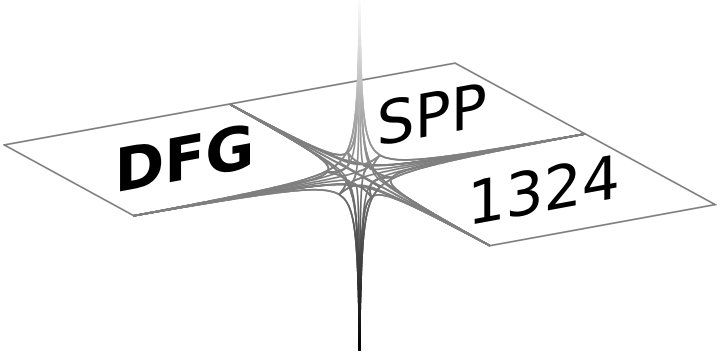
DFG-Schwerpunktprogramm 1324

„Extraktion quantifizierbarer Information aus komplexen Systemen”

Black Box Low Tensor Rank Approximation Using Fibre-Crosses

M. Espig, L. Grasedyck, W. Hackbusch

Preprint 4



Edited by

AG Numerik/Optimierung
Fachbereich 12 - Mathematik und Informatik
Philipps-Universität Marburg
Hans-Meerwein-Str.
35032 Marburg

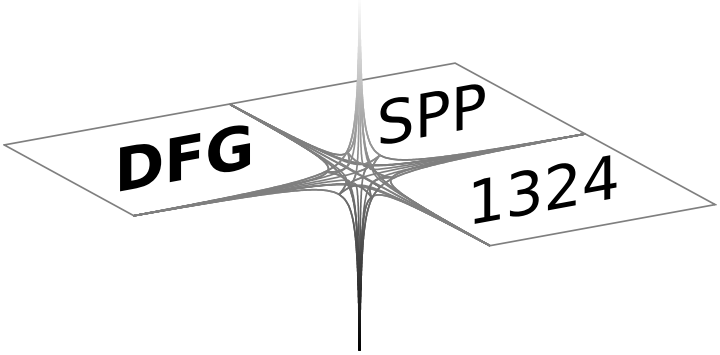
DFG-Schwerpunktprogramm 1324

„Extraktion quantifizierbarer Information aus komplexen Systemen”

Black Box Low Tensor Rank Approximation Using Fibre-Crosses

M. Espig, L. Grasedyck, W. Hackbusch

Preprint 4



The consecutive numbering of the publications is determined by their chronological order.

The aim of this preprint series is to make new research rapidly available for scientific discussion. Therefore, the responsibility for the contents is solely due to the authors. The publications will be distributed by the authors.

Black Box Low Tensor Rank Approximation using Fibre-Crosses

Mike Espig * Lars Grasedyck *† Wolfgang Hackbusch *

October 8, 2008

In this article we introduce a black box type approximation algorithm for tensors A in high dimension d. The algorithm determines adaptively the positions of entries of the tensor that have to be computed or read, and using these (few) entries it constructs a low rank tensor approximation X that minimizes the ℓ_2-distance between A and X at the chosen positions. The full tensor A is not required, only the evaluation of A at a few positions. The minimization problem is solved by Newton's method which requires the computation and evaluation of the Hessian. For efficiency reasons the positions are located on fibre-crosses of the tensor so that the Hessian can be assembled and evaluated in a data-sparse form requiring a complexity of $\mathcal{O}(Pd)$, where P is the number of fibre-crosses and d the order of the tensor.

1 Introduction

Computations with tensors $A \in \mathbb{R}^{n^d}$ require in general a storage complexity in $\mathcal{O}(n^d)$. In order to keep problems tractable for $d \gg 2$ on standard computers, one has to assume some kind of data-sparsity, i.e., that there exists an (approximate) representation X of the tensor A such that X can be described by fewer data. One such format is the low rank format

$$X = \sum_{i=1}^{k} \bigotimes_{\mu=1}^{d} x_{i,\mu}, \tag{1}$$

that allows to store the tensor X in $\mathcal{O}(kdn)$. The number k of addends is the tensor rank and the number d of factors is the order of the tensor (or the dimension). Since the dimension d enters only linearly, this is applicable even in very high dimensions. However, the set of tensors

$$\mathcal{T}(k,d,n) := \Big\{ X \in \mathbb{R}^{n^d} \mid X \text{ allows a representation of the form (1)} \Big\}$$

*Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany
†TU-Berlin, Berlin Mathematical School (BMS), Berlin, Germany

is not a linear space. In particular, when adding two tensors of tensor rank k, the result is in general a tensor of rank $2k$. An elementary step for computations is therefore a projection from a larger rank k to a smaller rank k'. This step has been analysed in detail and it is highly non-trivial: it is neither the case that a best approximation of rank k' always exists [4], nor that it is possible to compute a best approximation if it exists. Nonetheless, available state-of-the-art methods are quite efficient and have proven to be reliable in many practical cases [1, 2, 5].

The situation changes when the tensor A to be approximated in the set $\mathcal{T}(k,d,n)$ is not yet given in low rank format but, e.g. by an explicit formula

$$A_{(i_1,\dots,i_d)} = f(z_{i_1,1},\dots,z_{i_d,d}), \qquad z_{i_\mu,\mu} \in [0,1], \quad i_\mu \in \{1,\dots,n\}, \quad \mu = 1,\dots,d,$$

with a smooth or even analytic function

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

In this case the initial approximation in low tensor rank format is not evident. The only reference — to our knowledge — available in the literature is the higher-dimensional cross approximation [8] which scales exponentially with the dimension d.

We propose a direct minimization of the distance between the tensor A and the set $\mathcal{T}(k,d,n)$, of course with some minor modifications in order to ensure that a local best approximation exists. Then we replace the distance measure by a heuristic approximation that involves only the evaluation of the tensor A in a few points. The choice of the points is adapted to the tensor A and will be determined on-the-fly. The restriction to a small set of points is reasonable, because the low rank format involves only kdn data whereas the whole tensor contains n^d data.

Remark 1 (Relation to Compressed Sensing) *The situation in this article is similar to the setting for compressed sensing with sparse vectors: most of the input data is redundant because of a sparsity assumption for the representation. Our problem is that we cannot take samples (inner products with random tensors) of the input tensor since we want to avoid the evaluation of the input tensor in too many points.*

The article is structured as follows. In the following Section 2 we introduce the direct minimization problem based on a given set of points. Section 3 is concerned with the efficient assembly and evaluation of the gradient and Hessian in data-sparse form. In Section 4 we specify Newton's method for the minimization and in Section 5 we formulate the heuristic black box approximation algorithm. The last Section 6 provides a few numerical examples that show the efficiency and reliability of the proposed approximation algorithm as well as the limitations.

2 Projection to fixed tensor rank

2.1 Basic Definitions

We start this section by introducing some of the basic tensor-related definitions that will be used throughout the article.

Definition 2 (Order, Elementary Tensor, Rank) *Let $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ for integers $n_1, \ldots, n_d, d \in \mathbb{N}$. The integer d is the* order *or dimension of the tensor. An order $d = 2$ tensor is simply a matrix. We call a tensor $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ an* elementary tensor*, if for all $\mu \in \{1, \ldots, d\}$ there exist vectors $x_\mu \in \mathbb{R}^{n_\mu}$ such that the entries of X can be represented in the following way*

$$X_{(i_1,\ldots,i_d)} = \prod_{\mu=1}^{d} (x_\mu)_{i_\mu}, \qquad x_\mu \in \mathbb{R}^{n_\mu}, \qquad i_\mu \in \{1, \ldots, n_\mu\}.$$

We use the short notation

$$X = \bigotimes_{\mu=1}^{d} x_\mu.$$

The rank *k of a tensor $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the minimal number $k \in \mathbb{N}_0$ such that there exist elementary tensors $X_1, \ldots, X_k$ with*

$$X = X_1 + \cdots + X_k = \sum_{i=1}^{k} \bigotimes_{\mu=1}^{d} x_{i,\mu} \qquad x_{i,\mu} \in \mathbb{R}^{n_\mu}. \tag{2}$$

The set of tensors of rank at most k is denoted by

$$\mathcal{T}(k, d) := \{X \in \mathbb{R}^{n_1 \times \cdots \times n_d} \mid \operatorname{rank}(X) \leq k\}$$

The elements of $\mathcal{T}(k, d)$ are called rank *k tensors.*

The definition of (tensor) rank for dimension $d = 2$ coincides with the rank of a matrix. If we write a matrix of rank k as the sum of k rank one matrices, then a tensor of rank k is the straightforward generalization to dimension $d \geq 3$, i.e., the sum of k rank one tensors (elementary tensors). In the literature the terms *PARAFAC rank*, *CANDECOMP rank* or *Kronecker rank* are sometimes used.

