elegant proof for this Lemma.

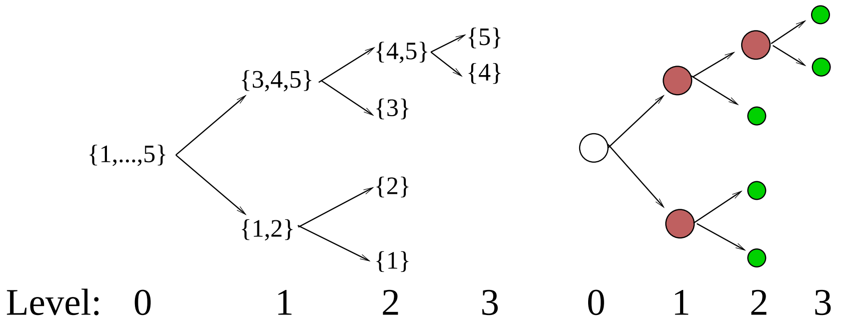

FIG. 3.1. *Left: A dimension tree for $d = 5$. Right: The interior nodes $\mathcal{I}(T_{\mathcal{I}})$ are colored dark (brown), the leaves $\mathcal{L}(T_{\mathcal{I}})$ are light (green) and the root is white.*

3. Hierarchical Tucker Format. The hierarchical Tucker format is a multilevel variant of the Tucker format — multilevel in terms of the order of the tensor. In order to define the format we have to introduce a hierarchy among the modes $\{1, \dots, d\}$.

DEFINITION 3.1 (Dimension tree). *A dimension tree or mode cluster tree $T_{\mathcal{I}}$ for dimension $d \in \mathbb{N}$ is a tree with root $\mathrm{Root}(T_{\mathcal{I}}) = \{1, \dots, d\}$ and depth $p = \lceil \log_2(d) \rceil := \min\{i \in \mathbb{N}_0 \mid i \geq \log_2(d)\}$ such that each node $t \in T_d$ is either*

1. *a leaf and singleton $t = \{\mu\}$ on level $\ell \in \{p-1, p\}$ or*
2. *the union of two disjoint successors $S(t) = \{s_1, s_2\}$:*

$$t = s_1 \,\dot{\cup}\, s_2. \tag{3.1}$$

The level ℓ of the tree is defined as the set of all nodes having a distance of exactly ℓ to the root, cf. Figure 3.1. We denote the level ℓ of the tree by

$$T_{\mathcal{I}}^{\ell} := \{t \in T_{\mathcal{I}} \mid \mathrm{level}(t) = \ell\}.$$

The set of leaves of the tree is denoted by $\mathcal{L}(T_{\mathcal{I}})$ and the set of interior (non-leaf) nodes is denoted by $\mathcal{I}(T_{\mathcal{I}})$. A node of the tree is a so-called mode cluster (a union of modes).

The dimension tree is almost a complete binary tree, except that on the last but one level there may appear leaves. In principle one could base the following considerations on arbitrary non-binary dimension trees, but for the ease of presentation we have restricted this. The **canonical dimension tree** is of the form presented in Figure 3.1 where each node $t = \{\mu_1, \dots, \mu_q\}$, $q > 1$, has two successors

$$t_1 := \{\mu_1, \dots, \mu_r\}, \quad r := \lfloor q/2 \rfloor := \max\{i \in \mathbb{N}_0 \mid i \leq q/2\}, \qquad t_2 := \{\mu_{r+1}, \dots, \mu_q\}.$$

LEMMA 3.2. *On each level ℓ of the dimension tree $T_{\mathcal{I}}$ of depth p the nodes are disjoint subsets of $\{1, \dots, d\}$. The number of nodes on level ℓ is*

$$\# T_{\mathcal{I}}^{\ell} = \begin{cases} 2^{\ell} & \text{for } \ell < p \text{ and} \\ 2d - 2^p \quad (\leq d) & \text{for } \ell = p. \end{cases}$$

For a complete binary tree $2d - 2^p = 2^{p+1} - 2^p = 2^p$ holds. The total number of nodes is $2d - 1$, the number of leaves is d and the number of interior nodes is $d - 1$.

Proof. The disjointness follows by (3.1). For levels $\ell = 0, \dots, p-1$ the tree is binary and thus the number of nodes doubles for each level. On the last but one level there are 2^{p-1} (disjoint) nodes, these can be either singletons (s) or two-element sets (t), thus $\#s + \#t = 2^{p-1}$. The total number of modes is d, thus $\#s + 2\#t = d$. Together we have $\#t = d - 2^{p-1}$, i.e. $2\#t = 2d - 2^p$ nodes (singletons) on level p. The total number of nodes is

$$\sum_{\ell=0}^{p-1} 2^\ell + 2d - 2^p = 2^p - 1 + 2d - 2^p = 2d - 1.$$

$\square$

Definition 3.3 (Matricization). *For a mode cluster t in a dimension tree $T_\mathcal{I}$ we define the complementary cluster $t' := \{1, \ldots, d\} \setminus t$,*

$$\mathcal{I}_t := \bigtimes_{\mu \in t} \mathcal{I}_\mu, \qquad \mathcal{I}_{t'} := \bigtimes_{\mu \in t'} \mathcal{I}_\mu,$$

and the corresponding t-matricization

$$\mathcal{M}_t : \mathbb{R}^\mathcal{I} \to \mathbb{R}^{\mathcal{I}_t \times \mathcal{I}_{t'}}, \qquad (\mathcal{M}_t(A))_{(i_\mu)_{\mu\in t},(i_\mu)_{\mu\in t'}} := A_{(i_1,\ldots,i_d)},$$

where the special case is $\mathcal{M}_\emptyset(A) := \mathcal{M}_{\{1,\ldots,d\}}(A) := A$. We use the short notation $A^{(t)} := \mathcal{M}_t(A)$.

We provide a simple example: let the tensor A be of the form

$$A = a \otimes b \otimes q \otimes r \in \mathbb{R}^{\mathcal{I}_1 \times \mathcal{I}_2 \times \mathcal{I}_3 \times \mathcal{I}_4},$$

where $\otimes$ denotes the usual outer product or tensor product

$$(x_1 \otimes \cdots \otimes x_d)_{(i_1,\cdots,i_d)} = (x_1)_{i_1} \cdot \ \cdots \ \cdot (x_d)_{i_d} = \prod_{\mu=1}^{d} (x_\mu)_{i_\mu}.$$

Then the matricizations with respect to $\{1,2\}$ and $\{2,3\}$ are

$$A^{(\{1,2\})} = (a \otimes b)(q \otimes r)^T \in \mathbb{R}^{(\mathcal{I}_1 \times \mathcal{I}_2) \times (\mathcal{I}_3 \times \mathcal{I}_4)},$$
$$A^{(\{2,3\})} = (b \otimes q)(a \otimes r)^T \in \mathbb{R}^{(\mathcal{I}_2 \times \mathcal{I}_3) \times (\mathcal{I}_1 \times \mathcal{I}_4)}.$$

Definition 3.4 (Hierarchical rank). *Let $T_\mathcal{I}$ be a dimension tree. The* hierarchical rank $(k_t)_{t \in T_\mathcal{I}}$ *of a tensor $A \in \mathbb{R}^\mathcal{I}$ is defined by*

$$\forall t \in T_\mathcal{I} : \quad k_t := \operatorname{rank}(A^{(t)}).$$

The set of all tensors of hierarchical rank (node-wise) at most $(k_t)_{t\in T_\mathcal{I}}$ is denoted by

$$\mathcal{H}\text{-Tucker}((k_t)_{t\in T_\mathcal{I}}) := \{A \in \mathbb{R}^\mathcal{I} \mid \forall t \in T_\mathcal{I} : \ \operatorname{rank}(A^{(t)}) \le k_t\}.$$

According to the definition of the hierarchical rank one can define the hierarchical SVD by the node-wise SVDs of the matrices $A^{(t)}$, cf. Figure 3.2. However, it is not obvious why and how this should lead to an efficient representation and correspondingly efficient algorithms. Instead, we will introduce a nested representation and reveal the connection to the node-wise SVDs afterwards.

Definition 3.5 (Frame tree, t-frame, transfer tensor). *Let $t \in T_\mathcal{I}$ be a mode cluster and $(k_t)_{t\in T_\mathcal{I}}$ a family of non-negative integers. We call a matrix $U_t \in \mathbb{R}^{\mathcal{I}_t \times k_t}$*

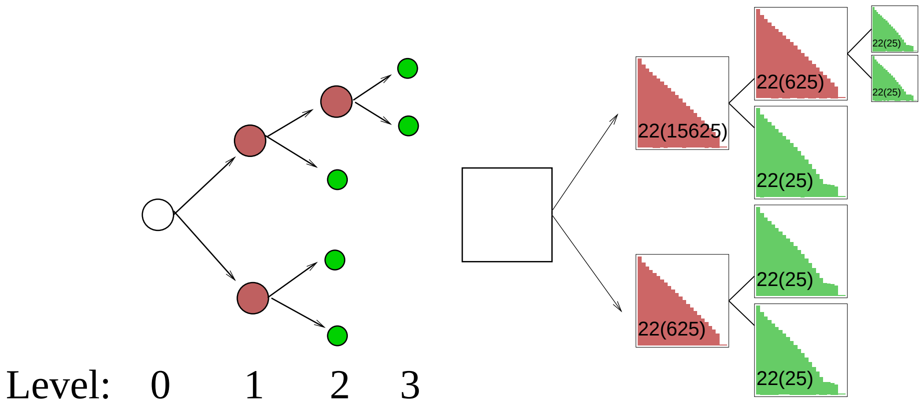

FIG. 3.2. *For each non-root node $t \in T_{\mathcal{I}}$ of the dimension tree (left, cf. Figure 3.1) a SVD of $A^{(t)}$ is computed. In each box for the right tree the largest 24 singular values of $A^{(t)}$ are plotted in logarithmic scale ranging from 1 down to 10^{-16} (on the abscissa the number of the singular value and on the ordinate the logarithm of the singular value). The first number printed in the box is the number of singular values larger than 10^{-14} and the number in brackets is the cardinality of $\mathcal{I}_t$.*

a t-frame and a tuple $(U_s)_{s\in T_{\mathcal{I}}}$ of frames a frame tree. A frame is called orthogonal if the columns are orthonormal and a frame tree is called orthogonal if each frame **except the root frame** *is orthogonal. A frame tree is nested if for each interior mode cluster t with successors $S(t) = \{t_1, t_2\}$ the following relation holds:*

$$\mathrm{span}\{(U_t)_i \mid 1 \le i \le k_t\} \ \subset\ \mathrm{span}\{(U_{t_1})_i \otimes (U_{t_2})_j \mid 1 \le i \le k_{t_1}, 1 \le j \le k_{t_2}\}.$$

The corresponding tensor $B_t \in \mathbb{R}^{k_t\times k_{t_1}\times k_{t_2}}$ of coefficients for the representation of the columns $(U_t)_i$ of U_t by the columns of U_{t_1}, U_{t_2},

$$(U_t)_i = \sum_{j=1}^{k_{t_1}} \sum_{l=1}^{k_{t_2}} (B_t)_{i,j,l}\ (U_{t_1})_j \otimes (U_{t_2})_l, \tag{3.2}$$

is called the transfer tensor*.*

For a nested frame tree it is sufficient to provide the transfer tensors B_t for all interior mode clusters $t \in \mathcal{I}(T_{\mathcal{I}})$ and the t-frames U_t only for the leaves $t \in \mathcal{L}(T_{\mathcal{I}})$. Note that we have not yet imposed an orthogonality condition on the t-frames.

