Sampling Rules for Tensor Reconstruction in Hierarchical Tucker Format

Melanie Kluge*

Bericht Nr. 392 **April 2014**

Key words: tensor completion, tensor approximation, tensor train

AMS Subject Classifications: 15A69, 65F99

Institut für Geometrie und Praktische Mathematik
RWTH Aachen

Templergraben 55, D–52056 Aachen (Germany)

---

* Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany, Email: kluge@igpm.rwth-aachen.de. Financial support from the DFG SPP-1324 under grant GRA2179/2-1 gratefully acknowledged.

Sampling Rules for Tensor Reconstruction in Hierarchical Tucker Format

Melanie Kluge *

April 16, 2014

The subject of this article is the development of an algorithm that reconstructs a high-dimensional tensor by a hierarchical ($\mathcal{H}$-) Tucker tensor with the help of a non-adaptive sampling rule. This sampling rule supports our approximation scheme coming from the matrix cross approximation and guarantees that we can build a tensor $A_{\mathcal{H}}$ in the desired format from only a few entries of the original tensor A. Under mild assumptions $A_{\mathcal{H}}$ is a reconstruction of A. In the numerical experiments we obtain convenient approximations also for tensors without low rank representation and for pertubed tensors.
Keywords: Tensor Completion, Tensor Approximation, Tensor Train
MSC: 15A69, 65F99,

1 Introduction

The aim of high-dimensional tensor reconstruction is to obtain the structure of a tensor from a small subset Ω of its entries coming from a calculation routine or a measurement process. The problem situation considered here allows one interaction with this routine or process and no further entry maybe requested. To motivate this task we consider an abstract scenario with a calculation routine and an applied scenario with a measurement process.
We start with an abstract scenario in the field of uncertainty quantification. A quantity of interest is, e.g., the expected value of a functional coming from a PDE solution, where the PDE solution is very expensive in terms of computational complexity and hence, the calculation routine maybe located externally. Then, only a few interactions with the routine are practicable. Let $\phi : \Gamma \to \mathbb{R}$ be a functional with $\Gamma \subset \mathbb{R}^d$ and defined as

$$\phi(y) := \int_S \Phi(u(x,y))\,dx \qquad S \subset \mathbb{R}^3$$

*Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany. Email: kluge@igpm.rwth-aachen.de. Financial support from the DFG SPP-1324 under grant GRA2179/2-1 gratefully acknowledged.

where $u : S \times \Gamma \to \mathbb{R}$ is a parameter dependent PDE solution with spatial variable $x \in S$ and an uncertain variable $y \in \Gamma$, e.g. a stochastic PDE solution as considered in [1, 2]. Let $f := f_1 \otimes \ldots \otimes f_d : \Gamma \to [0, \infty)$ be a d-dimensional density function, then the expected value is given by:

$$\mathbb{E}(\phi) = \int_\Gamma \phi(y) f(y)\, dy.$$

Monte Carlo (MC) and quasi-Monte Carlo (QMC) methods [3] approximate this integral value via a finite sum of N pseudorandom or quasirandom sampling points $y_{\mathbf{j}}$

$$\mathbb{E}(\phi) \approx \sum_{\mathbf{j}=1}^{N} \phi(y_{\mathbf{j}}) F(y_{\mathbf{j}}),$$

where $F := F_1 \otimes \ldots \otimes F_d : \Gamma \to [0, 1]$ is the distribution function. The set of the sampled points $y_{\mathbf{j}}$ is generated independently from the structure of ϕ and hence, only one interaction with the calculation routine to obtain the values $\phi(y_{\mathbf{j}})$ is necessary. Their independence from the structure of ϕ may also be a disadvantage. If the functional has a low rank structure or a low rank approximation, this information is not used in the sampling strategy.

In this article we describe a sampling rule that supports the structure of a low rank approximation. We consider the following discretization over a tensor grid:

$$\mathbb{E}_T(\phi) = \sum_{i_1=1}^{n_1} \ldots \sum_{i_d=1}^{n_d} \underbrace{\phi(\nu_{i_1,\ldots,i_d})}_{=:A_{i_1,\ldots,i_d}=:A_{\mathbf{i}}} L_{1,i_1}(y_1) F_1(y_1) \cdot \ldots \cdot L_{d,i_d}(y_d) F_d(y_d)$$

with 1-dimensional Chebyshev points ν_{i_ℓ} of order $n_\ell - 1$ and the corresponding Lagrange polynomials L_{ℓ,i_ℓ} ($\ell = 1, \ldots, d$). The value $\mathbb{E}_T(\phi)$ is given in a Tucker representation [4], which is known to suffer from the curse of dimensionality, and an approximation of the core tensor A in a low-rank tensor format becomes interesting.

We continue with an applied scenario where the object of interest is a high-dimensional functional $\Phi : \Gamma \to \mathbb{R}$ describing a physical phenomenon. If a representation of Φ is not known, a way to handle this is to discretise it over a tensor grid $G \subseteq \Gamma$. Then the resulting object is a high-dimensional tensor $A := (\Phi(y_{\mathbf{j}}))_{\mathbf{j} \in G}$. In practice the measurement process restricts the amount of measuring positions $y_{\mathbf{j}}$ and a reconstruction of A is desired. If it also restricts the time interval for data collection, e.g. in case of seismic data as considered in [5] or of EEG data as considered in [6], each measurement process is a nonrecurring one and hence, more than one interaction with the process is impossible.

We have these abstract and applied scenario in mind by considering the tensor reconstruction with the hierarchical Tucker format as the selected tensor format. The aim of this article is to develop a reconstruction algorithm with a sampling rule that

1. is an non-adaptive one, to handle the situation of one possible interaction with the calculation routine or measurement process to obtain tensor entries, and

2. supports an approximation scheme in the hierarchical Tucker format, to get a reconstruction or even a convenient approximation of the original tensor A.

In the following section we define the hierarchical Tucker format and give important properties. We start the third section with the formulation of the reconstruction problem in this format and consider the reconstruction with a random sampling strategy. Afterwards, we adapt a matrix approximation technique to the tensor format and develop a non-adaptive sampling rule. We merge all considerations in section four by the reconstruction algorithm itself. In the last section the algorithm is applied to high-dimensional unperturbed and perturbed tensors. The examples verify that the reconstruction is successful under mild assumptions and also that we obtain convenient approximations for tensors not fulfilling these assumptions or for perturbed ones.

2 Hierarchical Tucker format

The hierarchical ($\mathcal{H}$-) Tucker format [7] of a d-dimensional tensor $A \in \mathbb{R}^{I_1 \times \ldots \times I_d}$ is defined by the hierarchy of certain matricizations and their corresponding matrix ranks organized in a dimension tree.

2.1 Definition of the format

We obtain a matricization of the tensor $A \in \mathbb{R}^I$ by splitting the index set $I := I_1 \times \ldots \times I_d$ into a row index set I_t and a column index set $I_{\hat{t}}$.