Lemma 3 *The storage complexity $N_{St,K}(d, k)$ for a tensor $X \in \mathcal{T}(k, d)$ in the form (2) is*

$$N_{St,K}(d, k) = k(n_1 + n_2 + \cdots + n_d)$$

but the number of degrees of freedom in the representation is only $k(n_1 + n_2 + \cdots + n_d - d + 1)$.

Proof: The storage complexity is trivial since we store only the k elementary tensors X_i in (2), each of which consists of vectors of length $n_1, n_2, \ldots, n_d$. The redundancy in the representation comes from the fact that one can normalize all but one factor to $\|x_{i,\mu}\|_2 = 1$ $(\mu = 1, \ldots, d,\ i = 2, \ldots, k)$. ∎

A different kind of rank will be introduced next. If we write an $n \times m$-matrix R of rank k as the product of an $n \times k$ matrix U with orthonormal columns, a $k \times k$ matrix C and a $k \times m$ matrix V with orthonormal rows,

$$R = UCV,$$

then the Tucker format introduced next is the generalisation of this to higher dimensions $d \geq 3$.

Definition 4 (Tucker-rank, Tucker-format) *The* Tucker-rank *of a tensor* $T \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *is the tupel* $(k_1, \ldots, k_d)$ *with minimal entries* $k_\mu \in \mathbb{N}_0$ *such that there exist vectors* $u_{i,\mu} \in \mathbb{R}^{n_\mu}$ *and a so-called* core tensor $C \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ *with*

$$T = \sum_{i_1=1}^{k_1} \cdots \sum_{i_d=1}^{k_d} C_{(i_1,\ldots,i_d)} \bigotimes_{\mu=1}^{d} u_{i_\mu,\mu} \tag{3}$$

The representation of the form (3) is called the Tucker-format, or shortly T *is called Tucker-tensor.*

The Tucker-rank coincides with the usual matrix rank in the case $d = 2$ in the sense that $k_1 = k_2 = k$. In dimension $d = 2$ one can choose the vectors $u_{i,\mu}$ such that the core tensor is diagonal. This is in general not possible in dimensions $d \geq 3$. The Tucker format (3) is simply the representation of the tensor T in the subspace bases $u_{i_\mu,\mu}$.

Lemma 5 *The storage complexity* $N_{St,T}(d, k_1, \ldots, k_d)$ *for a tensor* X *of Tucker rank* $(k_1, \ldots, k_d)$ *in the representation (3) is*

$$N_{St,T}(d, k_1, \ldots, k_d) = \sum_{\mu=1}^{d} k_\mu n_\mu + \prod_{\mu=1}^{d} k_\mu.$$

Proof: The storage complexity involves two parts, first the k_μ vectors of length n_μ to store the $u_{i,\mu}$ and the core tensor C of size $k_1 \times \cdots \times k_d$. ∎

In dimensions $d \leq 3$ the dominating part of the complexity is the storage of the basis vectors $u_{i,\mu}$, since typically $k_\mu \ll n_\mu$. If the dimension becomes larger, then the complexity grows only linearly in the dimension d for the first term (assuming a moderate increase of the ranks k_μ), whereas the second term $\prod_{\mu=1}^{d} k_\mu$ grows exponentially, thus this format is not suitable for large dimensions $d \gg 3$. However, the subspace basis vectors $u_{i,\mu}$ can be stored in $\mathcal{O}(\sum_{\mu=1}^{d} k_\mu n_\mu)$ so that only the explicit assembly of the core tensor C has to be avoided.

2.2 Decomposition and Approximation of Tensors in Tucker-Format

Definition 6 (Fibre) *Let* $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *and* $\mu \in \{1, \ldots, d\}$. *For a multiindex* $I \in \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_d\}$ *we define*

$$I^j := (I_1, \ldots, I_{\mu-1}, j, I_{\mu+1}, \ldots, I_d), \quad j = 1, \ldots, n_\mu.$$

The set of fibres *of* X *in direction* μ *is defined by*

$$\mathcal{W}(X, \mu) := \{w \in \mathbb{R}^{n_\mu} \mid w_j = X_{I^j},\ j = 1, \ldots, n_\mu\}.$$

Example 7 *In the matrix case* $d = 2$ *the set* $\mathcal{W}(X, 1)$ *contains the column vectors of* X *and* $\mathcal{W}(X, 2)$ *contains the row vectors of* X.

The vectors $u_{i_\mu,\mu}$ used in the representation (3) of a tensor X can be obtained as follows: For each $\mu = 1, \ldots, d$ and $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ we choose an orthonormal basis

$$\text{span}\mathcal{W}(X, \mu) = \text{span}\{w_{1,\mu}, \ldots, w_{k_\mu,\mu}\}, \quad w_{i,\mu} \in \mathbb{R}^{n_\mu}, \quad \mu = 1, \ldots, d.$$

The Tucker representation (3) is then given by

$$X = \sum_{i_1=1}^{k_1} \cdots \sum_{i_d=1}^{k_d} C_{(i_1,\ldots,i_d)} \bigotimes_{\mu=1}^{d} w_{i_\mu,\mu}$$

with the core tensor $C \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ being uniquely determined by X and $w_{i,\mu}$ ($\mu = 1, \ldots, d$, $i = 1, \ldots, k_\mu$). In theory, one can form the whole set $\mathcal{W}(X, \mu)$ and compute an orthonormal basis, but in practice the set is much too large ($\prod_{\nu \neq \mu} n_\nu$ many vectors). For the practical realization we will restrict the set $\mathcal{W}(X, \mu)$ to a small subset of fibres that will be chosen adaptively. We will describe the construction later, for now let us assume that for each μ a subset of orthonormal vectors

$$\mathcal{V}(X, \mu) = \{v_{1,\mu}, \ldots, v_{k_\mu,\mu}\} \subset \text{span}\mathcal{W}(X, \mu) \tag{4}$$

is given.

Remark 8 *So far we have represented a tensor T in the format (3) by use of subspace basis vectors $u_{i,\mu}$. If we want to approximate the tensor T by a best Tucker-rank $(k'_1, \ldots, k'_d)$ approximation $\tilde{T}$, then the vectors $u'_{i,\mu}$ in a used for the representation cannot be described easily (for $d = 2$ they are the largest singular vectors). They are in general not the left dominant singular vectors of the matrix of fibres from $\mathcal{W}(X, \mu)$. By choosing the dominant singular vectors in each direction $\mu = 1, \ldots, d$ and discarding the singular values $\sigma_i^{(\mu)}$, $i = k'_\mu + 1, \ldots, k_\mu$, one introduces an error of the size [3]*

$$\|T - \tilde{T}\|^2 \leq \sum_{\mu=1}^{d} \sum_{i=k'_\mu+1}^{k_\mu} (\sigma_i^{(\mu)})^2$$

which is at most d times the squared best approximation error, i.e. the reduced subspaces $\mathcal{V}(X, \mu)$ formed from the dominant singular vectors lead to an approximation error in the Frobenius norm that is at most $\sqrt{d}$ times the best approximation error.

We conclude that the restriction to subspaces $\mathcal{V}(X, \mu)$ is harmless, because we have a computable error bound and the error bound is close to the best approximation error.

Our goal is to compute directly a low rank tensor approximation of the form (1) where each vector is of the special structure

$$x_{i,\mu} = \sum_{j=1}^{k_\mu} \alpha_{i,j,\mu} v_{j,\mu}, \qquad \alpha_{i,j,\mu} \in \mathbb{R}. \tag{5}$$

This means that we (theoretically) project the tensor into the subspace spanned by the vectors $v_{j,\mu}$ and use a low rank approximation of the corresponding core tensor.

2.3 Approximation by Low Rank Tensors

In the following we consider the representation or approximation of tensors

$$A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$$

by sums of elementary tensors, i.e. in the set $\mathcal{T}(k,d)$ of low rank tensors:

$$A \approx X = \sum_{i=1}^{k} \bigotimes_{\mu=1}^{d} x_{i,\mu}, \qquad x_{i,\mu} \in \mathbb{R}^{n_\mu}. \tag{6}$$

We further restrict this format by requiring that each of the addends $X_i := \bigotimes_{\mu=1}^{d} x_{i,\mu}$ is an elementary tensor of the form (5) based on the selected vectors $\mathcal{V}(A,\mu)$.

Remark 9 *There is one essential difference from the matrix case: a best approximation of A by a sum of k elementary tensors can — in general — not be written as the rank $k-1$ best approximation plus an elementary tensor [7, 11]. For each rank k a different set of elementary tensors might be necessary.*

2.4 Full minimization

The full minimization problem is to find $X = X(\alpha)$ such that the function

$$\tilde{f}(\alpha) := \|A - X\|_2^2 = \langle A - X, A - X \rangle$$

is minimized — the tilde-notation is due to the fact that we will replace $\tilde{f}$ later by a simpler function f. For this minimization of $\tilde{f}$ we would have to access all entries of A. If the tensor A is already given in a suitable form, then this minimization problem can be solved by a Gauss-Newton method [10], by Newton's method [9], by a trust region Newton method [5] or by an alternating least squares algorithm [6, 1]. In our setting however, the tensor A is not yet given in a suitable format and we want to avoid the evaluation of A for too many indices $(i_1, \ldots, i_d)$. Instead, we pick a suitable subset of the fibres.