DEFINITION 3.6 (Hierarchical Tucker format). *Let $T_{\mathcal{I}}$ be a dimension tree, $(k_t)_{t\in T_{\mathcal{I}}}$ a family of non-negative integers and $A \in \mathcal{H}\text{-Tucker}((k_t)_{t\in T_{\mathcal{I}}})$. Let $(U_t)_{t\in T_{\mathcal{I}}}$ be a nested frame tree with transfer tensors $(B_t)_{t\in\mathcal{I}(T_{\mathcal{I}})}$ and*

$$\forall t \in T_{\mathcal{I}} : \mathrm{image}(A^{(t)}) = \mathrm{image}(U_t), \qquad A = U_{\{1,\dots,d\}}.$$

Then the representation $((B_t)_{t\in\mathcal{I}(T_{\mathcal{I}})}, (U_t)_{t\in\mathcal{L}(T_{\mathcal{I}})})$ is a hierarchical Tucker representation *of A. The family $(k_t)_{t\in T_{\mathcal{I}}}$ is the hierarchical representation rank. Note that the columns of U_t need not be linear independent.*

The representation of a tensor $A \in \mathcal{H}\text{-Tucker}((k_t)_{t\in T_{\mathcal{I}}})$ in the hierarchical Tucker format with orthogonal frame tree and minimal k_t is unique up to orthogonal transformation of the t-frames.

LEMMA 3.7 (Storage complexity). *Let $T_{\mathcal{I}}$ be a dimension tree and $A \in \mathcal{H}\text{-Tucker}((k_t)_{t\in T_{\mathcal{I}}})$ given in hierarchical Tucker representation*

$((B_t)_{t\in\mathcal{I}(T_\mathcal{I})},(U_t)_{t\in\mathcal{L}(T_\mathcal{I})})$ and $B_t \in \mathbb{R}^{k_t\times k_{t_1}\times k_{t_2}}$ for $S(t)=\{t_1,t_2\}$, i.e. B_t of minimal size. Then the total storage for all transfer tensors $(B_t)_{t\in\mathcal{I}(T_\mathcal{I})}$ and leaf-frames $(U_t)_{t\in\mathcal{L}(T_\mathcal{I})}$ in terms of number of entries is bounded by

$$\mathrm{Storage}((B_t)_{t\in\mathcal{I}(T_\mathcal{I})},(U_t)_{t\in\mathcal{L}(T_\mathcal{I})}) \;\le\; (d-1)k^3 + k\sum_{\mu=1}^{d} n_\mu, \qquad k := \max_{t\in T_\mathcal{I}} k_t, \quad (3.3)$$

i.e. linearly in the dimension d (provided the representation parameter k is uniformly bounded).

Proof. For each leaf $t=\{\mu\}$ of the dimension tree we have to store the t-frame $U_t\in\mathbb{R}^{n_\mu\times k_t}$ which yields the second term in (3.3). For all $d-1$ interior mode clusters (Lemma 3.2) we have to store the transfer tensors $B_t\in\mathbb{R}^{k_t\times k_{t_1}\times k_{t_2}}$, each has at most k^3 entries. □

LEMMA 3.8 (Successive truncation). *Let $A\in\mathbb{R}^\mathcal{I}$ and π_t,π_s orthogonal projections. Then*

$$\|A-\pi_t\pi_s A\|^2 \le \|A-\pi_t A\|^2 + \|A-\pi_s A\|^2.$$

Proof. We have

$$\|A-\pi_t\pi_s A\| = \|(I-\pi_t)A + \pi_t(A-\pi_s A)\|.$$

Due to the orthogonality of $(I-\pi_t),\pi_t$ we conclude

$$\|A-\pi_t\pi_s A\|^2 = \|(I-\pi_t)A\|^2 + \|\pi_t(A-\pi_s A)\|^2 \le \|(I-\pi_t)A\|^2 + \|A-\pi_s A\|^2.$$

□

DEFINITION 3.9 (Orthogonal frame projection). *Let $T_\mathcal{I}$ be a dimension tree, $t\in T_\mathcal{I}$ and U_t an orthogonal t-frame. Then we define the orthogonal frame projection $\pi_t:\mathbb{R}^\mathcal{I}\to\mathbb{R}^\mathcal{I}$ in matricized form by*

$$(\pi_t A)^{(t)} := U_t U_t^T A^{(t)} \quad (t\ne\{1,\dots,d\}), \qquad \pi_{\{1,\dots,d\}}A := A.$$

In particular Lemma 3.8 proves Lemma 2.6: let U_t, $t=\{\mu\}$, denote the matrix of the k_t singular vectors of $A^{(t)}$ corresponding to the largest singular values. Then $\|A-\pi_t A\|^2 = \sum_{i=k_\mu+1}^{n_\mu}\sigma_{\mu,i}^2$. Since $\pi_t A$ is the best approximation of A with μ-mode rank k_t, we also have $\|A-\pi_t A\|^2 \le \|A-A^{\mathrm{best}}\|^2$ for the best Tucker approximation A^{best} and thus

$$\|A-\mathcal{T}_{(k_1,\dots,k_d)}(A)\| \le \sqrt{d}\|A-A^{\mathrm{best}}\|.$$

The order of the projections in a product of the form $(\prod_{t\in T_\mathcal{I}}\pi_t)$ is relevant (the π_t do not necessarily commute). One has to be careful with the ordering, because the result of the product of the projections differs structurally.

LEMMA 3.10. *Let $T_\mathcal{I}$ be a dimension tree and $A\in\mathbb{R}^\mathcal{I}$. For all $t\in T_\mathcal{I}$ let $U_t\in\mathbb{R}^{\mathcal{I}_t\times k_t}$ be orthogonal t-frames. Then for any order of the projections π_t holds*

$$\|A-\prod_{t\in T_\mathcal{I}}\pi_t A\|^2 \le \sum_{t\in T_\mathcal{I}}\|A-\pi_t A\|^2.$$

Proof. Apply Lemma 3.8 successively for all nodes of the dimension tree. □

THEOREM 3.11 (Hierarchical truncation error). *Let $T_{\mathcal{I}}$ be a dimension tree and $A \in \mathbb{R}^{\mathcal{I}}$. Let A^{best} denote the best approximation of A in $\mathcal{H}$-*Tucker$((k_t)_{t\in\mathcal{I}})$ *and let π_t be the orthogonal frame projection for the t-frame U_t that consists of the left singular vectors of $A^{(t)}$ corresponding to the k_t largest singular values $\sigma_{t,i}$ of $A^{(t)}$. Then for any order of the projections $\pi_t, t \in T_{\mathcal{I}}$, holds*

$$\|A - \prod_{t\in T_{\mathcal{I}}} \pi_t A\| \le \sqrt{\sum_{t\in T_{\mathcal{I}}} \sum_{i>k_t} \sigma_{t,i}^2} \le \sqrt{2d-2}\|A - A^{\mathrm{best}}\|.$$

Proof. For any of the projections holds $\|A - \pi_t A\|^2 = \sum_{i>k_t} \sigma_{t,i}^2 \le \|A - A^{\mathrm{best}}\|$ and for the root $\|A - \pi_{\{1,\dots,d\}} A\| = 0$ (w.l.o.g. $k_{\{1,\dots,d\}} = 1$). Applying Lemma 3.10 and Lemma 3.2 yields

$$\|A - \prod_{t\in T_{\mathcal{I}}} \pi_t A\|^2 \le \sum_{t\in T_{\mathcal{I}}} \sum_{i>k_t} \sigma_{t,i}^2 \le (2d-2)\|A - A^{\mathrm{best}}\|^2.$$

□

REMARK 3.12. *The estimate given in the previous theorem is not optimal and it can be improved as follows: for the root t of the dimension tree and its successors t_1, t_2 one can combine both projections $\pi_{t_1}.\pi_{t_2}$ into a single projection via the SVD. This combined projection (with the pairs of the singular vectors) then has the same error as any of the two projections π_{t_1} or π_{t_2}. Thereby, the error of the truncation can be estimated by*

$$\|A - \prod_{t\in T_{\mathcal{I}}} \pi_t A\| \le \sqrt{2d-3}\|A - A^{\mathrm{best}}\|.$$

In dimension $d = 2$ this coincides with the SVD estimate and in $d = 3$ this coincides with the original one-level Tucker estimate by De Lathauwer et al.

DEFINITION 3.13 (Kronecker product). *The Kronecker product $A \otimes_{\mathcal{K}} B$ of two matrices $A \in \mathbb{R}^{I\times J}, B \in \mathbb{R}^{K\times L}$ is defined by*

$$(A \otimes_{\mathcal{K}} B)_{(i,k),(j,\ell)} := A_{i,j} B_{k,\ell}, \qquad A \otimes_{\mathcal{K}} B \in \mathbb{R}^{(I\times K)\times(J\times L)}.$$

EXAMPLE 3.14 (Increasing the rank by projection). *We consider the tensor $A \in \mathbb{R}^{3\times3\times3}$ in matricized form*

$$A^{(\{1,2\})} := \left[\begin{array}{c|c|c} u_1 \otimes q_1 & u_2 \otimes q_2 & u_1 \otimes q_2 \end{array}\right], \quad A^{(\{1,2\})}_{(i,j),\ell} = \begin{cases} (u_1 \otimes q_1)_{i,j} & \text{if } \ell = 1 \\ (u_2 \otimes q_2)_{i,j} & \text{if } \ell = 2 \\ (u_1 \otimes q_2)_{i,j} & \text{if } \ell = 3 \end{cases}$$

with vectors

$$u_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, q_1 = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}, q_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

The mode cluster $t = \{1, 2\}$ has the two successors $t_1 = \{1\}, t_2 = \{2\}$ and we consider the orthogonal mode frames

$$U_t := \left[\begin{array}{c|c|c} u_1 \otimes q_1 & u_2 \otimes q_2 & q_1 \otimes q_2 \end{array}\right], U_{t_1} := \left[\begin{array}{c|c} u_1 & u_2 \end{array}\right], U_{t_2} := \left[\begin{array}{c|c} q_1 & q_2 \end{array}\right].$$

Clearly, π_{t_1} *will project to* t_1*-rank* $\operatorname{rank}((\pi_{t_1}A)^{(1)}) = 2$*. We will now show that the rank is at least* 3 *if we apply all three projectors. The matrix* Q *for the projection* $\pi_{t_1}\pi_{t_2}$ *is given by the Kronecker product*

$$Q = U_{t_1}U_{t_1}^T \otimes_{\mathcal{K}} U_{t_2}U_{t_2}^T = (u_1u_1^T + u_2u_2^T) \otimes_{\mathcal{K}} (q_1q_1^T + q_2q_2^T).$$