Definition 1 (t-Matricization) *Let $A \in \mathbb{R}^I$ and $t \subseteq D := \{1, \ldots, d\}$ be a non-empty and fixed subset of the dimensions with the complement set $\hat{t} := D \setminus t$. Then the t-matricization $A^{(t)} \in \mathbb{R}^{I_t \times I_{\hat{t}}}$ with $I_t = \times_{\mu \in t} I_\mu$ and $I_{\hat{t}} = \times_{\nu \in \hat{t}} I_\nu$ is defined by*

$$A^{(t)}_{(i_\mu)_{\mu \in t}, (i_\nu)_{\nu \in \hat{t}}} := A_{i_1, \ldots, i_d} \quad \forall (i_1, \ldots, i_d) \in I.$$

We remark that the determination of the matrix rank $k_t = \operatorname{rank}(A^{(t)})$ is independent of the index order. For simplicity we choose the lexicographical order. In the next example we illustrate the Definition 1 by a small 3-dimensional tensor.

Example 2 (Matricization) *Let a 3-dimensional tensor A be given entry-wise by the formulation $A_{i_1,i_2,i_3} := i_1 + (i_2 - 1) \cdot 2 + (i_3 - 1) \cdot 4$ for all $(i_1, i_2, i_3) \in \{1, 2\}^3$. Then all possible t-matricizations are*

$$A^{(\{1\})} = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix} = (A^{(\{2,3\})})^T \text{ (Figure 1),}$$

$$A^{(\{2\})} = \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix} = (A^{(\{1,3\})})^T, \quad A^{(\{3\})} = \begin{pmatrix} 1 & 3 & 2 & 4 \\ 5 & 7 & 6 & 8 \end{pmatrix} = (A^{(\{1,2\})})^T$$

$$\text{and } A^{(\{1,2,3\})} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}.$$

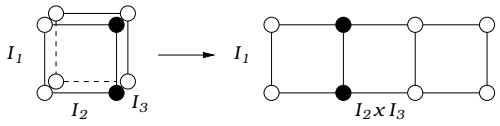

Figure 1: Left: 3-dimensional tensor A with 8 entries and the black marked fibre $(A_{1,2,1}, A_{2,2,1})^T$. Right: Matricization $A^{(\{1\})}$, where the black marked fibre in the tensor is the same vector as the second column in the matrix.

The idea to build a hierarchy of certain matricizations comes from the following nestedness property.

Lemma 3 (Nestedness property) *Let $t := \{q, \dots, s\} \subseteq D$ be split into the disjoint subsets $t_1 := \{q, \dots, r\}$ and $t_2 := \{r+1, \dots, s\}$ with $q \le r < s$. Then the corresponding matricizations are nested as follows:*

$$span\left(A^{(t)}\right) \subset span\left(A^{(t_1)} \otimes A^{(t_2)}\right) \tag{1}$$

Proof: We follow the proof of Lemma 3.1 in [8]. Select an arbitrary column index $j \in I_{\hat{t}}$ and define a vector v with $v_{(i_\mu)_{\mu \in t}} := A^{(t)}_{(i_\mu)_{\mu \in t}, j} \in \mathbb{R}^{I_t}$. Then the t_1-matricization $v^{(t_1)} = (v^{(t_2)})^T$ fulfills the relation $\text{span}\left(v^{(t_1)}\right) \subset \text{span}\left(A^{(t_1)}\right)$. By means of the Moore-Penrose pseudoinverse Z^+ of Z, we can determine $v^{(t_1)}$ by

$$v^{(t_1)} = A^{(t_1)} \, (A^{(t_1)})^+ \, (v^{(t_2)})^T = A^{(t_1)} \, (A^{(t_1)})^+ \, v^{(t_1)} \, [(A^{(t_2)})^+]^T \, (A^{(t_2)})^T.$$

■

This property motivates to organize the hierarchy in a binary tree.

Definition 4 (Dimension tree) *A dimension tree T_D is a binary tree with root D, where each node $\emptyset \neq t \in T_D$ is either*

(i) a leaf or singleton $t = \{\mu\}$ with no successor (belongs to $\mathcal{L}(T_D)$) or

(ii) an interior node with two successors (named sons) t_1 and t_2, such that $t = t_1 \dot{\cup} t_2$ (belongs to $\mathcal{I}(T_D)$).

With the dimension tree as a binary tree we may use matrix techniques for an $\mathcal{H}$-Tucker tensor like a hierarchical form of the SVD [9]. The dimension $r \in \{q, \dots, s\}$ of Lemma 3, where the set $\{q, \dots, s\}$ splits into t_1 and t_2, is not further specified. The two following, most popular settings lead to different dimension trees and hence, different rank tuples $\underline{k} = (k_t)_{t \in T_D}$ [10]:

(1) The canonical tree: The set $t = \{q, \dots, s\}$ halves into $t_1 = \{q, \dots, \lceil \frac{q+s}{2} \rceil\}$ and $t_2 = \{\lceil \frac{q+s}{2} \rceil + 1, \dots, s\}$.

(2) The TT-tree [11]: The set $t = \{q, \dots, s\}$ divides into the first dimension $t_1 = \{q\}$ and the remaining ones $t_2 = \{q+1, \dots, s\}$.

The choice of the tree is of high importance for the quality of the approximation. Finding an optimal dimension tree to a given problem is NP-hard and it is not part of this article. In [12] an algorithm to find a tree in an adaptive way is constructed. Under the assumption that

$$\operatorname{rank}(A^{(t)}) \leq K \quad \forall t \subseteq D \tag{2}$$

a dimension tree can be fixed a priori. Then, the choice of the tree mainly influences the storage complexity of the resulting low-rank tensor representation. We prefer the canonical tree, cf. Figure 2, because it has the minimal depth $p = \lfloor \log_2(d) \rfloor$ and we may parallelize the calculation.

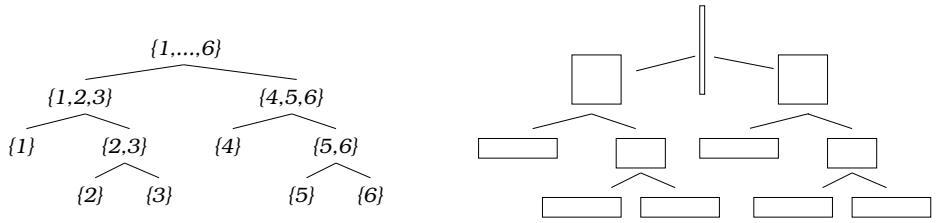

Figure 2: Left: Canonical tree for dimension $d = 6$ with minimal depth $p = 3$. Right: The scheme of the corresponding matricizations shows that in the root of the tree the tensor entries are lined up in one long vector and by going down to the leaves the entries are rearranged to more columns and fewer rows.