2.5 Partial minimization

Let $\mathcal{P} := \{I^1, \ldots, I^P\} \subset \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_d\}$ be a set of multiindices. We define the corresponding index sets

$$J^{p,\mu} := \{I \in \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_d\} \mid \forall \nu \in \{1, \ldots, d\} \setminus \{\mu\} : I_\nu = I^p_\nu\}$$

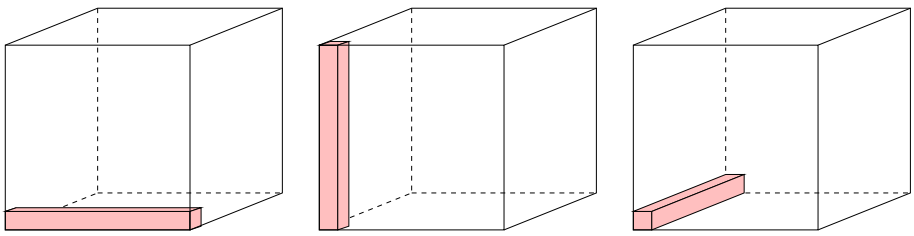


Figure 1: The index sets $J^{p,1}, J^{p,2}, J^{p,3}$ for the pivot index $I^p := (1,1,1)$ in $d = 3$.

and replace the full scalar product $\langle \cdot, \cdot \rangle$ by the bilinearform

$$\langle B, D \rangle_{\mathcal{P}} := \sum_{p=1}^{P} \sum_{\mu=1}^{d} \langle B, D \rangle_{p,\mu} := \sum_{p=1}^{P} \sum_{\mu=1}^{d} \sum_{I \in J^{p,\mu}} B_I D_I.$$

The function to be minimized is now

$$f(\alpha) := \langle A - X, A - X \rangle_{\mathcal{P}}.$$

The bilinearform $\langle \cdot, \cdot \rangle_{\mathcal{P}}$ requires only the evaluation of the tensor A at a small set of indices $i \in J^{p,\ell}$ corresponding to p fibres per direction $\mu = 1, \ldots, d$ of length n_μ.

The choice of the set of pivot indices $\mathcal{P}$ will be described later. For now we assume that, in addition to the vectors $v_{i,\nu}$ used for the representation of X, also the pivots $\mathcal{P}$ are fixed. Our goal is to minimize f over the parameters $\alpha_{i,j,\mu}$ using a Newton type iteration which requires the first and second partial derivatives of f with respect to $\alpha_{i,j,\mu}$.

3 Gradient and Hessian of f

3.1 Partial Derivatives of f

We recall that the orthonormal subspace basis vectors $v_{i,\mu}$ (4), the dimension $d > 1$ and the number of basis vectors per direction k_μ are fixed. Also, the representation (6) and the vectors $x_{i,\mu}$ are fixed.

For all $i^* = 1, \ldots, k$, $\mu^* = 1, \ldots, d$, $j^* = 1, \ldots, k_\mu$ we define the elementary tensors

$$Y_{i^*}^{(\mu^*,j^*)} := \bigotimes_{\mu=1}^{\mu^*-1} x_{i^*,\mu} \otimes v_{j^*,\mu^*} \otimes \bigotimes_{\mu=\mu^*+1}^{d} x_{i^*,\mu}.$$

Let also $\nu^* \in \{1, \ldots, d\} \setminus \{\mu^*\}$ and $m^* \in \{1, \ldots, k_{\nu^*}\}$ be given, and w.l.o.g. $\mu^* < \nu^*$. Then we define

$$Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} := \bigotimes_{\mu=1}^{\mu^*-1} x_{i^*,\mu} \otimes v_{j^*,\mu^*} \otimes \bigotimes_{\mu=\mu^*+1}^{\nu^*-1} x_{i^*,\mu} \otimes v_{m^*,\nu^*} \otimes \bigotimes_{\mu=\nu^*+1}^{d} x_{i^*,\mu}$$

We omit to define $Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)}$ for the case $\nu^* = \mu^*$, because this term will later not be relevant since it is multiplied by zero. We use the short notation for the complementary Kronecker-δ

$$\bar{\delta}_{i,j} := 1 - \delta_{i,j}.$$

Lemma 10 *The first partial derivatives of*

$$f(\alpha) = \langle A - X(\alpha), A - X(\alpha) \rangle_{\mathcal{P}}$$

with respect to the variable α_{i^*,j^*,μ^*} ($i^* \in \{1, \ldots, k\}, j^* \in \{1, \ldots, k_{\mu^*}\}, \mu^* \in \{1, \ldots, d\}$) are

$$\partial_{\alpha_{i^*,j^*,\mu^*}} f(\alpha) = 2 \left\langle X - A, \;\; Y_{i^*}^{(\mu^*,j^*)} \right\rangle_{\mathcal{P}}.$$

The second partial derivatives with respect to the variables α_{i^*,j^*,μ^*} and $\alpha_{\ell^*,m^*,\nu^*}$ ($\ell^* \in \{1, \ldots, k\}, m^* \in \{1, \ldots, k_{\nu^*}\}, \nu^* \in \{1, \ldots, d\}$) are

$$\partial_{\alpha_{\ell^*,m^*,\nu^*}} \partial_{\alpha_{i^*,j^*,\mu^*}} f(\alpha) \;\; = \;\; 2 \left\langle Y_{i^*}^{(\mu^*,j^*)}, \;\; Y_{\ell^*}^{(\nu^*,m^*)} \right\rangle_{\mathcal{P}} + \delta_{i^*,\ell^*} \bar{\delta}_{\mu^*,\nu^*} 2 \left\langle X - A, \;\; Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} \right\rangle_{\mathcal{P}}.$$

Proof: Let $i^* \in \{1,\dots,k\}, \mu^* \in \{1,\dots,d\}$ and $j^* \in \{1,\dots,k_{\mu^*}\}$. Let $\varepsilon \in \mathbb{R}$. We define the coefficients

$$\beta_{i,j,\mu} := \alpha_{i,j,\mu} + \delta_{i^*,i}\delta_{j^*,j}\delta_{\mu^*,\mu}\,\varepsilon$$

and observe (by the multilinearity of the tensor product)

$$\begin{aligned} X(\beta) &= \sum_{i=1}^{k}\bigotimes_{\mu=1}^{d}\sum_{j=1}^{k_\mu}\beta_{i,j,\mu}v_{j,\mu} \\ &= \sum_{i\neq i^*}^{k}\bigotimes_{\mu=1}^{d} x_{i,\mu} + \bigotimes_{\mu=1}^{\mu^*-1} x_{i^*,\mu}\otimes\left(\sum_{j=1}^{k_{\mu^*}}\beta_{i^*,j,\mu^*}\ v_{j,\mu^*}\right)\otimes\bigotimes_{\mu=\mu^*+1}^{d} x_{i^*,\mu} \\ &= X(\alpha) + \bigotimes_{\mu=1}^{\mu^*-1} x_{i^*,\mu}\otimes\varepsilon v_{j^*,\mu^*}\otimes\bigotimes_{\mu=\mu^*+1}^{d} x_{i^*,\mu} \ =\ X(\alpha) + \varepsilon Y_{i^*}^{(\mu^*,j^*)}. \end{aligned}$$

The first partial derivatives of f are now simply (due to the bilinearity of $\langle\cdot,\cdot\rangle_{\mathcal{P}}$)

$$\begin{aligned} \lim_{\varepsilon\to 0}\frac{f(\beta)-f(\alpha)}{\varepsilon} &= \lim_{\varepsilon\to 0}\frac{\langle A - X - \varepsilon Y_{i^*}^{(\mu^*,j^*)}, A - X - \varepsilon Y_{i^*}^{(\mu^*,j^*)}\rangle_{\mathcal{P}} - \langle A-X, A-X\rangle_{\mathcal{P}}}{\varepsilon} \\ &= \lim_{\varepsilon\to 0}\frac{-2\langle A-X, \varepsilon Y_{i^*}^{(\mu^*,j^*)}\rangle_{\mathcal{P}} + \langle \varepsilon Y_{i^*}^{(\mu^*,j^*)}, \varepsilon Y_{i^*}^{(\mu^*,j^*)}\rangle_{\mathcal{P}}}{\varepsilon} \\ &= 2\langle X - A, Y_{i^*}^{(\mu^*,j^*)}\rangle_{\mathcal{P}}. \end{aligned}$$

Analogously we obtain the second partial derivatives. ∎

3.2 Changing the representation

For the remainder of this article we consider low rank tensors $X(\alpha)$ of the form (1,(5)) represented by the basis vectors $v_{j,\mu}$ and the coefficients $\alpha_{i,j,\mu}$. Sometimes, we require the values of $X(\alpha)$ at all points $I \in J^{p,\mu}$. The computation of these values can be done in two steps:

1. First we compute coefficients $\tilde{\alpha}_{p,j,\mu}$ such that
$$X(\alpha)|_{J^{p,\mu}} = \sum_{j=1}^{k_\mu}\tilde{\alpha}_{p,j,\mu}v_{j,\mu} \tag{7}$$
2. Second we compute the linear combinations in (7).