We thus obtain

$$QU_t = \left[\begin{array}{c|c|c} u_1 \otimes q_1 & u_2 \otimes q_2 & \frac{1}{\sqrt{2}}u_1 \otimes q_2 \end{array}\right].$$

The combined projection reads

$$\begin{aligned}(\pi_t\pi_{t_1}\pi_{t_2}A)^{(\{1,2\})} &= U_tU_t^TQA^{(\{1,2\})} = U_t(QU_t)^TA^{(\{1,2\})}\\ &= U_t\left[\begin{array}{c|c|c} u_1 \otimes q_1 & u_2 \otimes q_2 & \frac{1}{\sqrt{2}}u_1 \otimes q_2 \end{array}\right]^T A^{(\{1,2\})}\\ &= U_t\left[\begin{array}{c|c|c} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & \frac{1}{\sqrt{2}} \end{array}\right] = \left[\begin{array}{c|c|c} u_1 \otimes q_1 & u_2 \otimes q_2 & \frac{1}{\sqrt{2}}q_1 \otimes q_2 \end{array}\right].\end{aligned}$$

The matricization with respect to $t_1 = \{1\}$ *is of rank three,*

$$(\pi_t\pi_{t_1}\pi_{t_2}A)^{(1)} = \left[\begin{array}{c|c|c} u_1q_1^T & u_2q_2^T & q_1(\frac{1}{\sqrt{2}}q_2)^T \end{array}\right],$$

because u_1, u_2, q_1 *are linearly independent. We conclude: the first projection* $\pi_{t_1}\pi_{t_2}$ *maps* A *into* $\operatorname{Tucker}(2,2,3)$*, but after the coarser projection* π_t *the* 1*-mode rank is three and thus* $\pi_t\pi_{t_1}\pi_{t_2}A \notin \operatorname{Tucker}(2,2,3)$*. This is because* π_t *mixes the* t_1*-frame and the* t_2*-frame.*

LEMMA 3.15 (Structure of the hierarchical truncation). *Let* $T_{\mathcal{I}}$ *be a dimension tree of depth* p*,* $A \in \mathbb{R}^{\mathcal{I}}$ *and* $(k_t)_{t\in\mathcal{I}}$ *a family of non-negative integers. Let* $(U_t)_{t\in T_{\mathcal{I}}}, U_t \in \mathbb{R}^{\mathcal{I}_t \times k_t}$*, be an orthogonal frame tree (not necessarily nested). Then the tensor*

$$A_{\mathcal{H}} := \prod_{t\in T_{\mathcal{I}}^p} \pi_t \cdots \prod_{t\in T_{\mathcal{I}}^1} \pi_t A$$

belongs to $\mathcal{H}$-$\operatorname{Tucker}((k_t)_{t\in\mathcal{I}})$*.*

Proof. We define the tensors

$$A_{\mathcal{H},\ell} := \prod_{t\in T_{\mathcal{I}}^\ell} \pi_t \cdots \prod_{t\in T_{\mathcal{I}}^1} \pi_t A.$$

We prove $\operatorname{rank}(A_{\mathcal{H},\ell}^{(t)}) \le k_t$ for all $t \in T_{\mathcal{I}}$ with $\operatorname{level}(t) \le \ell$ by induction over the level $\ell = 1, \dots, p$. Level $\ell = 1$ is the Tucker truncation and thus the statement is true for $\ell = 1$. Now let $\ell > 1$ and assume that

$$\forall t \in T_{\mathcal{I}},\ \operatorname{level}(t) \le \ell - 1: \qquad \operatorname{rank}(A_{\mathcal{H},\ell-1}^{(t)}) \le k_t.$$

By construction

$$A_{\mathcal{H},\ell} = \prod_{t\in T_{\mathcal{I}}^\ell} \pi_t A_{\mathcal{H},\ell-1}.$$

This is the Tucker truncation on level ℓ applied to $A_{\mathcal{H},\ell-1}$ and thus for all $t \in T_{\mathcal{I}}^{\ell}$ on level ℓ the rank bound is fulfilled. It remains to show that for all levels $0, \ldots, \ell-1$ the rank bound is (still) fulfilled, i.e., that the rank is not increased by the projections on level ℓ. Now let $t \in T_{\mathcal{I}}^{j}, j < \ell$. Let $s \in T_{\mathcal{I}}^{\ell}$. We will show that the rank of $A_{\mathcal{H},\ell-1}^{(t)}$ is not increased by the projection π_s. Due to the tree structure s is either a subset of t or they are disjoint.
Case $s \subset t$: Let $\hat{s} := t \setminus s$. Then the projection π_s is of the matricized form

$$(\pi_s A)^{(t)} = \left(U_s U_s^T \otimes_{\mathcal{K}} I\right) A^{(t)}$$

with I being the $\mathcal{I}_{\hat{s}} \times \mathcal{I}_{\hat{s}}$ identity. The rank is not increased by the multiplication.
Case $s \cap t = \emptyset$: Let $\hat{s} := \{1, \ldots, d\} \setminus (t \cup s)$. Then the projection π_s is of the matricized form

$$(\pi_s A)^{(t)} = A^{(t)} \left(U_s U_s^T \otimes_{\mathcal{K}} I\right)$$

with I being the $\mathcal{I}_{\hat{s}} \times \mathcal{I}_{\hat{s}}$ identity. The rank is not increased by the multiplication. □

Notation 3.16. *By*

$$\psi_{t,k}(A) \quad \in \mathbb{R}^{\mathcal{I}_t \times k}$$

we denote the $\mathcal{I}_t \times k$ matrix whose columns are the left singular vectors of $A^{(t)}$ corresponding to the k largest singular values of $A^{(t)}$.

Definition 3.17 (Hierarchical root-to-leaves truncation). *Let $T_{\mathcal{I}}$ be a dimension tree of depth p, $(k_t)_{t\in\mathcal{I}}$ a family of non-negative integers and $A \in \mathbb{R}^{\mathcal{I}}$. We define the hierarchical root-to-leaves truncation $A_{\mathcal{H}} \in \mathcal{H}$-*Tucker*$((k_t)_{t\in\mathcal{I}})$ by*

$$A_{\mathcal{H}} := \prod_{t\in T_{\mathcal{I}}^{p}} \pi_t \cdots \prod_{t\in T_{\mathcal{I}}^{1}} \pi_t A,$$

where π_t are the projections based on $U_t := \psi_{t,k_t}(A) \in \mathbb{R}^{\mathcal{I}_t \times k_t}$.

The hierarchical Tucker representation of $A_{\mathcal{H}}$ from the previous definition is obtained by projection of the t-frames into the span of the sons $U_{t_1} \otimes_{\mathcal{K}} U_{t_2}$. The procedure for the construction is given in Algorithm 1. We want to remark that the algorithm is formulated for arbitrary tensors and the specialization to $\mathcal{H}$-Tucker tensors is the topic of the next section.

Theorem 3.18 (Characterization of hierarchical approximability). *Let $T_{\mathcal{I}}$ be a dimension tree, $A \in \mathbb{R}^{\mathcal{I}}$, $(k_t)_{t\in T_{\mathcal{I}}}$ a family of non-negative integers and $\varepsilon > 0$. If there exists a tensor A^{best} of hierarchical rank $(k_t)_{t\in\mathcal{I}}$ and $\|A - A^{\text{best}}\| \le \varepsilon$, then the singular values of $A^{(t)}$ for each node t can be estimated by*

$$\sqrt{\sum_{i>k_t} \sigma_i^2} \le \varepsilon.$$

*On the other hand, if the singular values fulfill the bound $\sqrt{\sum_{i>k_t} \sigma_i^2} \le \varepsilon/\sqrt{2d-3}$, then the truncation yields an $\mathcal{H}$-*Tucker* tensor $A_{\mathcal{H}} := \prod_{t\in T_{\mathcal{I}}} \pi_t A$ such that*

$$\|A - A_{\mathcal{H}}\| \le \varepsilon.$$

Proof. The second part is proven by Theorem 3.11. The first part follows from the fact that $(A^{\text{best}})^{(t)}$ is a rank k_t approximation of $A^{(t)}$ with $\|A^{(t)} - (A^{\text{best}})^{(t)}\|_F \le \varepsilon$. ∎

In Algorithm 1 we provide a method for the truncation of an arbitrary tensor to hierarchical rank $(k_t)_{t\in T_{\mathcal{I}}}$, of course one can as well prescribe node-wise tolerances ε_t for the truncation of singular values: according to Theorem 3.18 one can prescribe node-wise tolerance $\varepsilon/\sqrt{2d-2}$ in order to obtain a guaranteed error bound of $\|A - A_{\mathcal{H}}\| \le \varepsilon$. The complexity of Algorithm 1 is estimated in Lemma 3.19.

Algorithm 1 Root-to-leaves truncation of arbitrary tensors to $\mathcal{H}$-Tucker format

Require: Input tensor $A \in \mathbb{R}^{\mathcal{I}}$, dimension tree $T_{\mathcal{I}}$ (depth $p > 0$), target representation rank $(k_t)_{t\in T_{\mathcal{I}}}$.

for each singleton $t \in \mathcal{L}(T_{\mathcal{I}})$ **do**

Compute an SVD of $A^{(t)}$ and store the dominant k_t left singular vectors in the columns of the t-frame U_t.

end for

for $\ell = p-1, \dots, 0$ **do**

for each mode cluster $t \in \mathcal{I}(T_{\mathcal{I}})$ on level ℓ **do**

Compute an SVD of $A^{(t)}$ and store the dominant k_t left singular vectors in the columns of the t-frame U_t.

Let U_{t_1} and U_{t_2} denote the frames for the successors of t on level $\ell+1$. Compute the entries of the transfer tensor:

$$(B_t)_{i,j,\nu} := \langle (U_t)_i, (U_{t_1})_j \otimes (U_{t_2})_\nu \rangle$$

end for

end for

Compute the entries of the root (with sons t_1, t_2) transfer tensor:

$$(B_{\{1,\dots,d\}})_{1,j,\nu} := \langle (A, (U_{t_1})_j \otimes (U_{t_2})_\nu \rangle$$

return $\mathcal{H}$-Tucker representation $((U_t)_{t\in\mathcal{L}(T_{\mathcal{I}})}, (B_t)_{t\in\mathcal{I}(T_{\mathcal{I}})})$ for $A_{\mathcal{H}} \in \mathcal{H}\text{-Tucker}((k_t)_{t\in T_{\mathcal{I}}})$.