Definition 5 (Hierarchical Tucker tensor) *Let T_D be a canonical tree for dimension d and $\underline{k} = (k_s)_{s \in T_D}$ a tuple of ranks. A tensor is a hierarchical Tucker tensor with dimension tree T_D and at most hierarchical ($\mathcal{H}$)-rank $\underline{k}$, if it is an element of the set*

$$\mathcal{HT}\left(T_D, \underline{k}\right) := \left\{ A \in \mathbb{R}^I : \mathit{rank}\left(A^{(t)}\right) \leq k_t \quad \forall\, t \in T_D \right\}.$$

The nestedness property of Lemma 3 and the $\mathcal{H}$-rank of Definition 5 lead to a data-sparse representation of an $\mathcal{H}$-Tucker tensor in the following way: Let $A^{(t)} = U_t V_t^T$ be a representation with $U_t \in \mathbb{R}^{I_t \times k_t}$ and $V_t \in \mathbb{R}^{I_{\dot{t}} \times k_t}$ for all $t \in T_D$. Then, there exist three-dimensional transfer tensors $B_t \in \mathbb{R}^{k_{t_1} \times k_{t_2} \times k_t}$ which satisfy the nestedness equation

$$(U_t)_j = \sum_{\alpha=1}^{k_{t_1}} \sum_{\beta=1}^{k_{t_2}} (B_t)_{\alpha,\beta,j} \, (U_{t_1})_\alpha \otimes (U_{t_2})_\beta \qquad j = 1, \ldots, k_t \tag{3}$$

with $t = t_1 \dot{\cup} t_2$ and $t_1, t_2 \in T_D$. The tuple $(U_s)_{s \in T_D}$ is called a frame tree, cf Figure 3.

2.2 Properties of the format

From Definition 5 and equation (3) it follows that a tensor in the $\mathcal{H}$-Tucker format is represented by the frames $U_\mu \in \mathbb{R}^{n_\mu \times k_\mu}$ in the leaves μ and the transfer tensors

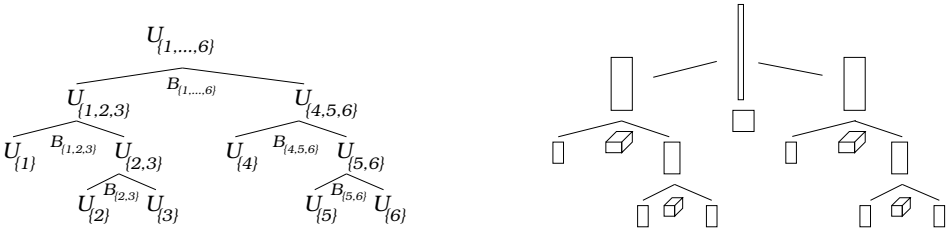

Figure 3: Left: Frame tree for dimension $d = 6$ with transfer tensors in the interior nodes. Right: The scheme of the corresponding matrix and tensor sizes gives an impression of the fact that the matrix U_t is smaller than $A^{(t)}$ and B_t again is usually smaller than U_t.

$B_t \in \mathbb{R}^{k_{t_1} \times k_{t_2} \times k_t}$ in the interior nodes $t = t_1 \dot{\cup} t_2$. The storage complexity is with $n := \max_{\mu \in \mathcal{L}(T_D)} n_\mu$ and $k := \max_{t \in T_D} k_t$ given by

$$\text{Storage}\left((U_\mu)_{\mu \in \mathcal{L}(T_D)}, (B_t)_{t \in \mathcal{I}(T_D)}\right) = \mathcal{O}\left(dnk + (d-1)k^3\right)$$

and hence, linear in the dimension. If an $\mathcal{H}$-Tucker approximation with $\mathcal{H}$-rank $\underline{k}$ is founded by Z, then there exist robust algorithms that approximate Z by a tensor $\overline{Z}$ in this format with smaller $\mathcal{H}$-rank $\underline{r}$. In [9] an error bound for a truncation by the $\mathcal{H}$-SVD is determined in terms of the best $\mathcal{H}$-Tucker approximation $\widetilde{Z}$ with $\mathcal{H}$-rank $\underline{r}$:

$$\left\|Z - \overline{Z}\right\|_2 \leq \sqrt{2d-3}\left\|Z - \widetilde{Z}\right\|_2.$$

It is calculated in $\mathcal{O}(dnk^2 + (d-1)k^4)$ with k and n from above.
The linearity in the dimension and the existence of robust truncation algorithms make the $\mathcal{H}$-Tucker format to an often used tensor format. For a more detailed introduction, also of other tensor formats, we refer the reader to [13].

3 Sampling rules

In the introduction we describe the tensor reconstruction problem in a general form. Now we specify it to the $\mathcal{H}$-Tucker format:

> Let the original tensor A fulfill the restriction (2). Then the task is to find an $\mathcal{H}$-Tucker tensor $A_{\mathcal{H}}$ with a canonical tree T_D and an $\mathcal{H}$-rank $\underline{k}$ with $k_t \leq K$ that approximates A in the known entries of Ω as good as possible:

$$A^*_{\mathcal{H}} = \underset{A_{\mathcal{H}} \in \mathcal{HT}(T_D, \underline{k})}{\operatorname{argmin}} \|A - A_{\mathcal{H}}\|_\Omega \quad \text{with} \quad \|Z\|_\Omega := \sqrt{\sum_{\mathbf{i} \in \Omega} Z_{\mathbf{i}}^2}. \tag{4}$$

We will see in subsection 4.1 that the $\mathcal{H}$-rank $\underline{k}$ can be determined in an adaptive way if a rank bound K is known. In this case the reconstruction of the original tensor

is successful. In the case where K is not known, we assume that $A^*_{\mathcal{H}}$ is a convenient approximation of A if the error in Ω does not exceed a given tolerance. In general we can verify this by determining the error outside of Ω, but due to the problem scenario only informations in Ω are given. A convenient possibility to handle this situation is to generate a test set. Therefore, we split the measured set Ω into a calculation set Ω_C, used to compute the approximation $A_{\mathcal{H}}$, and a test set Ω_T, only used to check the approximation. The quality of the reconstruction does not only depend on the amount of the sampled points in Ω_C but also on the structure of Ω_C as exemplified in the next subsection.

3.1 Why a sampling rule

In the introduction we remark that a standard choice for the sampling points are pseudorandom or quasirandom points coming from MC or QMC methods. Thereby, the points are for example uniformly distributed. If they are generated by QMC they provide greater uniformity.
We will now assume that a random sampling with uniformly distribution took place. Furthermore we assume that $\sharp I = n^d$ for simplicity and $\sharp\Omega_C \leq n^{\lceil \frac{d}{2} \rceil}$. For the reconstruction process, we have in mind that we want to use the matrix hierarchy of the format and hence, do not consider the direct optimization over the representation system $\big((U_\mu)_{\mu\in\mathcal{L}(T_D)}, (B_t)_{t\in\mathcal{I}(T_D)}\big)$. The probability is almost one that in every column at most one entry is known irrespective of the t-matricization under consideration. Let J_t^Ω be the set of columns with a known entry, i_j^t the corresponding rows and I_t^Ω the set of the rows with known entries. Then, we can determine a rank-1-approximation $M_t \in \mathbb{R}^{I_t \times I_{\hat{t}}}$ which is exact in the given entries:

$$M_t = u_t\, v_t^T \quad \text{with} \quad (u_t)_i = \begin{cases} 1, & i \in I_t^\Omega \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad (v_t)_j = \begin{cases} (A^{(t)})_{i_j^t, j}, & j \in J_t^\Omega \\ 0, & \text{otherwise} \end{cases}$$