The representation by $\tilde{\alpha}$ will later be used in Newton's method. The coefficients $\tilde{\alpha}$ can be obtained in three steps

$$\alpha^{(1)}_{p,i,\nu} := \sum_{j=1}^{k_\nu}\alpha_{i,j,\nu}(v_{j,\nu})_{I_\nu^p}, \quad \alpha^{(2)}_{p,i,\mu} := \prod_{\nu\neq\mu}\alpha^{(1)}_{p,i,\nu} \quad \tilde{\alpha}_{p,j,\mu} := \sum_{i=1}^{k}\alpha^{(2)}_{p,i,\mu}\alpha_{i,j,\mu}$$

in complexity $\mathcal{O}(Pk\sum_{\mu=1}^{d}k_\mu)$. The second term $\alpha^{(2)}$ can (if $\alpha^{(1)}_{p,i,\mu}\neq 0$) be factorized into

$$\alpha^{(2)}_{p,i,\mu} = \left(\prod_{\nu=1}^{d}\alpha^{(1)}_{p,i,\nu}\right)/\alpha^{(1)}_{p,i,\mu}.$$

If one of the factors $\alpha^{(1)}_{p,i,q} = 0$, then $\alpha^{(2)}_{p,i,\mu} = 0$ for all $\mu \neq q$ and the computation of $\alpha^{(2)}_{p,i,q}$ is possible in $\mathcal{O}(Pkd)$.

3.3 Efficient Computation of the Gradient

In order to be able to compute the entries of the gradient of f at a position α efficiently, we have to consider the bilinear form again.

Lemma 11 *Let $p \in \{1, \dots, P\}$ and $\mu \in \{1, \dots, d\}$. For any index $I \in J^{p,\mu}$ there holds*

$$\left(Y_{i^*}^{(\mu^*,j^*)}\right)_I = \begin{cases} \left(\prod_{\substack{\nu=1 \\ \nu \neq \mu^*}}^{d} (x_{i^*,\nu})_{I_\nu^p}\right) (v_{j^*,\mu^*})_{I_{\mu^*}} & \text{if } \mu = \mu^*, \\ \left(\prod_{\substack{\nu=1 \\ \nu \neq \mu^*,\mu}}^{d} (x_{i^*,\nu})_{I_\nu^p}\right) (v_{j^*,\mu^*})_{I_{\mu^*}^p (x_{i^*,\mu})_{I_\mu}} & \text{otherwise.} \end{cases}$$

Proof: Follows directly from the Definition of $Y_{i^*}^{(\mu^*,j^*)}$. ∎

As a consequence of Lemma 11, we can precompute the values

$$\gamma(p, i^*, \mu) := \prod_{\nu \neq \mu} (x_{i^*,\nu})_{I_\nu^p} \tag{8}$$

for all p, i^*, μ and use these for all μ^* and all multi-indices I. The $P \cdot d \cdot k$ values each require a d-fold product to be computed. This can be done efficiently by consideration of the whole product

$$\bar{\gamma}(p, i^*) := \prod_{\nu=1}^{d} (X_{i^*,\nu})_{I_\nu^p}.$$

If more than one of the factors $(X_{i^*,\nu})_{I_\nu^p}$ is zero, then $\gamma \equiv 0$. If one of the factors is zero, then the corresponding non-zero entries of γ (at most Pk) can be computed in $\mathcal{O}(Pdk)$. If all factors are non-zero, then there holds

$$\gamma(p, i^*, \mu) = \bar{\gamma}(p, i^*)/(x_{i^*,\mu})_{I_\mu^p}.$$

Remark 12 *The values $\gamma(p, i^*, \mu)$ can be computed in $\mathcal{O}(Pdk)$ for all $p \in \{1, \dots, P\}, i^* \in \{1, \dots, k\}, \mu \in \{1, \dots, d\}$, cf. Algorithm 1.*

We define the entries of the defect $X - A$ on the sets of multiindices $J^{p,\mu}$ by

$$R_I^{(p,\mu)} := X_I - A_I, \qquad I \in J^{p,\mu}, p \in \{1, \dots, P\}, \mu \in \{1, \dots, d\}.$$

Lemma 13 (Computation of the defect) *Let $\mu \in \{1, \dots, d\}$ and $p \in \{1, \dots, P\}$ be fixed and let $I \in J^{p,\mu}$. Then*

$$R_I^{(p,\mu)} = \langle x, y \rangle - A_I,$$

where the two vectors $x, y \in \mathbb{R}^k$ are

$$x_i := (x_{i,\mu})_{I_\mu}, \qquad y_i := \prod_{\substack{\nu=1 \\ \nu \neq \mu}}^{d} (x_{i,\nu})_{I_\nu^p}.$$

The complexity for the computation of $R^{(p,\mu)}$ for all μ, p is $\mathcal{O}(Pk \sum_{\nu=1}^{d} n_\nu)$.

Algorithm 1 Computation of $\gamma(p,i,\mu)$ for $\mu = 1,\dots,d$

{Choose a tolerance $0 < \delta < 1$ (e.g. $\delta = 10^{-16}$)}
$\pi := \prod_{\nu=1}^{d} (x_{i,\nu})_{I_\nu^p}$; $\mu := 0$;
for $\nu = 1,\dots,d$ **do**
  if $|(x_{i,\nu})_{I_\nu^p}| \le \delta$ **then**
    $\mu := \nu$;
  end if
end for
if $\mu = 0$ **then**
  for $\nu = 1,\dots,d$ **do**
    $\gamma(p,i,\nu) := \pi/(x_{i,\nu})_{I_\nu^p}$
  end for
else
  for $\nu \in \{1,\dots,d\} \setminus \{\mu\}$ **do**
    $\gamma(p,i,\nu) := 0$
  end for
  $y := 1.0$
  for $\nu \in \{1,\dots,d\} \setminus \{\mu\}$ **do**
    $y := y \cdot (x_{i,\nu})_{I_\nu^p}$
  end for
  $\gamma(p,i,\mu) := y$
end if

Proof: *(Representation).* Let $\mu \in \{1,\dots,d\}$, $p \in \{1,\dots,P\}$ and $I \in J^{p,\mu}$. Then

$$X_I = \sum_{i=1}^{k} (X_i)_I = \sum_{i=1}^{k} (x_{i,\mu})_{I_\mu} \prod_{\nu\neq\mu} (x_{i,\nu})_{I_\nu^p} = \sum_{i=1}^{k} x_i y_i.$$

(Complexity). If the vectors $x, y \in \mathbb{R}^k$ are available for all $\mu, p, I \in J^{p,\mu}$, then the complexity to compute $R_I^{(p,\mu)}$ is that of a scalar product of length k. In total this is $\mathcal{O}(Pk\sum_{\nu=1}^{d} n_\nu)$. The entries of x are given explicitly. The entries of y are $\gamma(p,i,\mu)$. ∎

Algorithm 2 Computation of $R_I^{(p,\mu)}$

Require: $\gamma(p,i,\mu)$
$z := 0$
for $i = 1,\dots,k$ **do**
  $z := z + (x_{i,\mu})_{I_\mu} \gamma(p,i,\mu)$
end for
$R_I^{(p,\mu)} := z$

The scalar products of the defect with either a component of X or a vector $v_{j,\nu}$ from the orthonormal basis is denoted by

$$s(p,\mu,j) := \langle R^{(p,\mu)}, v_{j,\mu} \rangle, \qquad t(p,\mu,i) := \langle R^{(p,\mu)}, x_{i,\mu} \rangle,$$

and all these values can be computed in $\mathcal{O}(P\sum_{\nu=1}^{d} k_\nu n_\nu + Pk\sum_{\nu=1}^{d} n_\nu)$.