LEMMA 3.19 (Complexity of Algorithm 1). *The complexity of Algorithm 1 for a tensor $A \in \mathbb{R}^{\mathcal{I}}$ and dimension tree $T_{\mathcal{I}}$ of depth $p > 0$ is in $\mathcal{O}\left(\left(\prod_{\mu=1}^d n_\mu\right)^{3/2}\right)$.*

Proof. We have to compute singular value decompositions for all $A^{(t)}$, and those decompositions have a complexity of $\mathcal{O}(\min(\#\mathcal{I}_t, \#\mathcal{I}_{t'})^2 \max(\#\mathcal{I}_t, \#\mathcal{I}_{t'}))$, where t' is the complementary mode cluster $t' := \{1, \dots\} \setminus t$. Without loss of generality we can assume $n_\mu \ge 2$ for all modes μ. Then the complexity of the SVD for the root is zero, that for the two successors t, t' of the root is

$$C_{SVD}(\min(\#\mathcal{I}_t, \#\mathcal{I}_{t'})^2 \max(\#\mathcal{I}_t, \#\mathcal{I}_{t'})) \le C_{SVD} \left(\prod_{\mu=1}^d n_\mu\right)^{3/2},$$

where C_{SVD} is a universal constant for the SVD. For each further level there are at most two times more nodes, but the cardinality of $\mathcal{I}_t, \mathcal{I}_{t'}$ is reduced by at least a factor of two ($n_\mu \ge 2$) so that the complexity for the SVDs is quartered. Therefore the total complexity is bounded by

$$\sum_{\ell=0}^{p} 2^{-\ell} C_{SVD} \left(\prod_{\mu=1}^{d} n_\mu \right)^{3/2} \leq 2C_{SVD} \left(\prod_{\mu=1}^{d} n_\mu \right)^{3/2}.$$

□

The truncation presented in Algorithm 1 requires the computation of all (full) SVDs. We want to avoid the superlinear complexity $\mathcal{O}(\prod_{\mu=1}^{d} n_\mu)^{3/2}$ and instead work with a core tensor that becomes smaller as we come closer to the root of the tree. This means that we compute the SVDs not for the original tensor but for an already truncated one. The algorithm for this is given in Algorithm 2 and the complexity is estimated in Lemma 3.21.

DEFINITION 3.20 (Hierarchical leaves-to-root truncation). *Let $T_\mathcal{I}$ be a dimension tree of depth p, $(k_t)_{t\in\mathcal{I}}$ a family of non-negative integers and $A \in \mathbb{R}^\mathcal{I}$. We denote $A_{\widetilde{\mathcal{H}},p+1} := A$. For all levels $\ell = p, \dots, 1$ and $t \in (T_\mathcal{I}^\ell)$ let π_t denote the frame projection for $U_t := \psi_{t,k_t}(A_{\widetilde{\mathcal{H}},\ell+1}) \in \mathbb{R}^{\mathcal{I}_t \times k_t}$ and*

$$A_{\widetilde{\mathcal{H}},\ell} := \prod_{t \in T_\mathcal{I}^\ell} \pi_t A_{\widetilde{\mathcal{H}},\ell+1}.$$

Then we define the hierarchical leaves-to-root truncation by $A_\mathcal{H} := A_{\widetilde{\mathcal{H}},1}$.

Algorithm 2 Leaves-to-root truncation of arbitrary tensors to $\mathcal{H}$-Tucker format

---

Require: Input tensor $A \in \mathbb{R}^\mathcal{I}$, dimension tree $T_\mathcal{I}$ (depth $p > 0$), target representation rank $(k_t)_{t\in T_\mathcal{I}}$.

for each singleton $t \in \mathcal{L}(T_\mathcal{I})$ **do**

Compute an SVD of $A^{(t)}$ and store the dominant k_t left singular vectors in the columns of the t-frame U_t.

end for

Compute the core tensor $C_p := (U_1^T, \dots, U_d^T) \circ A$.

for $\ell = p-1, \dots, 0$ **do**

Initialize $C_\ell := C_{\ell+1}$.

for each mode cluster $t \in \mathcal{I}(T_\mathcal{I})$ on level ℓ **do**

Compute an SVD of $(C_{\ell+1})^{(t)}$ and store the dominant k_t left singular vectors in the columns of the t-frame $U_t \in \mathbb{R}^{k_{t_1} k_{t_2} \times k_t}$. Let U_{t_1} and U_{t_2} denote the corresponding frames for the successors t_1, t_2 of t on level $\ell + 1$. Compute the entries of the transfer tensor

$$(B_t)_{i,j,\nu} := \langle (U_t)_i, (U_{t_1})_j \otimes (U_{t_2})_\nu \rangle.$$

Update the core tensor $C_\ell := U_t^T \circ_t C_\ell$.

end for

end for

return $\mathcal{H}$-Tucker representation $((U_t)_{t\in\mathcal{L}(T_\mathcal{I})}, (B_t)_{t\in\mathcal{I}(T_\mathcal{I})})$ for $A_\mathcal{H} \in \mathcal{H}$-Tucker$((k_t)_{t\in T_\mathcal{I}})$.

LEMMA 3.21 (Complexity of leaves-to-root truncation). *The complexity of Algorithm 2 for a tensor $A \in \mathbb{R}^\mathcal{I}$ and dimension tree $T_\mathcal{I}$ of depth $p > 0$ is bounded* *by*

$$\mathcal{O}\left(\sum_{\mu=1}^{d} n_\mu \prod_{\nu=1}^{d} n_\nu + dk^2 \prod_{\nu=1}^{d} n_\nu\right), \qquad k := \max_{t\in T_\mathcal{I}} k_t.$$

Proof. For all leaves $t = \{\mu\}$ we have to compute the singular value decompositions of $A^{(\mu)}$ which is of complexity (C_{SVD} being again the generic constant for the SVD)

$$\sum_{\mu=1}^{d} C_{SVD}\, n_\mu^2 \prod_{\nu\neq\mu} n_\nu = C_{SVD} \sum_{\mu=1}^{d} n_\mu \prod_{\nu=1}^{d} n_\nu.$$

For all other levels $\ell = 0, \dots, p-1$ we have to compute SVDs of matrices of size at most $k_{t_1}k_{t_2} \times \prod_{\nu\notin t} n_\nu$. The complexity for this is at most

$$C_{SVD} k_{t_1}^2 k_{t_2}^2 \prod_{\nu\notin t} n_\nu \leq C_{SVD} k_{t_1} k_{t_2} \prod_{\nu=1}^{d} n_\nu \leq C_{SVD} k^2 \prod_{\nu=1}^{d} n_\nu.$$

Summing this up over all nodes of the tree yields the estimate. ☐

THEOREM 3.22 (Leaves-to-root truncation). *Let $T_\mathcal{I}$ be a complete binary dimension tree and $A \in \mathbb{R}^\mathcal{I}$. Let A^{best} denote the best approximation of A in* $\mathcal{H}$-Tucker$((k_t)_{t\in\mathcal{I}})$. *Then the error of the Leaves-to-Root truncation $A_{\widetilde{\mathcal{H}}}$ (Algorithm 2) is bounded by*

$$\|A - A_{\widetilde{\mathcal{H}}}\| \leq (2+\sqrt{2})\sqrt{d}\|A - A^{\mathrm{best}}\|.$$

Proof. The first truncation step on level $\ell = p$ is the Tucker truncation which yields t-frames U_t for all nodes $t \in T_\mathcal{I}^\ell$ and an error bound of the form

$$\|A - A_{\widetilde{\mathcal{H}},p}\| = \|A - \prod_{t\in T_\mathcal{I}^p} \pi_t A\| \leq \sqrt{2^p}\|A - A^{\mathrm{best}}\|,$$

where A^{best} is the best approximation (possibly worse than the one-level best approximation) in $\mathcal{H}$-Tucker$((k_t)_{t\in\mathcal{I}})$. On any level $\ell = p-1, \dots, 0$ we construct the t-frames U_t for all nodes $t \in T_\mathcal{I}^\ell$ that yield a Tucker truncation of $A_{\widetilde{\mathcal{H}},\ell+1}$ the error of which is bounded in terms of the best possible approximation A_ℓ^{best} of $A_{\widetilde{\mathcal{H}},\ell+1}$ using frames on level ℓ:

$$\|A_{\widetilde{\mathcal{H}},\ell+1} - A_{\widetilde{\mathcal{H}},\ell}\| \leq \sqrt{2^\ell}\|A_{\widetilde{\mathcal{H}},\ell+1} - A_\ell^{\mathrm{best}}\|.$$

Now let $\pi_t^*, t \in T_\mathcal{I}^\ell$ be projections that yield the best approximation of A in the Tucker format defined by the nodes t and ranks k_t on level ℓ of the dimension tree. Then $\prod_{t\in T_\mathcal{I}^\ell} \pi_t^* A$ fulfills the rank bound on level ℓ and due to Lemma 3.15 also the additional projection to the finer nodes $\prod_{i=\ell+1}^{p}\prod_{t\in T_\mathcal{I}^i}\pi_t \prod_{t\in T_\mathcal{I}^\ell}\pi_t^* A$ fulfills the rank bound. This constructed approximation is not better than the best approximation on level ℓ:

$$\begin{aligned}\|A_{\widetilde{\mathcal{H}},\ell+1} - A_\ell^{\mathrm{best}}\| &\leq \|\prod_{i=\ell+1}^{p}\prod_{t\in T_\mathcal{I}^i}\pi_t A - \prod_{i=\ell+1}^{p}\prod_{t\in T_\mathcal{I}^i}\pi_t \prod_{t\in T_\mathcal{I}^\ell}\pi_t^* A\| \\ &\leq \|A - \prod_{t\in T_\mathcal{I}^\ell}\pi_t^* A\| \quad\leq\quad \|A - A^{\mathrm{best}}\|.\end{aligned}$$

Thus we can estimate

$$\begin{aligned}\|A - A_{\widetilde{\mathcal{H}}}\| &\leq \|A - A_{\widetilde{\mathcal{H}},p}\| + \sum_{\ell=1}^{p-1} \|A_{\widetilde{\mathcal{H}},\ell+1} - A_{\widetilde{\mathcal{H}},\ell}\| \\ &\leq (\sqrt{2^p} + \sum_{\ell=1}^{p-1} \sqrt{2^\ell})\|A - A^{\mathrm{best}}\| \quad \leq \quad (2+\sqrt{2})\sqrt{d}\|A - A^{\mathrm{best}}\|.\end{aligned}$$

□

4. Truncation of Hierarchical Tucker Tensors. In this Section we want to derive an efficient realization of the truncation procedures from the previous section for the special case that the input tensor is already given in a data-sparse format, namely the hierarchical Tucker format.