This examination leads to a $\mathcal{H}$-Tucker tensor $A_{\mathcal{H}}$ with ($D = s \dot\cup \hat{s}$):

$$\begin{aligned} &U_\mu = u_\mu \quad \forall \mu \in \mathcal{L}(T_D) \\ &B_t \in \mathbb{R} \text{ with } B_t = \begin{cases} = \frac{\sum_{i\in\Omega_C} A_i}{\sharp\Omega_c}, & t = \hat{s}(\text{best choice}) \\ = 1, & \text{otherwise} \end{cases} \qquad \forall t \in \mathcal{I}(T_D) \end{aligned}$$

The error $\|A - A_{\mathcal{H}}\|_\Omega$ is the standard deviation of the known entries. It is unlikely that the reconstruction in the unknown entries is successful.
A sampling rule that chooses a more convenient set Ω_C is an important aspect of our algorithm. Therefore, we use a matrix approximation technique extended to our tensor format as described in [14]. It is a specified form of the adaptive cross approximation ACA of [15] explained in the next Subsection 3.2 and changed to a non-adaptive strategy in Subsection 3.3.

3.2 Cross approximation

The name of the technique implicates to define a cross over a matrix element called pivot element. Let $M_{p_1,q_1} \neq 0$ be a chosen pivot element of the matrix $M \in \mathbb{R}^{I\times J}$ with $\sharp I = m_1$ and $\sharp J = m_2$. Then, the column $M|_{I\times q_1}$, the pivot element and the row $M|_{p_1\times J}$ define a cross as shown in Figure 4.

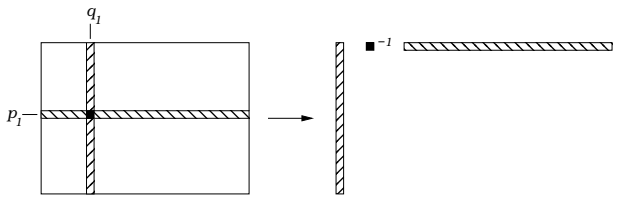

Figure 4: Left: Define a pivot element $M_{p_1,q_1} \neq 0$ with the pivot column $M|_{I\times q_1}$ and the pivot row $M|_{p_1\times J}$. Right: Then a rank-1 cross approximation of M is given by equation (5) visualised here.

We obtain a rank-1 cross approximation of M by:

$$\widetilde{M_1} := M|_{I\times q_1} \cdot \left(M|_{p_1\times q_1}\right)^{-1} \cdot M|_{p_1\times J}\,. \tag{5}$$

This approach will be improved by a rank-1 cross approximation of the remainder $M - \widetilde{M_1}$, leading to a rank-2 cross approximation, and so on. If we select r pivot elements and define $P_r := \{p_1,\ldots,p_r\}$ and $Q_r := \{q_1,\ldots,q_r\}$ then a rank-r approximation is described by

$$\widetilde{M_r} := M|_{I\times Q_r} \cdot \left(M|_{P_r\times Q_r}\right)^{-1} \cdot M|_{P_r\times J}\,. \tag{6}$$

Thereby, the following question arises: Is there an error estimation for $M - \widetilde{M_r}$? A known existence result for an approximation similar to $\widetilde{M_r}$ is given in [16]: Assume that $M, F \in \mathbb{R}^{m_1\times m_2}$, $\operatorname{rank}(M-F) \leq r$ and $\|F\|_2 \leq \varepsilon$ for some $\varepsilon > 0$. Then there exist r columns $C := M|_{I\times Q_r}$ and r rows $R := M|_{P_r\times J}$ which determine a pseudo skeleton component CGR with $G \in \mathbb{R}^{r\times r}$ such that

$$\|M - CGR\|_2 \leq \varepsilon\,(1 + 2\sqrt{rm_1} + 2\sqrt{rm_2})\,.$$

If G is the inverted intersection of C and R then the error gets the additional term $\varepsilon\left(1 + 5r\sqrt{m_1\,m_2}\right)$. The same authors motivate in [17] and [18] the idea to identify G with an $(r\times r)$ - submatrix in M of maximal determinant in modulus called maximum volume. The main results therein are build upon the knowledge of the rank of M. In [15] a strategy is given that can handle matrices without knowing their ranks. A rank-h submatrix is searched in an adaptive way by rank-1 approximations described above with the following full pivot choice (full ACA):

The pivot element with index (p_h, q_h) in step h is best chosen if

$$\left|\left(M - \widetilde{M}_{h-1}\right)_{p_h,q_h}\right| \geq \left|\left(M - \widetilde{M}_{h-1}\right)_{i,j}\right| \quad \forall (i,j) \in I\times J,\, h = 1,\ldots,r \tag{7}$$

with $\widetilde{M}_0 = 0 \in \mathbb{R}^{I \times J}$.

If the matrix M has rank r then the strategy finds a representation after r steps. But until r is reached an error estimation for the rank-h approximation is not known. In [14] this procedure is reduced to the search of only quasi-maximal elements and adapted to the $\mathcal{H}$-Tucker format.

3.3 The non-adaptive sampling rule

The assumption (2) gives an upper bound for every matricization rank $k_t \leq K$. The cross approximation finds a representation

$$A^{(t)} = A^{(t)}\Big|_{I_t \times P_{\hat{t}}} \cdot \left(A^{(t)}\Big|_{P_t \times P_{\hat{t}}} \right)^{-1} \cdot A^{(t)}\Big|_{P_t \times I_{\hat{t}}} =: C_t \; S_t^{-1} \; R_t \tag{8}$$

after k_t steps and the relation $span(C_t) = span(A^{(t)})$ holds. It is meaningful to restrict the pivot selection for the cross approximation in the sons of t based on the following lemma.

Lemma 6 *Let $t = t_1 \dot{\cup} t_2$ be an interior node of the canonical tree T_D and $span(C_t) = span(A^{(t)})$. Then C_t contains also the structure of $A^{(t_1)}$ and $A^{(t_2)}$:*

$$span\left((C_t)^{(t_1)}\right) = span(A^{(t_1)}) \quad and \quad span\left((C_t)^{(t_2)}\right) = span(A^{(t_2)})$$

Proof: Select an arbitrary column $A_j^{(t)}$ of the matricization $A^{(t)}$, which is determined by the columns of C_t. The t_1-matricization of $A_j^{(t)}$ leads to vectors $(A_j^{(t)})_{j_2}^{(t_1)} \in \mathbb{R}^{I_{t_1}}$ for $j_2 \in I_{t_2}$ described by

$$(A_j^{(t)})_{j_2}^{(t_1)} = A_{(j_2,j)}^{(t_1)} = \sum_{\nu=1}^{k_t} c_\nu^j \left((C_t)_\nu^{(t_1)}\right)_{j_2}$$

The t_2-matricization of $A_j^{(t)}$ leads to a similar formulation. ∎

Lemma 6 motivates to select the pivot elements for the son cross approximation from the pivot columns $P_{\hat{t}} \subset I_{\hat{t}}$ of the father.