Lemma 14 (Computation of the gradient) *The $k\sum_{\mu=1}^{d} k_\mu$ entries $\partial_{\alpha_{i^*,j^*,\mu^*)}} f(\alpha)$ of the gradient of f at position α can be computed with a storage (N_{St}) and work (N_{co}) complexity of*

$$N_{St}(\nabla) = \mathcal{O}\left(P\sum_{\mu=1}^{d}(k+k_\mu+n_\mu)\right), \qquad N_{co}(\nabla) = \mathcal{O}\left(P\sum_{\mu=1}^{d}(kk_\mu+kn_\mu+k_\mu n_\mu)\right).$$

Proof: 1.(Setup) For the setup we compute and store the entries of $\gamma(p,i^*,\mu)$ according to Remark 12 in $\mathcal{O}(Pkd)$. The defects $R^{(p,\mu)}$ are computable in $\mathcal{O}(Pk\sum_{\nu=1}^{d} n_\nu)$ (Lemma 13) and require $\mathcal{O}(P\sum_{\nu=1}^{d} n_\nu)$ units of storage. The scalar products $s(p,\mu,j), t(p,\mu,i)$ require $\mathcal{O}(P\sum_{\nu=1}^{d} k_\nu + Pdk)$ units of storage and their assembly takes $\mathcal{O}(P\sum_{\nu=1}^{d}(k+k_\nu)n_\nu)$. In total the storage requirements $N_{St,1}$ and complexity $N_{co,1}$ of the setup phase are

$$N_{St,1} = \mathcal{O}\left(P\sum_{\mu=1}^{d}(k+k_\mu+n_\mu)\right), \qquad N_{co,1} = \mathcal{O}\left(P\sum_{\mu=1}^{d}(k+k_\mu)n_\mu)\right).$$

2.(Computation) We have to compute entries of the form $\langle R^{(p,\mu)}, Y_{i^*}^{(\mu^*,j^*)}\rangle_{p,\mu}$.
2.1.($\mu=\mu^*$) In the case $\mu=\mu^*$ there holds

$$\langle R^{(p,\mu)}, Y_{i^*}^{(\mu^*,j^*)}\rangle_{p,\mu} = \gamma(p,i^*,\mu)s(p,j^*,\mu^*),$$

i.e., these values are available in $\mathcal{O}(1)$ each, in total

$$N_{St,2} = 0, \qquad N_{co,2} = \mathcal{O}\left(Pk\sum_{\nu=1}^{d} k_\nu\right).$$

2.2.($\mu\neq\mu^*$) From now on we consider the cases $\mu^*\neq\mu$. Then the entries are of the form

$$\langle R^{(p,\mu)}, Y_{i^*}^{(\mu^*,j^*)}\rangle_{p,\mu} = \left(\prod_{\substack{\nu=1\\ \nu\neq\mu^*,\mu}}^{d} (x_{i^*,\nu})_{I_\nu^p}\right)(v_{j^*,\mu^*})_{I_{\mu^*}^p} t(p,\mu,i^*). \tag{9}$$

In the following we will fix $p\in\{1,\dots,P\}$ and $i^*\in\{1,\dots,k\}$. We distinguish three cases:

1. first there are $q\neq a\in\{1,\dots,d\} : (x_{i^*,a})_{I_a^p} = 0 = (x_{i^*,q})_{I_q^p}$;
2. second that there exists exactly one $q\in\{1,\dots,d\}$ such that $(x_{i^*,q})_{I_q^p}=0$ and
3. third that $(x_{i^*,\nu})_{I_\nu^p}\neq 0$ for all $\nu=1,\dots,d$.

2.2.1. *(Case $(x_{i^*,q})_{I_q^p} = 0 = (x_{i^*,a})_{I_a^p}$).* All entries except for $\mu^*=q,\mu=a$ or $\mu^*=a,\mu=q$ are zero. The two non-zero entries can be computed by forming the product $\prod_{\nu\neq q,a}(x_{i^*,\nu})_{I_\nu^p}$ in $\mathcal{O}(d)$ for each i^*,p, in total

$$N_{St,3} = \mathcal{O}(Pkd), \qquad N_{co,3} = \mathcal{O}\left(Pk\sum_{\nu=1}^{d} k_\nu\right).$$

2.2.2. *(Case $(x_q^{(i^*)})_{I_q^p} = 0$).* In this case we have to consider only the combinations $\mu^* = q$ or $\mu = q$, otherwise the product in (9) is zero. The sought non-zero products are

$$\gamma(p, i^*, q)/(x_{i^*,\mu})_{I_\mu^p} \quad \text{for } \mu^* = q, \qquad \gamma(p, i^*, q)/(x_{i^*,\mu^*})_{I_{\mu^*}^p} \quad \text{for } \mu = q$$

and yield the complexity of

$$N_{St,3} = 0, \qquad N_{co,3} = \mathcal{O}\left(Pk \sum_{\nu=1}^{d} k_\nu\right).$$

2.2.3. *(Case $(x_{i^*,\nu})_{I_\nu^p} \neq 0$ for all $\nu = 1, \dots, d$).* Now things are a bit trickier, because there are d^2 combinations of μ^* and μ, but we want to reach a complexity that is only linear in d. We define

$$\tilde{t}(p, \mu, i) := t(p, \mu, i)/(x_{i,\mu})_{I_\mu^p}, \qquad \bar{t}(p, \mu^*, i) := \sum_{\mu=1}^{d} \tilde{t}(p, \mu, i) - \tilde{t}(p, \mu^*, i).$$

Both require Pdk units of storage and work. Finally, we obtain in $\mathcal{O}(Pk \sum_{\nu=1}^{d} k_\nu)$

$$\sum_{\mu \neq \mu^*} \langle R^{(p,\mu)}, Y_{i^*}^{(\mu^*,j^*)} \rangle_{p,\mu} = \sum_{\mu \neq \mu^*} \gamma(p, i^*, \mu^*)(v_{j^*,\mu^*})_{I_{\mu^*}^p} \tilde{t}(p, \mu, i^*) = \gamma(p, i^*, \mu^*)(v_{j^*,\mu^*})_{I_{\mu^*}^p} \bar{t}(p, \mu^*, i^*).$$

All parts together require $N_{St}(\nabla) = \sum_{\iota=1}^{3} N_{St,\iota}$ storage and $N_{co}(\nabla) = \sum_{\iota=1}^{3} N_{co,\iota}$ work. ∎

Lemma 4 shows that the gradient can be computed in almost optimal complexity: the number of input data is $\sum_{\mu=1}^{d}(kk_\mu + k_\mu n_\mu)$ for X and $P \sum_{\mu=1}^{d} n_\mu$ for A, i.e., we require only an additional factor of either P or k. The complete procedure is summarized in Algorithm 3.

3.4 Efficient Computation of the Hessian

The Hessian is defined as

$$H_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} := \partial_{\alpha_{\ell^*,m^*,\nu^*}} \partial_{\alpha_{i^*,j^*,\mu^*}} f(\alpha),$$

and its structure is analyzed in the following. The whole matrix has $K^2 := (k \sum_{\mu=1}^{d} k_\mu)^2$ entries and each of the entries requires the computation of P scalar products in dimension d. If $k_\mu = k$ for all $\mu = 1, \dots, d$ and $P, n_\mu = \mathcal{O}(k)$ then a naive approach would require

$$N_{H,naive} = \mathcal{O}(d^3 k^6)$$

operations for the setup of H and $d^2 k^4$ units of storage. The nonic complexity makes this approach unattractive for high dimensions and high ranks. Our aim is to reduce the setup time and to provide a form of H that allows for a fast matrix-vector multiplication in $\mathcal{O}(Pk^2 \sum_{\mu=1}^{d} k_\mu)$, i.e., only quintic complexity. In addition, we derive a suitable preconditioner for the iterative solution of the Hessian.

The matrix H is of the form $H = 2(D + C)$, where

$$D_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} = \sum_{p=1}^{P} \sum_{\mu=1}^{d} \langle Y_{\ell^*}^{(\nu^*,m^*)}, \; Y_{i^*}^{(\mu^*,j^*)} \rangle_{p,\mu} \tag{10}$$

Algorithm 3 Computation of $\mathrm{grad}(i^*, j^*, \mu^*) := \partial_{\alpha_{i^*,j^*,\mu^*}} f(\alpha)$ for all i^*, j^*, μ^*

Require: $\gamma(p, i, \mu), R_I^{(p,\mu)}, s(p, j, \mu), t(p, \mu, i)$
{Choose a tolerance $0 < \delta < 1$ (e.g. $\delta = 10^{-16}$)}
Initialise $\mathrm{grad}(i^*, j^*, \mu^*) := 0$
for $p = 1, \ldots, P$ **do**
  for $i^* = 1, \ldots, k$ **do**
    Determine the two smallest factors $\theta_1 := |(X_{i^*,q})_{I_q^p}| \leq \theta_2 := |(X_{i^*,a})_{I_a^p}|$
    for $\mu^* = 1, \ldots, d$ **do**
      for $j^* = 1, \ldots, k_{\mu^*}$ **do**
        $\mathrm{grad}(i^*, j^*, \mu^*) := \mathrm{grad}(i^*, j^*, \mu^*) + 2\gamma(p, i^*, \mu^*)s(p, j^*, \mu^*)$
      end for
    end for
    if $\theta_2 \leq \delta$ **then**
      $y := \prod_{\mu \neq q,a} (X_{i^*,\mu})_{I_\mu^p}$
      for $j^* = 1, \ldots, k_a$ **do**
        $\mathrm{grad}(i^*, j^*, a) := \mathrm{grad}(i^*, j^*, a) + 2y(v_{j^*,a})_{I_a^p} t(p, q, i^*)$
      end for
      for $j^* = 1, \ldots, k_q$ **do**
        $\mathrm{grad}(i^*, j^*, q) := \mathrm{grad}(i^*, j^*, q) + 2y(v_{j^*,q})_{I_q^p} t(p, a, i^*)$
      end for
    else
      if $\theta_1 \leq \delta$ **then**
        for $\mu^* \in \{1, \ldots, d\} \setminus \{q\}$ **do**
          for $j^* = 1, \ldots, k_{\mu^*}$ **do**
            $\mathrm{grad}(i^*, j^*, \mu^*) := \mathrm{grad}(i^*, j^*, \mu^*) + 2\gamma(p, i^*, q)(v^{(j^*)}_{j^*,\mu^*})_{I_{j^*}^p} t(p, q, i^*)/(X_{i^*,\mu^*})_{I_{\mu^*}^p}$
          end for
        end for
        for $\mu \in \{1, \ldots, d\} \setminus \{q\}$ **do**
          for $j^* = 1, \ldots, k_q$ **do**
            $\mathrm{grad}(i^*, j^*, q) := \mathrm{grad}(i^*, j^*, q) + 2\gamma(p, i^*, q)(w_{j^*,q})_{I_q^p} t(p, \mu, i^*)/(X_{i^*,q})_{I_q^p}$
          end for
        end for
      else
Require: $\bar{t}(p, \mu, i^*)$ for $\mu = 1, \ldots, d$
        for $\mu^* = 1, \ldots, d$ **do**
          for $j^* = 1, \ldots, k_{\mu^*}$ **do**
            $\mathrm{grad}(i^*, j^*, \mu^*) := \mathrm{grad}(i^*, j^*, \mu^*) + 2\gamma(p, i^*, \mu^*)(w_{j^*,\mu^*})_{I_{\mu^*}^p} \bar{t}(p, \mu^*, i^*)$
          end for
        end for
      end if
    end if
  end for
end for