Definition 4.1 (Brother of a mode cluster). *Let $T_{\mathcal{I}}$ be a dimension tree and $t \in T_{\mathcal{I}}$ a non-root mode cluster with father f. Then we define the unique mode cluster $\bar{t} \in T_{\mathcal{I}}$ such that $f = t \,\dot\cup\, \bar{t}$ as the* brother *of t.*

Lemma 4.2. *Let $T_{\mathcal{I}}$ be a dimension tree and $t \in \mathcal{I}(T_{\mathcal{I}})$ an interior node with two successors $t = t_1 \dot\cup t_2$. Further, let*

$$A^{(t)} = \sum_{\nu=1}^{k} u_\nu v_\nu^T$$

be a matricization of A. Let

$$u_\nu = \sum_{j=1}^{k_1} \sum_{l=1}^{k_2} c_{\nu,j,l} x_j \otimes y_l, \quad x_j \in \mathbb{R}^{\mathcal{I}_{t_1}}, y_l \in \mathbb{R}^{\mathcal{I}_{t_2}}, \qquad \nu = 1, \ldots, k$$

be a representation of the u_ν. Then the matricization of A with respect to t_1 is given by

$$A^{(t_1)} = \sum_{j=1}^{k_1} x_j \left(\sum_{\nu=1}^{k} \sum_{l=1}^{k_2} c_{\nu,j,l} y_l \otimes v_\nu \right)^T.$$

Proof. For the first matricization holds

$$\begin{aligned}A_{(i_1,\ldots,i_d)} = A^{(t)}_{(i_\mu)_{\mu\in t},(i_\mu)_{\mu\in t'}} &= \sum_{\nu=1}^{k} \sum_{j=1}^{k_1} \sum_{l=1}^{k_2} c_{\nu,j,l} (x_j)_{(i_\mu)_{\mu\in t_1}} (y_l)_{(i_\mu)_{\mu\in t_2}} (v_\nu)_{(i_\mu)_{\mu\in t'}} \\ &= \sum_{j=1}^{k_1} (x_j)_{(i_\mu)_{\mu\in t_1}} \left(\sum_{\nu=1}^{k} \sum_{l=1}^{k_2} c_{\nu,j,l} (y_l)_{(i_\mu)_{\mu\in t_2}} (v_\nu)_{(i_\mu)_{\mu\in t'}} \right) \\ &= \sum_{j=1}^{k_1} (x_j)_{(i_\mu)_{\mu\in t_1}} \left(\sum_{\nu=1}^{k} \sum_{l=1}^{k_2} c_{\nu,j,l} y_l \otimes v_\nu \right)_{(i_\mu)_{\mu\in t_1'}}.\end{aligned}$$

□

Lemma 4.3 (Matricization of tensors in hierarchical Tucker format). *Let $T_{\mathcal{I}}$ be a dimension tree, $A \in \mathcal{H}$-*Tucker*$((k_t)_{t\in\mathcal{I}})$ with nested orthogonal frame tree*

$(U_t)_{t\in T_{\mathcal{I}}}$ *and corresponding transfer tensors* $(B_t)_{t\in T_{\mathcal{I}}}$. *Let* $t \in T_{\mathcal{I}}^{(p)}, p \geq 1$, *and* $\mathrm{Root}(T_{\mathcal{I}}) = t_0, t_1, \ldots, t_{p-1}, t_p = t$ *a path of length* p. *Let* $\bar{U}^1, \ldots, \bar{U}^p$ *denote the frames of the corresponding brothers,* $B^0, \ldots, B^{p-1}$ *the corresponding transfer tensors and* $k_0, \ldots, k_p$ *the corresponding representation ranks. For convenience of notation we assume that the brother* $\bar{t}_\ell$ *is always the first and* t_ℓ *the second successor, i.e.*

$$(U_{t_\ell})_\nu = \sum_i \sum_j B^\ell_{\nu,i,j} \bar{U}^{\ell+1}_i \otimes (U_{t_{\ell+1}})_j.$$

Then the t*-matricization has the form*

$$A^{(t)} = \sum_{\nu=1}^{k_t} (U_t)_\nu (V_t)_\nu^T = U_t V_t^T,$$

where the complementary frame V_t *is defined by its columns* $(V_t)_1, \ldots, (V_t)_{k_t}$:

$$(V_t)_{j_p} = \left(\sum_{i_1=1}^{\bar{k}_1} \sum_{j_1=1}^{k_1} \cdots \sum_{i_{p-1}=1}^{\bar{k}_{p-1}} \sum_{j_{p-1}=1}^{k_{p-1}} \sum_{i_p=1}^{\bar{k}_p} B^0_{1,i_1,j_1} \cdots B^{p-1}_{j_{p-1},i_p,j_p} \right) \bar{U}^1_{i_1} \otimes \cdots \otimes \bar{U}^p_{i_p}$$

Proof. We prove the statement by induction over the level p of the mode cluster t. The start $p = 1$ is trivial: the tensor A has the representation (Lemma 4.2)

$$A = \sum_{i_1=1}^{k_{\bar{t}}} \sum_{j_1=1}^{k_t} B^0_{1,i_1,j_1} \bar{U}^1_{i_1} \otimes (U_t)_{j_1}, \qquad A^{(t)} = \sum_{j_1=1}^{k_t} (U_t)_{j_1} \left(\sum_{i_1=1}^{k_{\bar{t}}} B^0_{1,i_1,j_1} \bar{U}^1_{i_1} \right)^T.$$

For the node t_{p-1} holds by induction

$$A^{(t_{p-1})} = \sum_{\nu=1}^{k_{p-1}} (U_{t_{p-1}})_\nu (V_{t_{p-1}})_\nu^T, \qquad (U_{t_{p-1}})_\nu = \sum_{i_p=1}^{\bar{k}_p} \sum_{j_p=1}^{k_p} B^{p-1}_{\nu,i_p,j_p} \bar{U}^p_{i_p} \otimes (U_t)_{j_p}.$$

Together we obtain by Lemma 4.2

$$A^{(t_{p-1})} = \sum_{j_p=1}^{k_p} (U_t)_{j_p} \left(\sum_{\nu=1}^{k_{p-1}} \sum_{i_p=1}^{\bar{k}_p} B^{p-1}_{\nu,i_p,j_p} \bar{U}^p_{i_p} \otimes (V_{t_{p-1}})_\nu \right)^T = \sum_{j_p=1}^{k_p} (U_t)_{j_p} (V_t)_{j_p}^T.$$

∎

Definition 4.4 (Accumulated transfer tensors). *Let* $T_{\mathcal{I}}$ *be a dimension tree,* $(k_t)_{t\in\mathcal{I}}$ *a family of non-negative integers,* $(B_t)_{t\in T_{\mathcal{I}}}$ *transfer tensors of corresponding size. Let* $t \in T_{\mathcal{I}}^{(p)}, p \geq 1$, *and* $\mathrm{Root}(T_{\mathcal{I}}) = t_0, t_1, \ldots, t_{p-1}, t_p = t$ *a path of length* p. *Let* $B^0, \ldots, B^{p-1}$ *denote the corresponding transfer tensors (assuming that the brother* $\bar{t}_\ell$ *is always the first and* t_ℓ *the second successor). Let* $k_\nu := k_{t_\nu}$ *and* $\bar{k}_\nu := k_{\bar{t}_\nu}$. *Then we define the accumulated transfer tensor* $\widehat{B}_t$ *by*

$$\begin{aligned}
(\widehat{B}^1)_{j_1,s_1} &:= \sum_{i_1=1}^{\bar{k}_1} B^0_{1,i_1,j_1} B^0_{1,i_1,s_1}, \\
(\widehat{B}^\ell)_{j_\ell,s_\ell} &:= \sum_{s_{\ell-1}=1}^{k_{\ell-1}} \sum_{i_\ell=1}^{\bar{k}_\ell} \left(\sum_{j_{\ell-1}=1}^{k_{\ell-1}} (\widehat{B}^{\ell-1})_{j_{\ell-1},s_{\ell-1}} B^{\ell-1}_{j_{\ell-1},i_\ell,j_\ell} \right) B^{\ell-1}_{s_{\ell-1},i_\ell,s_\ell}, \qquad \ell = 2, \ldots, p, \\
\widehat{B}_t &:= \widehat{B}^p.
\end{aligned}$$

Remark 4.5. *The first accumulated tensors* $\widehat{B}_{t_1}, \widehat{B}_{t_2}$ *for the two sons of the root* t *can be computed in* $\mathcal{O}(k_{t_1}k_{t_2}^2 + k_{t_1}^2 k_{t_2})$. *For each further node the second formula in Definition 4.4 has to be applied and it involves inside the bracket a matrix multiplication of complexity* $\mathcal{O}(k_t^2 k_{t_1} k_{t_2})$ *for each son and the outer multiplication of complexity* $\mathcal{O}(k_t k_{t_1} k_{t_2}^2 + k_t k_{t_1}^2 k_{t_2})$. *For all nodes of the tree this sums up to*

$$\mathcal{O}\left(\sum_{t\in\mathcal{I}(T_\mathcal{I})} k_t k_{t_1} k_{t_2}^2 + k_t k_{t_1}^2 k_{t_2}\right) = \mathcal{O}\left(d \max_{t\in T_\mathcal{I}} k_t^4\right).$$

Lemma 4.6 (Gram matrices of complementary frames). *Let* $T_\mathcal{I}$ *be a dimension tree,* $A \in \mathcal{H}\text{-Tucker}((k_t)_{t\in\mathcal{I}})$ *with nested orthogonal frame tree* $(U_t)_{t\in T_\mathcal{I}}$ *and corresponding transfer tensors* $(B_t)_{t\in T_\mathcal{I}}$. *For each* $t \in T_\mathcal{I}$ *let* V_t *be the complementary frame from Lemma 4.3. Then* $\widehat{B}_t$ *is the Gram matrix for* V_t*:*

$$V_t^T V_t = \widehat{B}_t, \qquad \langle (V_t)_\nu, (V_t)_\mu \rangle = (\widehat{B}_t)_{\nu,\mu}.$$

Proof. We use the definitions and notations from Lemma 4.3. According to Lemma 4.3 and due to the orthogonality of each of the $\bar{t}_\ell$-frames $\bar{U}^\ell$ we obtain

$$\begin{aligned}
\langle (V_t)_\nu, (V_t)_\mu \rangle = &\sum_{i_1=1}^{\bar{k}_1} \sum_{j_1=1}^{k_1} \sum_{s_1=1}^{k_1} \cdots \sum_{i_{p-1}=1}^{\bar{k}_{p-1}} \sum_{j_{p-1}=1}^{k_{p-1}} \sum_{s_{p-1}=1}^{k_{p-1}} \sum_{i_p=1}^{\bar{k}_p} \\
&B^0_{1,i_1,j_1} \cdots B^{p-2}_{j_{p-2},i_{p-1},j_{p-1}} B^{p-1}_{j_{p-1},i_p,\nu} B^0_{1,i_1,s_1} \cdots B^{p-2}_{s_{p-2},i_{p-1},s_{p-1}} B^{p-1}_{s_{p-1},i_p,\mu} \\
= &\sum_{i_1=1}^{\bar{k}_1} \sum_{j_1=1}^{k_1} \sum_{s_1=1}^{k_1} \sum_{i_2=1}^{\bar{k}_2} \cdots \sum_{j_{p-1}=1}^{k_{p-1}} \sum_{s_{p-1}=1}^{k_{p-1}} \sum_{i_p=1}^{\bar{k}_p} \\
&B^0_{1,i_1,j_1} B^0_{1,i_1,s_1} \cdots B^{p-2}_{j_{p-2},i_{p-1},j_{p-1}} B^{p-2}_{s_{p-2},i_{p-1},s_{p-1}} B^{p-1}_{j_{p-1},i_p,\nu} B^{p-1}_{s_{p-1},i_p,\mu} \\
= &(\widehat{B}_t)_{\nu,\mu}.
\end{aligned}$$

□

According to the previous Lemma we can easily compute the left singular vectors of V_t^T which are the eigenvectors of the $k_t \times k_t$ matrix $\widehat{B}_t$. The matrix Q_t of singular vectors is the transformation matrix such that $U_t Q_t$ is the matrix of the left singular vectors of $A^{(t)}$ the singular values of which are the square roots of the eigenvalues of $\widehat{B}_t$. Thus, one can truncate either to fixed rank or one can determine the rank adaptively in order to guarantee a truncation accuracy of ε.