We start the sampling strategy from the top of the dimension tree. The root D has only one column. Hence, we choose the first pivot elements in a son t of the root. We randomly generate K_t admissible pivot elements (p_i^t, q_i^t) in the sense that they are in different rows and columns, and so $\sharp P_t = \sharp P_{\hat{t}} = K_t$. The second son $\hat{t}$ of the root can use the same elements due to the relation $A^{(t)} = A^{(\hat{t})^T}$. Afterwards, in both sons of t (respectively $\hat{t}$), the pivot selection is restricted as seen in Algorithm 1.

It is done equivalently for the other son t_2, and for their sons and so on. This restricted pivot selection is the core of the sampling rule.

Algorithm 1 Restricted pivot selection in son t_1

1: Given: t_1, brother t_2 and father columns $P_{\hat{t}}$
2: **for** $j = 1, \ldots, K_{t_1}$ **do**
3: Choose randomly and admissible: $p_j \in I_{t_1}$ and $q_j \in I_{t_2} \times P_{\hat{t}}$
4: $P_{t_1} = P_{t_1} \cup p_j$ and $P_{\hat{t}_1} = P_{\hat{t}_1} \cup q_j$
5: **end for**
6: Return: P_{t_1} and $P_{\hat{t}_1}$

Remark 7 (Rank initialisation) *In most situations the representation $\mathcal{H}$-rank $\underline{k}$ or the rank bound K of the original tensor is not known. Hence, the initialisation ranks K_s should be chosen large enough for all $s \in T_D$. If the approximations do not achieve a given tolerance the first step is to increase the ranks K_s. It is obvious that in this case we do not discard the entry set Ω of the first run. With the pivot selection strategy (7), we can estimate the cross approximation of $A^{(s)}$ by the last selected pivot element. If a second interaction to obtain tensor entries is possible, we use this information to increase the entry set for every node $s \in T_D$ individually. Then, we start the reconstruction again.*

4 Reconstruction

The hierarchical Tucker format with its binary tree structure enables us to use matrix techniques for calculations in the tensor case. We choose the cross approximation and restrict the choice of the pivot elements. The next subsection contains the calculation of the format ingredients.

4.1 Main theorem

Theorem 8 (Representation) *Let a cross approximation of $s \in T_D$ be given by*

$$A^{(s)} = C_s \, S_s^{-1} \, R_s + F_s$$

with C_s, S_s and R_s of equation (8) and an error matrix $F_s \in \mathbb{R}^{I_s \times I_{\hat{s}}}$ for two sons $t_1, t_2 \in T_D \setminus D$ of an interior node $t \in \mathcal{I}(T_D) \setminus root(T_D)$. Then a representation of $A^{(t)}$ in the selected columns $P_{\hat{t}}$ is described by:

$$\left(A^{(t)}\Big|_{I_t \times P_{\hat{t}}} \right)_{(i_1,i_2),j} = \sum_{\alpha=1}^{k_{t_1}} \sum_{\beta=1}^{k_{t_2}} (B_t)_{\alpha,\beta,j} \, (C_{t_1})_{i_1,\alpha} \, (C_{t_2})_{i_2,\beta} + (F_t)_{(i_1,i_2),j}$$

$$
\begin{aligned}
\textit{with} \quad (B_t)_{\alpha,\beta,j} &= \sum_{p_1 \in P_{t_1}} \sum_{p_2 \in P_{t_2}} \left(S_{t_1}^{-1}\right)_{\alpha,p_1} \left(A^{(t)}\Big|_{(P_{t_1} \times P_{t_2}) \times P_{\hat{t}}} \right)_{(p_1,p_2),j} \left(S_{t_2}^{-1}\right)_{\beta,p_2} \\
\textit{and} \quad (F_t)_{(i_1,i_2),j} &= \frac{1}{2} \left(\sum_{p_1 \in P_{t_1}} \left(C_{t_1} S_{t_1}^{-1}\right)_{i_1,p_1} (F_{t_2})_{i_2,(p_1,j)} + (F_{t_1})_{i_1,(i_2,j)} \right. \\
&\quad \left. + \sum_{p_2 \in P_{t_2}} \left(C_{t_2} S_{t_2}^{-1}\right)_{i_2,p_2} (F_{t_1})_{i_1,(p_2,j)} + (F_{t_2})_{i_2,(i_1,j)} \right)
\end{aligned}
$$

for all $(i_1, i_2) \in I_t$ *and* $j \in P_{\hat{t}}$.

Proof: We prove the theorem by first rearranging $A^{(t)}$ to $A^{(t_1)}$ and use the cross approximation above and second rearranging R_{t_1} to $R_{t_1}^{(t_2)}$ and use again the cross approximation:

$$
\begin{aligned}
&\left(A^{(t)}\Big|_{I_t \times P_{\hat{t}}} \right)_{(i_1,i_2),j} \\
&= \sum_{p_1 \in P_{t_1}} \sum_{\alpha=1}^{k_{t_1}} (C_{t_1})_{i_1,\alpha} \left(S_{t_1}^{-1}\right)_{\alpha,p_1} \left(A^{(t_1)}\Big|_{P_{t_1} \times (I_{t_2} \times P_{\hat{t}})} \right)_{p_1,(i_2,j)} + (F_{t_1})_{i_1,(i_2,j)} \\
&= \sum_{p_1} \sum_{\alpha} (C_{t_1})_{i_1,\alpha} \left(S_{t_1}^{-1}\right)_{\alpha,p_1} \left[\sum_{p_2 \in P_{t_2}} \sum_{\beta=1}^{k_{t_2}} (C_{t_2})_{i_2,\beta} \left(S_{t_2}^{-1}\right)_{\beta,p_2} \left(A^{(t_2)}\Big|_{P_{t_2} \times (P_{t_1} \times P_{\hat{t}})} \right)_{p_2,(p_1,j)} \right. \\
&\qquad \left. + (F_{t_2})_{i_2,(p_1,j)} \right] + (F_{t_1})_{i_1,(i_2,j)}
\end{aligned}
$$

The equivalence $A^{(t_2)}\big|_{P_{t_2} \times (P_{t_1} \times P_{\hat{t}})} = A^{(t)}\big|_{(P_{t_1} \times P_{t_2}) \times P_{\hat{t}}}$ gives the description of the transfer tensor. The expression of the error matrix in the theorem is symmetric in the sons of t. Therefore, we change the role of t_1 and t_2, leading to the same description of the transfer tensor as before, and take the average. ■

Corollary 9 (Reconstruction) *If in Theorem 8 the matrices* S_s *fulfill:*

$$\textit{rank}(S_s) = \textit{rank}(A^{(s)})$$

for all $s \in T_D$*, then all error matrices* F_s *vanish and the reconstruction is successful.*

We calculate the transfer tensor B_t with $t = t_1 \dot{\cup} t_2$ by inverting S_{t_1} and S_{t_2}. If the initialisation ranks K_{t_1} and K_{t_2} are large enough, then these matrices are singular or ill-conditioned. This problem is handled by truncating the approximations.