Definition 15 (Splitting of D) *Let $K := k\sum_{\mu=1}^{d} k_\mu$. We introduce the products*

$$\zeta(p,i,\mu_1,\mu_2) := \prod_{\substack{\nu=1\\ \nu\neq\mu_1,\mu_2}}^{d} (x_{i,\nu})_{I_\nu^p}$$

and define the matrices $D^1, D^2, D^3 \in \mathbb{R}^{K\times K}$ by

$$D^1_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} := \sum_{p=1}^{P} \delta_{j^*,m^*}\delta_{\mu^*,\nu^*}\gamma(p,i^*,\mu^*)\gamma(p,\ell^*,\nu^*), \tag{11}$$

$$\begin{aligned} D^2_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} := \sum_{p=1}^{P} \bar{\delta}_{\mu^*,\nu^*}\Big(&\gamma(p,i^*,\mu^*)\zeta(p,\ell^*,\mu^*,\nu^*)(v_{m^*,\nu^*})_{I_{\nu^*}^p}\alpha_{\ell^*,j^*,\mu^*} \\ &+\zeta(p,i^*,\mu^*,\nu^*)\gamma(p,\ell^*,\nu^*)(v_{j^*,\mu^*})_{I_{\mu^*}^p}\alpha_{i^*,m^*,\nu^*}\Big), \end{aligned} \tag{12}$$

$$\begin{aligned} D^3_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} := \sum_{p=1}^{P}\sum_{\substack{\mu=1\\ \mu\neq\mu^*,\nu^*}}^{d} &\zeta(p,i^*,\mu^*,\mu)\zeta(p,\ell^*,\nu^*,\mu)\times \\ &\times(v_{j^*,\mu^*})_{I_{\mu^*}^p}(v_{m^*,\nu^*})_{I_{\nu^*}^p}\langle x_{i^*,\mu}, x_{\ell^*,\mu}\rangle. \end{aligned} \tag{13}$$

Lemma 16 *The matrices D^1, D^2, D^3 sum up to D:*

$$D = D^1 + D^2 + D^3$$

Proof: We split the summation over μ into the three parts

- (D^1) $\quad \mu = \mu^* = \nu^*$
- (D^2) $\quad \mu = \mu^* \neq \nu^*$ or $\mu = \nu^* \neq \mu^*$
- (D^3) $\quad \mu \neq \mu^*$ and $\mu \neq \nu^*$.

According to the Definition of Y and Lemma 11 we obtain the stated form. ∎

If we use the lexicographical ordering with respect to (μ^*, i^*, j^*) and (ν^*, ℓ^*, m^*), then D^1 is a block-diagonal matrix with d blocks on the diagonal. The first blocking with respect to μ^*, ν^* gives a block-diagonal matrix with matrices of the following form

$$\hat{D}^{1,\mu} = \left(\sum_{p=1}^{P} \gamma_{p,\mu}(\gamma_{p,\mu})^T\right) \otimes \mathbf{Id}_{k_\mu\times k_\mu}, \tag{14}$$

where the entries of $\gamma_{p,\mu} \in \mathbb{R}^k$ are

$$(\gamma_{p,\mu})_i := \gamma(p,i,\mu) = \prod_{\nu\neq\mu}(x_{i,\nu})_{I_\nu^p}, \quad i \in \{1,\dots,k\},$$

see equation (8). Moreover we have

$$(\gamma_{p,\mu})_i = \prod_{\nu\neq\mu}\langle x_{i,\nu}, e_{I_\nu^p}\rangle_{\mathbb{R}^{k_\mu}} = \left\langle \bigotimes_{\nu\neq\mu} x_{i,\nu}, \bigotimes_{\nu\neq\mu} e_{I_\nu^p}\right\rangle,$$

where $e_{I_\nu^p} \in \mathbb{R}^{k_\mu}$ is a canonical unit vector. Using standard calculations we can show that

$$\begin{aligned}
\hat{D}^{1,\mu} &= \left(\sum_{p=1}^{P} \left((X^\mu)^t E_P^\mu \right) \left((X^\mu)^t E_P^\mu \right)^t \right) \otimes \mathbf{Id}_{k_\mu \times k_\mu} \\
&= \left((X^\mu)^t \sum_{p=1}^{P} E_P^\mu \left(E_P^\mu \right)^t X^\mu \right) \otimes \mathbf{Id}_{k_\mu \times k_\mu} \\
&= (X^\mu)^t P^\mu X^\mu \otimes \mathbf{Id}_{k_\mu \times k_\mu} = G^\mu \otimes \mathbf{Id}_{k_\mu \times k_\mu},
\end{aligned}$$

where X^μ, E_P^μ and P^μ are defined as follows:

$$\begin{aligned}
X^\mu &:= \left(\bigotimes_{\nu \neq \mu} x_{1\mu}, \ldots, \bigotimes_{\nu \neq \mu} x_{k\mu} \right) \in \mathbb{R}^{K_\mu \times k}, \quad K_\mu := \prod_{\nu \neq \mu} k_\mu, \\
E_P^\mu &:= \bigotimes_{\nu \neq \mu} e_{I_\nu^p} \in \mathbb{R}^{K_\mu}, \\
\mathrm{P}^\mu &:= \sum_{p=1}^{P} \bigotimes_{\nu \neq \mu} e_{I_\nu^p} (e_{I_\nu^p})^t \in \mathbb{R}^{K_\mu \times K_\mu}, \quad G^\mu := (X^\mu)^t P^\mu X^\mu \in \mathbb{R}^{k \times k}.
\end{aligned}$$

Remark 17 *Without loss of generality, we can assume that the rank of the matrix P^μ is equal to P, since otherwise we have constructed at least two fibres in direction μ exactly at the same position.*

Lemma 18 *Let $X \in \mathcal{T}(d,k)$ with tensor rank exactly k. Then we have for all $\mu \in \{1,\ldots,d\}$ that the matrix X^μ has rank k.*

Proof: Assume there exists $\mu \in \{1,\ldots,d\}$ with $\left\{ \bigotimes_{\nu\neq\mu} x_{1\mu}, \ldots, \bigotimes_{\nu\neq\mu} x_{k\mu} \right\}$ linear dependent. Then there are $\lambda_1,\ldots,\lambda_k \in \mathbb{R}$ and $i_0 \in \{1,\ldots,k\}$ with $\sum_{i=0}^{k} \lambda_i \bigotimes_{\nu\neq\mu} x_{i\mu} = 0$ and $\lambda_{i_0} \neq 0$. Without lost of generality, let $\mu = 1$ and $i_0 = k$. Then we have

$$\bigotimes_{\nu=2}^{d} x_{k\nu} = \sum_{i=1}^{k-1} \underbrace{\frac{-\lambda_i}{\lambda_k}}_{\tilde{\lambda}_i :=} \bigotimes_{\nu=2}^{d} x_{i\nu}$$

and

$$\begin{aligned}
X &= \sum_{i=1}^{k-1} \bigotimes_{\nu=1}^{d} x_{i\nu} + \bigotimes_{\nu=1}^{d} x_{k\nu} = \sum_{i=1}^{k-1} x_{i1} \otimes \bigotimes_{\nu=2}^{d} x_{i\nu} + \tilde{\lambda}_i x_{r1} \otimes \bigotimes_{\nu=2}^{d} x_{i\nu} \\
&= \sum_{i=1}^{k-1} \underbrace{(x_{i1} + \lambda_i x_{r1}) \otimes \bigotimes_{\nu=2}^{d} x_{i\nu}}_{\tilde{X}_i :=} = \sum_{i=1}^{k-1} \tilde{X}_i.
\end{aligned}$$