The nested mode frames were required to be orthogonal. If this is not yet the case, one has to orthogonalize the frame tree. The procedure for this is explained next and the complexity is estimated afterwards.

Lemma 4.7 (Frame transformation). *Let* $t \in T_\mathcal{I}$ *be a mode cluster with* t*-frame* U_t*, transfer tensor* B_t *and two sons* t_1, t_2 *with frames* U_{t_1}, U_{t_2}*, such that the columns fulfill*

$$(U_t)_i = \sum_{j=1}^{k_1} \sum_{l=1}^{k_2} (B_t)_{i,j,l} (U_{t_1})_j \otimes (U_{t_2})_l, \quad i = 1, \ldots, k.$$

Let $X \in \mathbb{R}^{k\times k}, Y \in \mathbb{R}^{k_1\times k_1}, Z \in \mathbb{R}^{k_2\times k_2}$ *and* Y, Z *invertible. Then we can rewrite the transformed frames as*

$$(U_t X)_i = \sum_{j=1}^{k_1'} \sum_{l=1}^{k_2'} (B_t')_{i,j,l} (U_{t_1} Y)_j \otimes (U_{t_2} Z)_l, \qquad B_t' := (X^T, Y^{-1}, Z^{-1}) \circ B_t.$$

Proof. The formula follows from elementary matrix multiplications. □

Algorithm 3 Orthogonalization of hierarchical Tucker tensors

Require: Input tensor $A_{\mathcal{H}} \in \mathcal{H}\text{-Tucker}((k_t)_{t\in T_{\mathcal{I}}})$ represented by $((U_t)_{t\in\mathcal{L}(T_{\mathcal{I}})}, (B_t)_{t\in\mathcal{I}(T_{\mathcal{I}})})$.

for each singleton $t \in \mathcal{L}(T_{\mathcal{I}})$ **do**

Compute a QR-decomposition of the t-frame U_t and define

$$U_t := Q, \qquad B_f := \begin{cases} (I, I, R) \circ B_f & \text{if } t \text{ is the second successor,} \\ (I, R, I) \circ B_f & \text{if } t \text{ is the first successor} \end{cases}$$

for the father f of t.

end for

for each mode cluster $t \in \mathcal{I}(T_{\mathcal{I}}) \setminus \{\mathrm{root}(T_{\mathcal{I}})\}$ **do**

Compute a QR-decomposition of $(B_t)^{(\{1,2\})}$,

$$(B_t)^{(\{1,2\})} = (Q_t)^{(\{1,2\})} R,$$

and set

$$B_t := Q_t, \quad B_f := \begin{cases} (I, I, R) \circ B_f & \text{if } t \text{ is the second successor,} \\ (I, R, I) \circ B_f & \text{if } t \text{ is the first successor} \end{cases}$$

of the father f of t.

end for

return nested orthogonal frames $(U_t)_{t\in\mathcal{L}(T_{\mathcal{I}})}$ and transfer tensors $(B_t)_{t\in\mathcal{I}(T_{\mathcal{I}})}$.

LEMMA 4.8 (Complexity for the orthogonalization of nested frame trees). *The complexity of Algorithm 3 for a tensor* $A_{\mathcal{H}} \in \mathcal{H}\text{-Tucker}((k_t)_{t\in T_{\mathcal{I}}})$ *with nested frames* $(U_t)_{t\in\mathcal{L}(T_{\mathcal{I}})}$ *and transfer tensors* $(B_t)_{t\in\mathcal{I}(T_{\mathcal{I}})}$ *is bounded by*

$$\mathcal{O}\left(\sum_{\mu=1}^{d} n_\mu k_\mu^2 + \sum_{t\in\mathcal{I}(T_{\mathcal{I}}),\mathrm{Sons}(t)=\{t_1,t_2\}} k_t^2 k_{t_1} k_{t_2} + k_t k_{t_1}^2 k_{t_2} + k_t k_{t_1} k_{t_2}^2\right).$$

Proof. For each interior node we have to compute QR decompositions which are of complexity $\mathcal{O}(k_t^2 k_{t_1} k_{t_2})$ and perform two mode multiplications $X \circ_\mu B_f$, $\mu = 2, 3$, which is of complexity $\mathcal{O}(k_t k_{t_1}^2 k_{t_2} + k_t k_{t_1} k_{t_2}^2)$. For the leaves $t = \{\mu\}$ a QR-factorization is of complexity $\mathcal{O}(n_\mu k_\mu^2)$. The sum over all nodes of the tree yields the desired bound. □

LEMMA 4.9 (Complexity for the $\mathcal{H}$-Tucker truncation). *The complexity for the truncation of an* $\mathcal{H}\text{-Tucker}((k_t)_{t\in T_{\mathcal{I}}})$*-Tensor* A *(not necessarily with orthogonal* *frames) to lower rank is*

$$\mathcal{O}(d \max_{t \in T_{\mathcal{I}}} k_t^4 + \sum_{\mu=1}^{d} n_\mu k_\mu^2).$$

Definition 4.10 (CANDECOMP, PARAFAC, Elementary Tensor Sum). *Let $A \in \mathbb{R}^{\mathcal{I}}$. The minimal number $k \geq 0$ such that*

$$A = \sum_{i=1}^{k} A_i, \qquad A_i = a_{i,1} \otimes \cdots \otimes a_{i,d}, \qquad a_{i,\mu} \in \mathbb{R}^{\mathcal{I}\mu}, \tag{4.1}$$

is the tensor rank or canonical rank of A. The rank one tensors A_i are called elementary tensors. A sum of the form (4.1) with arbitrary $k \geq 0$ is called an elementary tensor sum with representation rank k. In the literature the alternative names CANDECOMP [3] or PARAFAC [11] or CP are commonly used.

Remark 4.11 (Conversion of elementary tensor sums to $\mathcal{H}$-Tucker format). *Let $A \in \mathbb{R}^{\mathcal{I}}$ be a tensor represented by an elementary tensor sum*

$$A = \sum_{i=1}^{k} \bigotimes_{\mu=1}^{d} a_{i,\mu}, \qquad a_{i,\mu} \in \mathbb{R}^{\mathcal{I}_\mu}.$$

Then A can immediately be represented in the hierarchical Tucker format by the t-frames

$$\forall t = \{\mu\} \in \mathcal{L}(T_{\mathcal{I}}): \quad (U_t)_i := a_{i,\mu}, \quad i = 1, \ldots, k, \quad k_\mu := k,$$

and the transfer tensors

$$\forall t \in \mathcal{I}(T_{\mathcal{I}}) \setminus \mathrm{Root}(T_{\mathcal{I}}): \quad (B_t)_{i,j,l} := \begin{cases} 1 & \text{if } i = j = l \\ 0 & \text{otherwise,} \end{cases}, \quad B_t \in \mathbb{R}^{k \times k \times k}, \ \ k_t := k.$$

The root transfer tensor is

$$(B_{\{1,\ldots,d\}})_{1,j,l} := \begin{cases} 1 & \text{if } j = l \\ 0 & \text{otherwise,} \end{cases}, \qquad B_{\{1,\ldots,d\}} \in \mathbb{R}^{1 \times k \times k}, \quad k_{\{1,\ldots,d\}} := 1.$$

The frames are not yet orthogonal, so a subsequent orthogonalization and truncation is advisable to find a reduced representation. If we store the transfer tensors in sparse format, then the amount of storage is $k(d-1) + k\sum_{\mu=1}^{d} n_\mu$, i.e. almost the same as for a tensor represented as an elementary tensor sum.

The opposite conversion from $\mathcal{H}$-Tucker to elementary tensor sums with (almost) minimal representation rank k is highly non-trivial.

5. Comparison with other Formats. As mentioned in the introduction the hierarchical Tucker format is identical or similar to several other tensor formats.

5.1. Sequential Unfolding SVD, PARATREE. The sequential unfolding SVD or PARATREE from [16] is defined quite similarly as the $\mathcal{H}$-Tucker format. The first separation is via the SVD of the familiar form

$$A = \sum_{i_1=1}^{r} U_{1,i_1} \otimes U_{2,i_1}$$

Each of the U_{ν,i_1} ($\nu = 1, 2$) on level 1 is then again split via the SVD into

$$U_{\nu,i_1} = \sum_{i_2=1}^{r} U_{\nu,i_1,1,i_2} \otimes U_{\nu,i_1,2,i_2},$$

i.e., for each i_1 a different set of vectors $U_{\nu,i_1,1,i_2}, U_{\nu,i_1,2,i_2}$ is used. Each of the vectors is then split on level $\ell = 2, \ldots$ separately. On level ℓ the frames are indexed by

$$U_{\nu_1,i_1,\ldots,\nu_\ell,i_\ell}.$$

Thus, the complexity is no longer linear in the dimension d but scales exponentially in the depth of the tree.

5.2. Hierarchical MCTDH and Φ-System. The hierarchical or multilayer format from [1, 18] is exactly of the $\mathcal{H}$-Tucker form. This format is used in the Multi-Configuration Time-Dependent Hartree method (MCTDH) and to the best of our knowledge this is the first occurrence of a hierarchical tensor format.

The Φ-system representation from [10] is identical to the $\mathcal{H}$-Tucker-representation. In fact, it has been the starting point for our work and for writing this article.

The here presented analysis and hierarchical SVD, as well as the (almost best) truncation with a priori error estimate is thus a framework to perform arithmetics in the hierarchical MCTDH formulation and for Φ-system representations.

5.3. Tree Tensor and Tensor Train. The tree tensor (TT) format from [14] uses the same tree concept as in this article (without approximation on the leaf-level). However, in that article the truncation is not developed (and thus not analyzed). In [13] the truncation algorithm based on the SVD is provided and partially analyzed, namely for the case of degenerate trees (cf. Figures 5.1,5.2) which is then coined Tensor Train or TT* format or again TT format (developed in parallel to our truncation framework).