Remark 10 (Truncate the Approximation) *Let* S_s *be singular or ill-conditioned. We approximate* S_s *by the cross approximation with pivot selection of (7) until the current pivot element in modulus is smaller than a given tolerance. Then,* P_s *and* $P_{\hat{s}}$ *with* $\sharp P_s = \sharp P_{\hat{s}} = K_s$ *are reduced to the subsets* $\overline{P}_s$ *and* $\overline{P}_{\hat{s}}$ *with* $\sharp \overline{P}_s = \sharp \overline{P}_{\hat{s}} = k_s$ *and we found a reduced cross representation* $\overline{S}_s$*. Thereby, the calculation in Theorem 8 does not change. We only replace* P_s *by* $\overline{P}_s$*,* $P_{\hat{s}}$ *by* $\overline{P}_{\hat{s}}$ *and* F_s *by* $\overline{F}_s$ *for* $s \in \{t_1, t_2, t, \hat{t}\}$*. To keep the notation as simple as possible we skip the over lined mark if not needed.*

To sum up, we obtain a frame of a leaf s from the tensor entries of $I_s \times P_{\hat{s}}$ and the transfer tensor of an interior node $t = t_1 \dot{\cup} t_2$ from the entries of $P_{t_1} \times P_{\hat{t}_1}$, $P_{t_2} \times P_{\hat{t}_2}$, and the coupled set $P_{t_1} \times P_{t_2} \times P_{\hat{t}}$.

4.2 Tensor Reconstruction Algorithm with a non-adaptive sampling rule

The reconstruction process starts with the detemination of Ω_C by the non-adaptive sampling rule of Algorithm 2 calling a function 'determine_omegac' of Algorithm 3. After the entries are sampled, the ingredients of $A_{\mathcal{H}}$ are recursively calculated by the function 'ht_ingredients' of Algorithm 4 beginning with the call 'ht_ingredients(D, ht, Ω_C,A_{Ω_C})'.

Algorithm 2 Determine Ω_C

1: Given: canonical tree T_D, root $D = t \dot{\cup} \hat{t}$ and $\hat{t} = \hat{t}_1 \dot{\cup} \hat{t}_2$
2: Left son t: determine_omegac(t, Ω_C)
3: Right son $\hat{t}$: determine_omegac$(\hat{t}_1, \Omega_C)$ and determine_omegac$(\hat{t}_2, \Omega_C)$
4: Return: sampling set Ω_C

Algorithm 3 determine_omegac(t, Ω_C)

1: Given: node t, set Ω_C and father pivot columns $P_{\hat{f}}$ if t is not direct son of D
2: **if** t is the left son of D **then**
3: unrestricted pivot selection
4: **else**
5: restricted pivot selection of Algorithm 1
6: **end if**
7: **if** t has sons t_1 and t_2 **then**
8: determine_omegac(t_1, Ω_C) and determine_omegac(t_2, Ω_C)
9: $\Omega_C = \Omega_C \cup (P_{t_1} \times P_{t_2} \times P_{\hat{t}}) \cup (P_t \times P_{\hat{t}})$
10: **else**
11: $\Omega_C = \Omega_C \cup (I_t \times P_{\hat{t}})$
12: **end if**
13: Return: Ω_C

In the next section we test the reconstruction for a perturbed and an unperturbed high dimensional problem.

5 Numerical Examples

We consider tensors where we know a low rank approximation in the $\mathcal{H}$-Tucker format. The reasons are first of all the relative 2-norm L_2 of an $\mathcal{H}$-Tucker tensor is directly determinable as seen in [9]. Second, the possibility to obtain the tensor entries of our special sampling rule. The calculation of the relative maximum error is not applicable.

Algorithm 4 ht_ingredients(t, ht, Ω_C,A_{Ω_C})

Given: node t, structure of ht to fill, sampling set Ω_C and entry set A_{Ω_C}
if t has sons t_1 and t_2 **then**
  reduce S_{t_1} to $\overline{S}_{t_1}$ with the full ACA
  if t is the root D **then**
    $B_t = \left(\overline{S}_{t_1}\right)^{-1}$
  else
    reduce S_{t_2} to $\overline{S}_{t_2}$ with the full ACA and calculate B_t (Theorem 8)
  end if
  ht_ingredients(t_1, ht, Ω_C,A_{Ω_C}) and ht_ingredients(t_2, ht, Ω_C,A_{Ω_C})
else
  $U_t = C_t$ (Theorem 8)
end if
Return: ht

Thus, we randomly generate a test set Ω_T of 10^5 entries and calculate the maximum error L_∞ over the subset Ω:

$$L_2 := \frac{\|A - A_{\mathcal{H}}\|_2}{\|A\|_2} \quad \text{and} \quad L_\infty := \frac{\|A - A_{\mathcal{H}}\|_{\infty,\Omega}}{\|A\|_{\infty,\Omega}} \tag{9}$$

The numerical experiments are focused on the following questions:

1. Does the restricted pivot selection of Algorithm 1 motivated by Lemma 6 lead to a suitable approximation of the original tensor, or even of one matricization?
2. Does the reconstruction algorithm find the representation ranks $\underline{k}$, if the original tensor A fulfills assumption (2) as in Example 12, or convenient approximation ranks, if A does not fulfil (2) as in Example 11?
3. Does the reconstruction in the perturbed case achieve an accuracy close to the perturbation?

For simplicity, we set the mode sizes all equal to $n_\mu = n$. We choose the discretization of a smooth function as the first example.

Example 11 (Smooth function) *The d-dimensional smooth function*

$$f(x) := \frac{\sqrt{d}}{\|x\|_2} \quad \textit{with} \quad x_i \in [1,2] \quad \forall i = 1,\dots,d$$

has function values in the interval $\left[\frac{1}{2}, 1\right]$. The discretization over a tensor grid leads to a tensor A given entrywise by

$$A_{i_1,\dots,i_d} := \sqrt{d}\left(\sum_{\mu=1}^{d} \left(m_{i_\mu}\right)^2\right)^{-\frac{1}{2}} \quad \textit{with } m_{i_\mu} := 1 + \frac{i_\mu - 1}{n_\mu - 1} \textit{ and } i_\mu \in I_\mu \, \forall \mu. \tag{10}$$