This contradicts the fact that $\mathrm{rank}(X) = k$. ∎

Corollary 19 *Let $X \in \mathcal{T}(d,k)$ with tensor rank exactly k and the pivot points constructed as mentioned in Remark 17. Then D^1 is positive definite hence regular. Furthermore, we have*

$$(D^1)^{-1} = \sum_{\mu=1}^{d} \mathbb{E}^\mu \otimes (G^\mu)^{-1} \otimes \mathbf{Id}_{k_\mu \times k_\mu}, \text{ where } \quad \mathbb{E}^\mu \in \mathbb{R}^{d\times d}, \quad (\mathbb{E}^\mu)_{\mu_1\mu_2} := \delta_{\mu\mu_1}\delta_{\mu\mu_2}.$$

Proof: Since $\hat{D}^{1,\mu} = G^\mu \otimes \mathbf{Id}_{k_\mu\times k_\mu}$ is a sum of positive semidefinite matrices, see equation (14), it is positive semidefinite. It follows from Lemma 18 and Remark 17 that G^μ is regular and therefore $\hat{D}^{1,\mu}$ is positive definite. Since D^1 is a block-diagonal matrix of positive definite matrices, it is positive definite. Moreover, we have

$$\begin{aligned} D^1(D^1)^{-1} &= \sum_{\mu=1}^{d}\sum_{\mu'=1}^{d} \mathbb{E}^\mu\mathbb{E}^{\mu'} \otimes G^\mu(G^{\mu'})^{-1} \otimes \mathbf{Id}_{k_\mu\times k_\mu} \\ &= \sum_{\mu=1}^{d} \mathbb{E}^\mu \otimes \mathbf{Id}_{k\times k} \otimes \mathbf{Id}_{k_\mu\times k_\mu} = \mathbf{Id}. \end{aligned}$$

■

Remark 20 (Preconditioner D^1) *The storage requirements for the matrix D^1 are $\mathcal{O}(dk^2)$. The matrices $G^\mu \in \mathbb{R}^{k\times k}$ can be factorized in $\mathcal{O}(k^3)$ (in total $\mathcal{O}(dk^3)$ for D^1), so that a subsequent matrix-vector multiplication with the inverse of D^1 takes $\mathcal{O}(k^2\sum_{\mu=1}^d k_\mu)$.*

As mentioned above a Newton-type iteration is used to solve the minimization problem. Therefore, we have to solve a linear system in every iteration step. The matrix D^1 is a very good preconditioner for this problem, see Section 4.

The treatment of the two dense matrices D^2 and D^3 is considerably more involved than that of D^1 and will be presented in the following two Lemmata.

Lemma 21 (Computation and Evaluation of D^2) *The matrix D^2 can be stored in data-sparse form requiring $\mathcal{O}(Pdk)$ units of memory, $\mathcal{O}(Pdk)$ basic arithmetic operations for the setup and $\mathcal{O}(Pk^2\sum_{\mu=1}^d k_\mu)$ for a subsequent matrix-vector multiplication.*

Proof: We consider blocks of D^2 corresponding to the indices $i^*, \ell^* = 1,\dots,k$ for fixed $p \in \{1,\dots,P\}$ and we treat only the first term (first line in (12))

$$D^{2,p,i^*,\ell^*}_{(j^*,\mu^*),(m^*,\nu^*)} := \bar{\delta}_{\mu^*,\nu^*}\gamma(p,i^*,\mu^*)\zeta(p,\ell^*,\mu^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\alpha_{\ell^*,j^*,\mu^*},$$

the second term can be treated analogously. Three cases can occur for a fixed ℓ^*: The number of indices μ for which $(x_{\ell^*,\mu})_{I^p_\mu} = 0$ can be zero, one or ≥ 2.

Case 1, none of the $(x_{\ell^*,\mu})_{I^p_\mu}$, $\mu = 1,\dots,d$, is zero. In this case we can write

$$\zeta(p,\ell^*,\mu^*,\nu^*) = \prod_{\substack{\mu=1\\ \mu\neq\mu^*,\nu^*}}^{d} (x_{\ell^*,\mu})_{I^p_\mu} = \gamma(p,\ell^*,\nu^*)/(x_{\ell^*,\mu^*})_{I^p_{\mu^*}}$$

and observe that inserting this into the definition of D^{2,p,i^*,ℓ^*} yields

$$\begin{aligned} D^{2,p,i^*,\ell^*}_{(j^*,\mu^*),(m^*,\nu^*)} &= \bar{\delta}_{\mu^*,\nu^*}\gamma(p,i^*,\mu^*)\gamma(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\alpha_{\ell^*,j^*,\mu^*}/(x_{\ell^*,\mu^*})_{I^p_{\mu^*}} \\ &= \left(\gamma(p,i^*,\mu^*)\alpha_{\ell^*,j^*,\mu^*}/(x_{\ell^*,\mu^*})_{I^p_{\mu^*}}\right)\left(\gamma(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right) \\ &\quad -\delta_{\mu^*,\nu^*}\left(\gamma(p,i^*,\mu^*)\alpha_{\ell^*,j^*,\mu^*}/(x_{\ell^*,\mu^*})_{I^p_{\mu^*}}\right)\left(\gamma(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right). \end{aligned}$$

The first product separates the variables j^*,μ^* and m^*,ν^*, i.e., we obtain a rank one matrix. The second product yields a block-diagonal matrix with rank one matrices on the block-diagonal. Both allow (for each p,i^*,ℓ^*) in the above representation a matrix-vector multiplication at a cost of $\mathcal{O}(\sum_{\mu=1}^d k_\mu)$ which gives in total $\mathcal{O}(Pk^2\sum_{\mu=1}^d k_\mu)$.

Case 2, there is exactly one $(x_{\ell^*,q})_{I^p_q}=0$. Then the term $\zeta(p,\ell^*,\mu^*,\nu^*)$ is non-zero only for $\mu^*=q$ or $\nu^*=q$. Consequently

$$\begin{aligned} D^{2,p,i^*,\ell^*}_{(j^*,\mu^*),(m^*,\nu^*)} &= \delta_{\mu^*,q}\bar{\delta}_{q,\nu^*}\gamma(p,i^*,\mu^*)\gamma(p,\ell^*,\mu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\alpha_{\ell^*,j^*,\mu^*}/(x_{\ell^*,\nu^*})_{I^p_{\nu^*}} \\ &\quad +\delta_{\nu^*,q}\bar{\delta}_{q,\mu^*}\gamma(p,i^*,\mu^*)\gamma(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\alpha_{\ell^*,j^*,\mu^*}/(x_{\ell^*,\mu^*})_{I^p_{\mu^*}} \\ &= (\delta_{\mu^*,q}\gamma(p,i^*,\mu^*)\gamma(p,\ell^*,\mu^*)\alpha_{\ell^*,j^*,\mu^*})\left(\bar{\delta}_{q,\nu^*}(v_{m^*,\nu^*})_{I^p_{\nu^*}}/(x_{\ell^*,\nu^*})_{I^p_{\nu^*}}\right) \\ &\quad +\left(\bar{\delta}_{q,\mu^*}\gamma(p,i^*,\mu^*)\alpha_{\ell^*,j^*,\mu^*}/(x_{\ell^*,\mu^*})_{I^p_{\mu^*}}\right)\left(\delta_{\nu^*,q}\gamma(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right) \end{aligned}$$

is a rank 2 matrix for each (p,i^*,ℓ^*) and allows a matrix-vector multiplication in $\mathcal{O}(Pk^2\sum_{\mu=1}^d k_\mu)$.