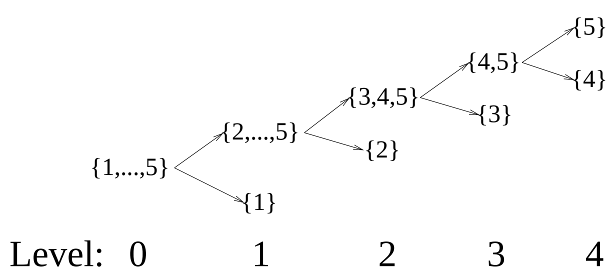

Fig. 5.1. *A degenerate tree for the TT format.*

Definition 5.1 (TT format). *The TT format of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ is given by a family of integers $(k_q)_{q=1}^{d-1}$, matrices G^1, G^d and tensors $G^2, \ldots, G^{d-1}$ such that*

$$A_{i_1,\ldots,i_d} = \sum_{j_1=1}^{k_1} \cdots \sum_{j_{d-1}=1}^{k_{d-1}} G^1_{i_1,j_1} G^2_{i_2,j_1,j_2} \cdots G^{d-1}_{i_{d-1},j_{d-2},j_{d-1}} G^d_{i_d,j_{d-1}} \tag{5.1}$$

Lemma 5.2. *Let $A \in \mathcal{H}$-Tucker$(T_{\mathcal{I}})$ for a degenerate tree $T_{\mathcal{I}}$ with root t_0 and successors $t_1, \ldots, t_{d-1}$ (these are always the first successors) that are leaves and*

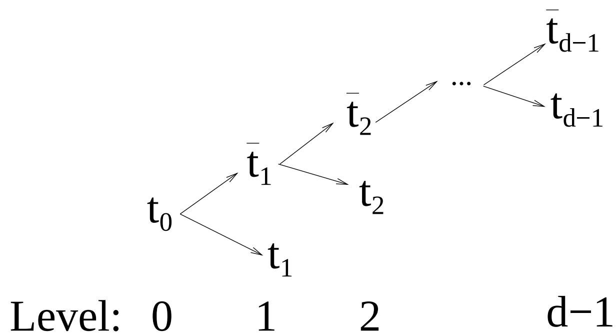

FIG. 5.2. *A degenerate tree for the TT format.*

$\bar{t}_1, \ldots, \bar{t}_{d-1}$ *their respective brothers (these are always the second successors; $\bar{t}_{d-1}$ is a leaf, too). Then A is of the form (5.1) with*

$$G^1_{i,j} := \sum_{\ell=1}^{k_{t_1}} (B_{t_0})_{1,\ell,j}(U_{t_1})_{i,\ell}, \qquad G^d_{i,j} := (U_{\bar{t}_{d-1}})_{i,j},$$

$$G^q_{i,j,m} := \sum_{\ell=1}^{k_{\bar{t}_q}} (B_{\bar{t}_{q-1}})_{j,\ell,m}(U_{t_q})_{i,\ell} \qquad \textit{for} \quad q = 2, \ldots, d-1.$$

Proof.

$$A_{i_1,\ldots,i_d} = \sum_{j_1=1}^{k_{\bar{t}_1}} \underbrace{\sum_{\ell_1=1}^{k_{t_1}} (B_{t_0})_{1,\ell_1,j_1}(U_{t_1})_{i_1,\ell_1}}_{G^1_{i_1,j_1}} (U_{\bar{t}_1})_{(i_2,\ldots,i_d),j_1},$$

$$(U_{\bar{t}_1})_{(i_2,\ldots,i_d),j_1} = \sum_{j_2=1}^{k_{\bar{t}_2}} \underbrace{\sum_{\ell_2=1}^{k_{t_2}} (B_{\bar{t}_1})_{j_1,\ell_2,j_2}(U_{t_2})_{i_2,\ell_2}}_{G^2_{i_2,j_1,j_2}} (U_{\bar{t}_2})_{(i_3,\ldots,i_d),j_2},$$

$$\ldots$$

$$(U_{\bar{t}_{d-2}})_{(i_{d-1},i_d),j_{d-2}} = \sum_{j_{d-1}=1}^{k_{\bar{t}_{d-1}}} \underbrace{\sum_{\ell_{d-1}=1}^{k_{t_{d-1}}} (B_{\bar{t}_{d-1}})_{j_{d-2},\ell_{d-1},j_{d-1}}(U_{t_{d-1}})_{i_{d-1},\ell_{d-1}}}_{G^{d-1}_{i_{d-1},j_{d-2},j_{d-1}}} \underbrace{(U_{\bar{t}_{d-1}})_{i_d,j_{d-1}}}_{G^d_{i_d,j_{d-1}}}.$$

□ Indeed, the truncation in the TT* format is a special case of the $\mathcal{H}$-Tucker-truncation (no truncation in the leaves). One can regard our analysis (the quasi-best approximation result) as an extension of the results from [13]. The main advantage of the TT* format is that it is easier to describe and program, because the SVD of the interior nodes in the $\mathcal{H}$-Tucker format is more involved. The essential question is: Does it pay off to use general trees or is it sufficient to work with degenerate ones ?

5.3.1. Parallelization. The tree structure for (almost) balanced trees, i.e., those having a depth proportional to $\log(d)$, is perfectly suited for parallelization. On each level all operations can be performed in parallel for all nodes on that level. Up to the logarithm for the depth of the tree one can obtain a perfect speedup. Using d processors one obtains a scaling of $\log(d)$ for the overall truncation complexity. This will be treated in more detail in a followup article.

The same kind of parallelization is not possible for the TT* format because the depth of the degenerate tree is proportional to d and the truncation is inherently sequential going from the root to the leaves.

5.3.2. Complexity. Let $T_{\mathcal{I}}$ be an arbitrary dimension tree and $A \in \mathbb{R}^{\mathcal{I}}$. We denote by

$$k_q := \operatorname{rank}(A^{(\{1,\ldots,q\})})$$

the rank of the matricizations appearing in the TT* format, i.e. the representation rank for a degenerate tree. Then the $\mathcal{H}$-Tucker-rank $k_t = \operatorname{rank}(A^{(t)})$ for a node $t = \{p, \ldots, \ell\}$ can be bounded by

$$k_t \le k_{p-1} k_\ell,$$

i.e. the largest necessary representation rank for the exact representation of A in $\mathcal{H}$-Tucker$((k_t)_{t\in T_{\mathcal{I}}})$ is bounded by the product of two TT*-ranks: By assumption there exists U, V with k_{p-1}, k_q columns such that

$$A^{(\{1,\ldots,p-1\})} = UU^T A^{(\{1,\ldots,p-1\})}, \quad A^{(\{1,\ldots,q\})} = VV^T A^{(\{1,\ldots,q\})}.$$

Then

$$A^{(t)} = A^{(t)}(UU^T \otimes_{\mathcal{K}} VV^T)$$

and the matrix on the right is of rank at most

$$\operatorname{rank}(UU^T \otimes_{\mathcal{K}} VV^T) \le \operatorname{rank}(UU^T)\operatorname{rank}(VV^T) \le k_{p-1}k_q.$$

The opposite is not true: Let $k_t = \operatorname{rank}(A^{(t)})$ for all $t \in T_{\mathcal{I}}$ be uniformly bounded by k. Then the TT*-rank k_q can only be bounded by

$$k_q \lesssim k^{\log_2(d)/2}$$

for a balanced tree $T_{\mathcal{I}}$ and

$$k_q \lesssim k^{d/2}$$

for an arbitrary tree $T_{\mathcal{I}}$ (proof not provided since the bound is worthless). In order to illustrate that this bound is sharp we give a numerical example. Let the tree $T_{\mathcal{I}}$ be of the form depicted in Figure 5.3. The tree has a depth of 10 in dimension $d = 10$. We construct a random tensor $A \in \mathcal{H}$-Tucker$((k_t)_{t\in T_{\mathcal{I}}})$, $k_t := 2$, $n_\mu := 2$ for all modes $\mu \in \{1, \ldots, d\}$. Then we compute the TT*-ranks k_q reported in Table 5.1. In particular $k_5 = 32 = 2^5 = k^{d/2}$.

By reordering the dimension indices one can obtain a degenerate tree as required by the TT* format, but that is not of interest here. Even if one allows an arbitrary

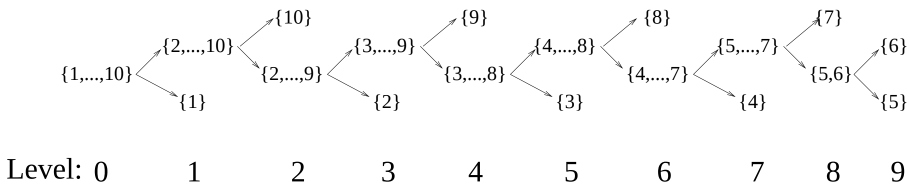

FIG. 5.3. *A special dimension tree $T_{\mathcal{I}}$ of depth* 10.

$q=$	1	2	3	4	5	6	7	8	9
$k_q=$	2	4	8	16	32	16	8	4	2

TABLE 5.1
TT-ranks for the special dimension tree $T_{\mathcal{I}}$ from Figure 5.3.*

reordering of the dimension indices for the TT* format, then a complete binary tree $T_{\mathcal{I}}$ will still provide an example where $\max_q k_q = k^{\log_2(d)/2}$.

For any tensor that allows an approximation by an elementary tensor sum with representation rank k, both hierarchical formats have their ranks bounded by $k_t \leq k, k_q \leq k$.

In practice it is of course a critical point to construct the dimension tree adaptively such that the ranks stay small, and this will be presented in a followup article.

6. Numerical Examples. The numerical examples in this section are focused on three questions:

1. How close to the measurements are the theoretical estimates of the truncation error, i.e. the ratio between node-wise errors and the total error ? In particular we are interested in the question whether or not the factor $\sqrt{d}$ appears.
2. What is the maximal attainable truncation accuracy, i.e. how close can we get to the machine precision ?
3. What are problem sizes that can realistically be tackled by the $\mathcal{H}$-Tucker format in terms of the dimension d and the maximal rank k ?

All computations are performed on an intel CPU with peak frequency 1.83 GHz and available main memory 1 GB.

6.1. Truncation from dense to $\mathcal{H}$-Tucker format. Our first numerical example is in $d = 5$ with mode size $n_\mu = 25$. The tensor A is a dense tensor with entries

$$A_{(i_1,\ldots,i_d)} := \left(\sum_{\mu=1}^{d} i_\mu^2\right)^{-1/2}$$

which corresponds to the discretization of the function $1/\|x\|$ on $[1,25]^5$. The time for the conversion (Algorithm 1) of the dense tensor to $\mathcal{H}$-Tucker format $A_{\mathcal{H}}$, the amount of storage needed for the frames U_t and transfer tensors B_t and the obtained relative approximation accuracy $\|A - A_{\mathcal{H}}\|$ are presented in Table 6.1. The node-wise SVD is shown in Figure 3.2. From the truncation we lose roughly 2–3 digits of precision compared to the maximal attainable machine precision `EPS` $\approx 10^{-16}$. It seems that

ε	$\|A - A_{\mathcal{H}}\|/\|A\|$	Storage (KB)	times (Sec)
1×10^{-2}	6.0×10^{-3}	3.6	105.8
1×10^{-4}	1.2×10^{-4}	11.2	104.1
1×10^{-6}	1.1×10^{-6}	29.3	103.5
1×10^{-8}	7.8×10^{-9}	58.1	104.8
1×10^{-10}	1.7×10^{-10}	92.7	108.0
1×10^{-12}	7.2×10^{-13}	153.2	104.8
1×10^{-14}	3.2×10^{-13}	298.0	104.0
1×10^{-16}	2.7×10^{-14}	615.1	106.9

TABLE 6.1
Converting a dense tensor to $\mathcal{H}$-Tucker *format.*

the node-wise rank is uniformly bounded (there is almost no variation between the ranks k_t) by $k \sim \log(1/\varepsilon)$.