It has a very close approximation $A^E \in \mathbb{R}^I$ *with* $\left\|A - A^E\right\|_\infty \leq 7.315\,10^{-10}$ *[19] by exponential sums*

$$A^E{}_{i_1,\ldots,i_d} := \sum_{j=1}^{K} w_j \prod_{\mu=1}^{d} exp\left(\frac{-\,m_{i_\mu}^2\,\alpha_j}{d}\right) \quad \forall (i_1,\ldots,i_d) \in I$$

with weights $(w_j)_{j=1,\ldots,K}$ *and exponential parts* $(\alpha_j)_{j=1,\ldots,K}$ *coming from the webpage 'http://www.mis.mpg.de/scicomp/EXP_SUM'. In [9] it is converted to an* $\mathcal{H}$*-Tucker tensor* $A^E_{\mathcal{H}}$ *and used here, instead of* A*, to calculate the* L_2 *errors of (9) directly.*

We start the numerical experiments with an unperturbed tensor. In the first test we consider the sampling strategy for the matricization of the left root son $t = t_1 \dot\cup t_2$ and its son t_1 for dimension $d = 8$ and mode size $n = 10$. We run the sampling 100 times and evaluate the L_∞ error of the cross approximation with $\sharp\Omega_T = 10^4$. In Figure 5 we plot a histogram of the L_∞ error.It shows that the cross approximation error with an unrestricted pivot selection in t lies with 68% mostly in the interval $(10^{-9}, 10^{-8}]$. For the son node t_1 the cross approximation error with the restricted selection of Algorithm 1 is in two-thirds of the runnings in $(10^{-11}, 10^{-9}]$. Hence, the example satisfies the positive expectations on the restricted pivot selection.

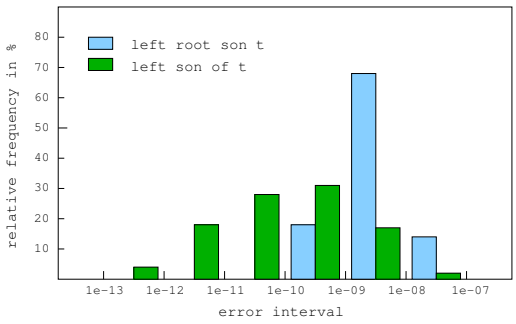

Figure 5: Cross approximation error for tensor A (10) with $n = 10$ and $d = 8$

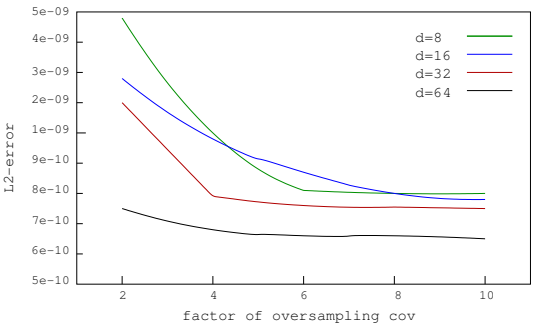

Figure 6: Reconstruction error for tensor A (10) with $n = 10$ and $d = 8, 16, 32, 64$

In the next test we consider the dependence of the reconstruction L_2 error from the dimension and the oversampling factor c_{ov} as the ratio between the number of sampled points and the number of variables to represent $A_{\mathcal{H}}$:

$$c_{ov} = \frac{\sharp\,\Omega_C}{\sharp \text{ Variables of } A_{\mathcal{H}}}.$$

We run the algorithm 10 times and take the average error. In Figure 6 we plot the L_2 error against the oversampling c_{ov} and vary the dimension $d = 8, 16, 32, 64$. The error difference between all dimensions is smaller than $3e-10$. For $c_{ov} = 2$ the L_2 error is around $5e-09$ and falls down to $1e-09$ for $c_{ov} = 6$. The error seems to be independent of the dimension and an oversampling factor higher than 6 seems to not improve the approximation.

To answer the second question, we consider the approximation ranks of A in (10) for $n = 10$ and vary the dimension $d = 8, 16, 32, 64$ in Table 1. We select $c_{ov} = 6$ corresponding to an L_2 error in $(1e-10, 1e-09)$. The orginal tensor A does not fulfil assumption (2). For the comparison with $A_{\mathcal{H}}$, we truncate the approximation tensor $A^E_{\mathcal{H}}$ to a node wise accuracy of $1e-10$ and consider the resulting ranks represented by the efficient rank r_{eff} to the same storage complexity $\mathcal{O}\left((d-1)r^3_{eff} + dnr_{eff}\right)$. The rank difference $rd := k_{eff} - r_{eff}$ to the efficient rank k_{eff} of the reconstruction decreases from $rd = 1.7$ for $d = 8$ to $rd = 1.02$ for $d = 64$. We summarize that the reconstruction finds approximation ranks close to the quasi-optimal ranks of the same accuracy.

d	r_{reff}	k_{eff}	rd
8	5.3	7	1.7
16	4.47	6.05	1.58
32	4.08	5.1	1.02
64	3.52	4.54	1.02

Table 1: Comparison of the efficient rank r_{eff} to node wise accuracy $1e-10$ and the reconstruction efficient rank k_{eff} of A (10) for $n = 10$ and $c_{ov} = 6$

c_{ov}	L_2	L_∞
2	8.5e-10	8.3e-09
4	7.57e-10	2e-09
6	7.57e-10	1.7e-09
8	7.57e-10	1.69e-09
10	7.56e-10	1.5e-09
12	7.56e-10	1.17e-09

Table 2: Comparison of L_2 error and L_∞ error of tensor A (10) for $d = 32$ and $n = 10$

In Table 2 we compare the L_2 error and the L_∞ error for fixed $d = 32$ and vary the oversampling. We see that the L_2 error is at most a factor of 10 better than the L_∞ error and conclude that they are close to each other [20].
We continue with a perturbed tensor $A^p := A + X$ where the tensor A has the form (10) and the perturbation is entry wise given by:

$$X_{\mathbf{i}} = \begin{cases} \text{random } \beta_{\mathbf{i}} \in [-1e-5, 1e-5], & \text{with probability } p = 0.01 \\ 0, & \text{with probability } p = 0.99 \end{cases}$$

We want to answer the third question from above for $d = 8$ and $d = 16$ where we fix $n = 10$. The algorithm stops the full ACA step to calculate the reduced matrix of S with rank r if $\left|(S - \widetilde{S}_r)_{(p_{r+1}, q_{r+1})}\right| \leq 1e-5$ and the matrix S is reduced to $\overline{S}$ as described in Remark 10. We consider the L_∞ error in Figure 7 and the L_2 error in Figure 8 in a histogram for 10 test runnings. The L_∞ error is with at least 80% in $(1e-4, 1e-3]$ irrespective of the dimension under consideration. The L_2 error is for $d = 16$ always in $(1e-5, 1e-4]$ and for $d = 8$ with 80% mostly in $(1e-6, 1e-5]$. We conclude that the reconstruction error of a perturbed tensor is close to the perturbation.

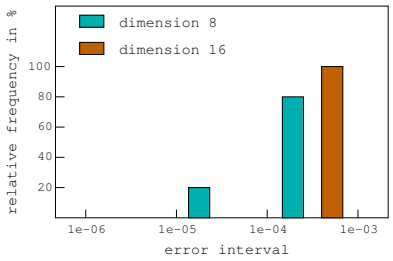

Figure 7: L_∞-error for the perturbed Tensor of (10) with $n = 10$

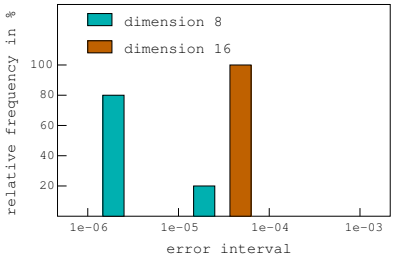

Figure 8: L_2-error for the perturbed Tensor of (10) with $n = 10$

The second example is a random $\mathcal{H}$-Tucker tensor.

Example 12 ($\mathcal{H}$-Tucker tensor) *The random $\mathcal{H}$-Tucker tensor have constant ranks $K = 7$ for all nodes $t \in T_D$ fulfilling assumption (2). The ingredients of the format are randomly generated with a uniform distribution in $\left[\frac{1}{2}, 1\right]$. Afterwards, the node wise singular values are forced to decay like α^j with $j = 1, \ldots, K$ and $\alpha \in (0, 1)$.*

We consider the cross approximation error in the matricization of the left root son t and the reconstruction error for 100 test runnings with $d = 8$, $n = 10$, $c_{ov} = 3$ and $\sharp\Omega_T = 10^5$. We call the reconstruction successful if the reconstruction error is in $(1e-15, 1e-13]$. In the first 50 runnings we set $\alpha = 0.25$ shown in Figure 9 and in the second 50 runnings $\alpha = 0.75$ shown in Figure 10 by histograms again. The singular values decrease for $\alpha = 0.25$ faster than in the case $\alpha = 0.75$. The cross approximation error is always smaller than $1e-11$ and often better than $1e-15$. The reconstruction error is always better than $1e-7$ and most of the time smaller than $1e-13$. Figure 9 shows a few test runnings with an error in $(1e-9, 1e-7]$. Therein, the resulting rank k_t of the observed node t is every time 5 instead of 7. In all other cases the $\mathcal{H}$-rank $\underline{k}$ of the representation $A_{\mathcal{H}}$ is node wise in $\{6, 7\}$. In summary, the reconstruction algorithm mostly finds the representation ranks of the original tensor.

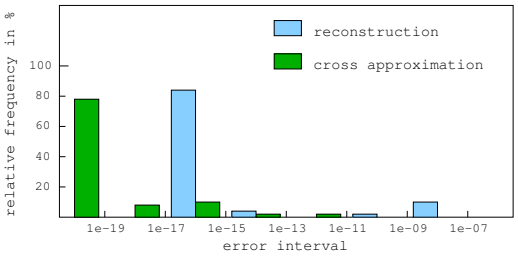

Figure 9: Random $\mathcal{H}$-Tucker tensor with $d = 8, n = 10$ and 50 test runnings for $\alpha = 0.25$

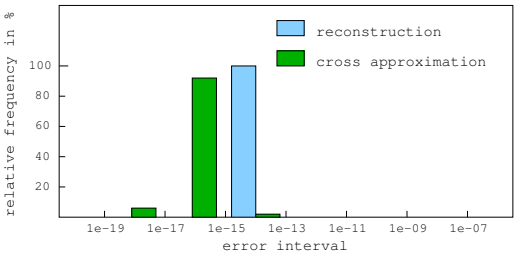

Figure 10: Random $\mathcal{H}$-Tucker tensor with $d = 8, n = 10$ and 50 test runnings for $\alpha = 0.75$

6 Conclusion

In this article we present a non-adaptive sampling rule for high-dimensional tensor reconstruction based on an approximation scheme in the $\mathcal{H}$-Tucker format. For tensors with $\mathcal{H}$-rank $\underline{k}$ the developed algorithm finds a reconstruction if the pivot matrix fulfills $\operatorname{rank}(S_t) = k_t$ for all nodes t of the canonical tree T_D. The numerical examples show that the reconstruction was successful, even if the resulting ranks are only close to the original ones. In the case where the tensor A has no low rank representation the algorithm found convenient approximation ranks.
In contrast to the non-adaptive sampling rule, the adaptive rule needs the interaction with the generation routine during the whole time. If this interaction is possible, an adaptive sampling as in [14] may lead to higher approximation accuracy.
In the field of tensor completion where one interaction is possible, the non-adaptive strategy may be useful.

References

[1] B. N. Khoromskij and I. V. Oseledets. Quantics-tt collocation approximation of parameter-dependent and stochastic elliptic pdes. *Comp. Meth. in Applied Math.*, 10(4):376394, 2010.

[2] A. Litvinenko, H. G. Matthies, and T. A. El-Moselhy. Low-rank tensor approximation of the response surface. *accepted by MCQMC*, 2013.

[3] R. E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numer.*, 7:1–49, 1998.

[4] J. Ballani. Fast evaluation of singular BEM integrals based on tensor approximations. *Numer. Math.*, 121(3):433–460, 2012.

[5] C. Da Silva and F. J. Herrmann. Hierarchical tucker tensor optimization - applications to tensor completion. SampTA 2013, 10th International Conference on Sampling Theory and Application, Jacobs University Bremen, 2013.

[6] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, March 2011.

[7] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *The journal of Fourier analysis and applications*, 15(5):706–722, 2009.

[8] C. Tobler. *Low-rank Tensor Methods for Linear Systems and Eigenvalue Problems.* PhD thesis, ETH Zürich, 2012.

[9] L. Grasedyck. Hierarchical Singular Value Decomposition of Tensors. *SIAM J. Matrix Anal. Appl.*, 31:2029–2054, 2010.

[10] L. Grasedyck and W. Hackbusch. An introduction to hierarchical (*mathcalH*-) rank and TT-rank of tensors with examples. *Computational methods in applied mathematics*, 11(3):291–304, 2011.

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

[12] J. Ballani and L. Grasedyck. Tree adaptive approximation in the hierarchical tensor format. *DFG SPP 1324 Preprint*, 141, 2013.

[13] W. Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42 of *Springer series in computational mathematics*. Springer, Heidelberg, 2012.

[14] J. Ballani, L. Grasedyck, and M. Kluge. Black box approximation of tensors in hierarchical Tucker format . *Linear Algebra Appl.*, 438(2):639–657, 2013.

[15] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.

[16] S. A. Goreinov, E. E Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Lin. Alg. Appl.*, 261:1–22, 1997.

[17] S. A. Goreinov, E. E Tyrtyshnikov, and N. L. Zamarashkin. Pseudo-skeleton approximations by matrices of maximal volume. *Mathematical Notes*, 62(4):515–519, 1997.

[18] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. How to find a good submatrix. *in: Matrix Methods: Theory, Algorithms,Applications,(World Scientific, Hackensack, NY)*, pages 247–256, 2010.

[19] W. Hackbusch. Entwicklung nach Exponentialsummen. Technical Report TR 4/2005, Max Planck Institute for Mathematics in the Sciences, 2005.

[20] Wolfgang Hackbusch. l^∞ estimation of tensor truncations. *Numerische Mathematik*, 125(3):419–440, 2013.