6.2. Truncation of elementary tensor sums in $\mathcal{H}$-Tucker format. The second example is in higher dimension d with mode size $n_\mu = 1000$. The entries of the tensor $A_{\mathcal{H}}$ are approximations of

$$A_{(i_1,\dots,i_d)} := \left(\sum_{\mu=1}^{d} i_\mu^2\right)^{-1/2}, \qquad i\mu = 1,\dots,1000,$$

by exponential sums,

$$A_E := \sum_{j=1}^{35} \omega_j \bigotimes_{\mu=1}^{d} a_{j,\mu}, \qquad (a_{j,\mu})_{i_\mu} = \exp(-i_\mu^2 \alpha_j/d)$$

such that each entry is accurate up to $\varepsilon_E = 10^{-10}$,

$$|A_{(i_1,\dots,i_d)} - (A_E)_{(i_1,\dots,i_d)}| \le 7.315\times10^{-10}.$$

The weights ω_j and exponents α_j were obtained from W. Hackbusch and are available via the web page ($k = 35$, $R = 1000000$)

`http://www.mis.mpg.de/scicomp/EXP_SUM`

The tensor A_E (represented as an elementary tensor sum) is then converted to $\mathcal{H}$-Tucker format (error zero), which we denote by $A_{\mathcal{H}}$. The hierarchical rank is $k_t = 35$ for every mode cluster $t \in T_{\mathcal{I}}$. From this input tensor we compute truncations $A_{\mathcal{H},\varepsilon}$ to lower hierarchical rank by prescribing the (relative) truncation accuracy ε. In Tables 6.2 and 6.3 we report the accuracy $\|A_{\mathcal{H}} - A_{\mathcal{H},\varepsilon}\|/\|A_{\mathcal{H}}\|$, the storage requirements for $A_{\mathcal{H},\varepsilon}$ in MB as well as the time in seconds used for the truncation. We observe that the accuracy is

$$\|A_{\mathcal{H}} - A_{\mathcal{H},\varepsilon}\|/\|A_{\mathcal{H}}\| \approx 3\varepsilon$$

independent of the dimension d. The maximal attainable accuracy seems to be roughly $\varepsilon_{\min} \approx 10^{-13}$.

6.3. Truncation of $\mathcal{H}$-Tucker tensors. The third test is not any more concerned with the approximation accuracy, but purely with the computational complexity. Here, we setup an $\mathcal{H}$-Tucker tensor with node-wise ranks $k_t \equiv k$ and mode

ε	$\frac{\|A_{\mathcal{H}}-A_{\mathcal{H},\varepsilon}\|}{\|A\|}$	Storage	time	ε	$\frac{\|A_{\mathcal{H}}-A_{\mathcal{H},\varepsilon}\|}{\|A\|}$	Storage	time
10^{-4}	1.4×10^{-4}	0.31	0.33	10^{-4}	1.3×10^{-4}	0.37	0.73
10^{-6}	1.3×10^{-6}	0.59	0.34	10^{-6}	2.5×10^{-6}	0.50	0.80
10^{-8}	2.1×10^{-8}	1.00	0.36	10^{-8}	1.7×10^{-8}	0.90	0.88
10^{-10}	1.7×10^{-10}	1.60	0.39	10^{-10}	3.7×10^{-10}	1.22	0.83
10^{-12}	1.3×10^{-12}	2.75	0.44	10^{-12}	2.2×10^{-12}	1.87	0.80
10^{-14}	1.8×10^{-14}	3.54	0.47	10^{-14}	3.9×10^{-14}	2.76	0.91
10^{-16}	2.0×10^{-15}	3.88	0.48	10^{-16}	2.6×10^{-14}	4.05	0.97

TABLE 6.2
Truncating an $\mathcal{H}$-Tucker tensor of rank $k_t = 35$ in $d = 8$ (left) and $d = 16$ (right).

ε	$\frac{\|A_{\mathcal{H}}-A_{\mathcal{H},\varepsilon}\|}{\|A\|}$	Storage	time	ε	$\frac{\|A_{\mathcal{H}}-A_{\mathcal{H},\varepsilon}\|}{\|A\|}$	Storage	time
10^{-4}	1.9×10^{-4}	0.49	1.65	10^{-4}	5.5×10^{-5}	0.98	3.19
10^{-6}	2.2×10^{-6}	0.74	1.54	10^{-6}	9.2×10^{-7}	1.48	3.39
10^{-8}	1.4×10^{-8}	1.00	1.55	10^{-8}	3.2×10^{-8}	1.49	3.23
10^{-10}	2.4×10^{-10}	1.27	1.57	10^{-10}	1.5×10^{-10}	2.00	3.36
10^{-12}	1.3×10^{-12}	1.83	1.65	10^{-12}	3.1×10^{-12}	2.52	3.50
10^{-14}	3.3×10^{-14}	2.24	1.61	10^{-14}	1.1×10^{-13}	3.09	3.31
10^{-16}	7.3×10^{-15}	3.49	1.74	10^{-16}	1.1×10^{-13}	4.53	3.52

TABLE 6.3
Truncating an $\mathcal{H}$-Tucker tensor of rank $k_t = 35$ in $d = 32$ (left) and $d = 64$ (right).

sizes $n_\mu \equiv 20$. Then, we vary the rank k and dimension parameter d and measure the storage complexity as well as the complexity for the truncation (which is essentially independent of the target rank or accuracy). The results are reported in Table 6.4 (dashes mean that for that problem size we ran out of memory (1GB)). We conclude

k	d=	10	100	1,000	10,000	100,000	1,000,000	
1	size	0.00	0.02	0.16	1.60	16.02	160.22	MB
1	time	0.00	0.00	0.01	0.08	0.86	7.88	Sec.
2	size	0.00	0.04	0.37	3.66	36.62	366.21	MB
2	time	0.00	0.00	0.03	0.22	1.63	15.43	Sec.
5	size	0.02	0.17	1.71	17.16	171.66	-	MB
5	time	0.00	0.02	0.08	0.81	7.60	-	Sec.
10	size	0.08	0.90	9.14	91.54	915.51	-	MB
10	time	0.01	0.10	0.61	5.79	55.86	-	Sec.
20	size	0.52	6.29	63.97	640.75	-	-	MB
20	time	0.20	1.23	7.36	72.12	-	-	Sec.

TABLE 6.4
Storage complexity (MB) and truncation time for the $\mathcal{H}$-Tucker format with mode size $n_\mu = 20$.

that it is indeed possible to perform reliable numerical computations in dimension $d = 1,000,000$, and also rather large ranks of $k = 50$ are not a problem for dimensions $d = 1000$ on a simple notebook computer, cf. Table 6.5. On a larger desktop machine one can use $k = 100$ in dimension $d = 10,000$ (uses roughly 80 GB and takes ca. 10 hours).

k	d=	10	100	1,000	
25	size	1.15	13.59	138.05	MB
25	time	0.24	1.97	19.39	Sec.
50	size	8.03	97.29	989.93	MB
50	time	2.60	30.20	306.02	Sec.
100	size	68.74	755.39	-	MB
100	time	57.05	685.98	-	Sec.

TABLE 6.5

*Storage complexity (MB) and truncation time for the $\mathcal{H}$-*Tucker *format with mode size $n_\mu = 100$.*

7. Conclusion. We have defined the hierarchical singular value decomposition for tensors of order $d \geq 2$ in the $\mathcal{H}$-Tucker format (also known as the Φ-system representation [10] or previously the multilayer MCTDH format [18]). It is based on standard matrix SVDs for matricizations of the tensor. We are able to derive an almost optimal complexity computational scheme for tensors either in dense or in $\mathcal{H}$-Tucker format with a priori or adaptive control of the accuracy. In particular, we have obtained a quasi-best approximation (up to a factor of $\sqrt{2d-3}$) result comparable to the Eckart-Young bound for matrices (best approximation) or the HOSVD bound for Tensors in Tucker format (best approximation up to a factor of $\sqrt{d}$). The complexity to compute the hierarchical SVD for $\mathcal{H}$-Tucker tensors of constant representation rank k, mode size n and order d is $\mathcal{O}(dnk^2 + dk^4)$ using $\mathcal{O}(dnk + dk^3)$ units of storage.

The $\mathcal{H}$-Tucker format with constant representation rank k is a specialization of the Tucker format with multilinear rank k and it contains all tensors of (border) rank k. It remains to apply the new format in several areas in order to assess the usefulness of the format in terms of the necessary rank $(k_t)_{t\in T_\mathcal{I}}$. This requires the adaptive construction of a suitable dimension tree $T_\mathcal{I}$, which is possibly different for different application areas. For large-scale computations a parallel (distributed) scheme has to be developed, and since the SVDs and QR decompositions in each node are hardly parallelizable and of complexity $\mathcal{O}(k^3)$ and $\mathcal{O}(k^4)$, respectively, the parallelization has to be done with respect to the order d of the tensor.

REFERENCES

[1] M.H. Beck, A. Jäckle, G.A. Worth, and H.-D. Meyer. The multiconfiguration time-dependent Hartree (mctdh) method: A highly efficient algorithm for propagating wavepackets. *Phys. Reports*, 324:1–105, 2000.

[2] G. Beylkin and M. J. Mohlenkamp. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences*, 99(16):10246–10251, 2002.

[3] J. D. Carroll and J. J.Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. *Psychometrika*, 35:283–319, 1970.

[4] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[5] V. de Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[6] L. Eldén and B. Savas. A Newton-Grassmann method for computing the best multi-linear rank-(r_1, r_2, r_3) approximation of a tensor. *SIAM J. Matrix Anal. Appl.*, 31:248–271, 2009.

[7] M. Espig. *Effiziente Bestapproximation mittels Summen von Elementartensoren in hohen Dimensionen*. PhD thesis, Universität Leipzig, 2008.

[8] M. Griebel. A parallelizable and vectorizable multi-level algorithm on sparse grids. In *Parallel algorithms for partial differential equations, 6th GAMM-Seminar Kiel*, volume 31 of *Notes Numer. Fluid Mech.*, pages 94–100, 1991.

[9] M. Griebel. Adaptive sparse grid multilevel methods for elliptic pdes based on finite differences. *Computing*, 61:151–179, 1998.

[10] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *J. Fourier Anal. Appl.*, 15:706–722, 2009.

[11] R. A. Harshman. Foundations of the parafac procedure. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[12] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[13] I.V. Oseledets. Compact matrix form of the d-dimensional tensor decomposition. Preprint 09-01, Institute of Numerical Mathematics RAS, Moscow, Russia, 2009.

[14] I.V. Oseledets and E.E. Tyrtyshnikov. Breaking the curse of dimensionality, or how to use svd in many dimensions. Preprint 09-03, HKBU, Kowloon Tong, Hong Kong, 2009.

[15] M. Rajih, P. Comon, and R. Harshman. Enhanced line search: a novel method to accelerate PARAFAC. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1148–1171, 2008.

[16] J. Salmi, A. Richter, and V. Koivunen. Sequential unfolding SVD for tensors with applications in array signal processing. *IEEE Trans. on Signal Processing*, 2009. accepted.

[17] B. Savas and L.-H. Lim. Best multilinear rank approximation of tensors with quasi-Newton methods on Grassmannians. Technical Report LITH-MAT-R-2008-01-SE, Department of Mathematics, Linköpings Universitet, 2008.

[18] H. Wang and M. Thoss. Multilayer formulation of the multiconfiguration time-dependent hartree theory. *J. Chem. Phys.*, 119:1289–1299, 2003.

[19] C. Zenger. Sparse grids, parallel algorithms for partial differential equations. In *Parallel algorithms for partial differential equations, 6th GAMM-Seminar Kiel*, volume 31 of *Notes Numer. Fluid Mech.*, pages 241–251, 1991.