Case 3, there exist $q\neq a$ such that $(x_{\ell^*,\iota})_{I^p_\iota}=0$ for $\iota\in\{q,a\}$. This yields non-zero entries only for the case $\mu^*,\nu^*\in\{a,q\}$. For these $\mathcal{O}(1)$ combinations the sub-matrix is of rank 1 because the variables j^* and m^* factorize. ∎

Lemma 22 (Computation and Evaluation of D^3) *The matrix D^3 can be stored in data-sparse form requiring $\mathcal{O}(Pdk)$ units of memory, $\mathcal{O}(Pdk)$ basic arithmetic operations for the setup and $\mathcal{O}(Pk^2\sum_{\mu=1}^d k_\mu)$ for a subsequent matrix-vector multiplication.*

Proof: We recall that D^3 is defined as

$$D^3_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} = \sum_{p=1}^P \sum_{\substack{\mu=1\\ \mu\neq\mu^*,\nu^*}}^d \zeta(p,i^*,\mu^*,\mu)\zeta(p,\ell^*,\nu^*,\mu)(v_{j^*,\mu^*})_{I^p_{\mu^*}}(v_{m^*,\nu^*})_{I^p_{\nu^*}}\langle x_{i^*,\mu},x_{\ell^*,\mu}\rangle.$$

Let $p\in\{1,\dots,P\}$ and $i^*,\ell^*\in\{1,\dots,k\}$ be fixed. We consider the submatrices

$$D^{3,p,i^*,\ell^*}_{(j^*,\mu^*),(m^*,\nu^*)} = \sum_{\substack{\mu=1\\ \mu\neq\mu^*,\nu^*}}^d \zeta(p,i^*,\mu^*,\mu)\zeta(p,\ell^*,\nu^*,\mu)(v_{j^*,\mu^*})_{I^p_{\mu^*}}(v_{m^*,\nu^*})_{I^p_{\nu^*}}\langle x_{i^*,\mu},x_{\ell^*,\mu}\rangle.$$

Two cases have to be distinguished:

- there exists $a\neq q\in\{1,\dots,d\}$ such that either $(x_{i^*,\mu})_{I^p_\mu}=0$ for $\mu=a,q$ or $(x_{\ell^*,\mu})_{I^p_\mu}=0$ for $\mu=a,q$;
- there exist $a,q\in\{1,\dots,d\}$ such that $(x_{i^*,\mu})_{I^p_\mu}\neq 0$ for $\mu\neq q$ and $(x_{\ell^*,\mu})_{I^p_\mu}\neq 0$ for $\mu\neq a$;

Case 1, there exists $a \neq q \in \{1, \dots, d\}$ such that $(X_{i^*,\mu})_{I^p_\mu} = 0$ for $\mu = a, q$ (analogously for ℓ^* instead of i^*). Then $\zeta(p, i^*, \mu^*, \mu) \neq 0$ only if $\mu^*, \mu \in \{a, q\}$. Now we can write

$$\begin{aligned} D^{3,p,i^*,\ell^*}_{(j^*,\mu^*),(m^*,\nu^*)} &= \delta_{\mu^*,q}\bar{\delta}_{\nu^*,a}\zeta(p, i^*, \mu^*, a)\zeta(p, \ell^*, \nu^*, a)(v_{j^*,\mu^*})_{I^p_{\mu^*}}(v_{m^*,\nu^*})_{I^p_{\nu^*}}\langle x_{i^*,a}, x_{\ell^*,a}\rangle \\ &\quad +\delta_{\mu^*,a}\bar{\delta}_{\nu^*,q}\zeta(p, i^*, \mu^*, q)\zeta(p, \ell^*, \nu^*, q)(v_{j^*,\mu^*})_{I^p_{\mu^*}}(v_{m^*,\nu^*})_{I^p_{\nu^*}}\langle x_{i^*,q}, x_{\ell^*,q}\rangle \\ &= \left(\delta_{\mu^*,q}\zeta(p, i^*, \mu^*, a)(v_{j^*,\mu^*})_{I^p_{\mu^*}}\langle x_{i^*,a}, x_{\ell^*,a}\rangle\right)\left(\bar{\delta}_{\nu^*,a}\zeta(p, \ell^*, \nu^*, a)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right) \\ &\quad + \left(\delta_{\mu^*,a}\zeta(p, i^*, \mu^*, q)(v_{j^*,\mu^*})_{I^p_{\mu^*}}\langle x_{i^*,q}, x_{\ell^*,q}\rangle\right)\left(\bar{\delta}_{\nu^*,q}\zeta(p, \ell^*, \nu^*, q)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right) \end{aligned}$$

as a rank 2 matrix (j^*, μ^* and m^*, ν^* are separated).

Case 2, there exist $a, q \in \{1, \dots, d\}$ such that $(x_{i^*\mu})_{I^p_\mu} \neq 0$ for $\mu \neq q$ and $(x_{\ell^*,\mu})_{I^p_\mu} \neq 0$ for $\mu \neq a$. We can split

$$\zeta(p, i^*, \mu^*, \mu) = \zeta_1(p, i^*, \mu^*)\zeta_2(p, i^*, \mu), \qquad \zeta(p, \ell^*, \nu^*, \mu) = \zeta_3(p, \ell^*, \nu^*)\zeta_4(p, \ell^*, \mu),$$

because at most one of the factors $(x_{i^*,\mu})_{I^p_\mu}$ (respectively $(x_{\ell^*,\mu})_{I^p_\mu}$) is zero for $\mu = 1, \dots, d$. $\zeta_1, \dots, \zeta_4$ can be obtained from γ and stored in $\mathcal{O}(Pkd)$. The matrix D^{3,p,i^*,ℓ^*} has the entries

$$\begin{aligned} D^{3,p,i^*,\ell^*}_{(j^*,\mu^*),(m^*,\nu^*)} &= \sum_{\substack{\mu=1 \\ \mu\neq\mu^*,\nu^*}}^{d} \zeta(p, i^*, \mu^*, \mu)\zeta(p, \ell^*, \nu^*, \mu)(v_{j^*,\mu^*})_{I^p_{\mu^*}}(v_{m^*,\nu^*})_{I^p_{\nu^*}}\langle x_{i^*,\mu}, x_{\ell^*,\mu}\rangle. \\ &= \left(\zeta_1(p, i^*, \mu^*)(v_{j^*,\mu^*})_{I^p_{\mu^*}}\right)\left(\zeta_3(p, \ell^*, \nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right) \times \\ &\quad \times \sum_{\substack{\mu=1 \\ \mu\neq\mu^*,\nu^*}}^{d} \zeta_2(p, i^*, \mu)\zeta_4(p, \ell^*, \mu)\langle x_{i^*,\mu}, x_{\ell^*,\mu}\rangle. \end{aligned}$$

The last summation is split into parts where μ^*, μ, ν^* are separated:

$$\begin{aligned} \sum_{\substack{\mu=1 \\ \mu\neq\mu^*,\nu^*}}^{d} \zeta_2(p, i^*, \mu)\zeta_4(p, \ell^*, \mu)\langle x_{i^*,\mu}, x_{\ell^*,\mu}\rangle &= \sum_{\mu=1}^{d} \zeta_2(p, i^*, \mu)\zeta_4(p, \ell^*, \mu)\langle x_{i^*,\mu}, x_{\ell^*,\mu}\rangle \\ &\quad -\zeta_2(p, i^*, \mu^*)\zeta_4(p, \ell^*, \mu^*)\langle x_{i^*,\mu^*}, x_{\ell^*,\mu^*}\rangle \\ &\quad -\zeta_2(p, i^*, \nu^*)\zeta_4(p, \ell^*, \nu^*)\langle x_{i^*,\nu^*}, x_{\ell^*,\nu^*}\rangle \\ &\quad +\delta_{\mu^*,\nu^*}\zeta_2(p, i^*, \mu^*)\zeta_4(p, \ell^*, \mu^*)\langle x_{i^*,\mu^*}, x_{\ell^*,\mu^*}\rangle \end{aligned}$$

The first term

$$\zeta_{24}(p, i^*, \ell^*) := \sum_{\mu=1}^{d} \zeta_2(p, i^*, \mu)\zeta_4(p, \ell^*, \mu)\langle x_{i^*,\mu}, x_{\ell^*,\mu}\rangle$$

can be computed in $\mathcal{O}(Pk^2d)$ and stored in $\mathcal{O}(Pk^2)$ units of memory. We get the represen-

tation

$$\begin{aligned}
D^{3,p,i^*,\ell^*}_{(j^*,\mu^*),(m^*,\nu^*)} &= \left(\zeta_1(p,i^*,\mu^*)(v_{j^*,\mu^*})_{I^p_{\mu^*}}\right) \times \\
&\times (\zeta_{24}(p,i^*,\ell^*) - \zeta_2(p,i^*,\mu^*)\zeta_4(p,\ell^*,\mu^*)\langle x_{i^*,\mu^*}, x_{\ell^*,\mu^*}\rangle) \times \\
&\times \left(\zeta_3(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right) \\
&- \left(\zeta_1(p,i^*,\mu^*)(v_{j^*,\mu^*})_{I^p_{\mu^*}}\right) \times \\
&\times (\zeta_2(p,i^*,\nu^*)\zeta_4(p,\ell^*,\nu^*)\langle x_{i^*,\nu^*}, x_{\ell^*,\nu^*}\rangle) \times \\
&\times \left(\zeta_3(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right) \\
&+ \delta_{\mu^*,\nu^*}\left(\zeta_1(p,i^*,\mu^*)(v_{j^*,\mu^*})_{I^p_{\mu^*}}\right) \times \\
&\times (\zeta_2(p,i^*,\mu^*)\zeta_4(p,\ell^*,\mu^*)\langle x_{i^*,\mu^*}, x_{\ell^*,\mu^*}\rangle) \times \\
&\times \left(\zeta_3(p,\ell^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}\right)
\end{aligned}$$

The first two terms lead to a rank two matrix and the third term leads to a block-diagonal matrix with blocks of rank one. The matrix-vector multiplication for a single matrix D^{3,p,i^*,ℓ^*} is of complexity $\mathcal{O}(\sum_{\mu=1}^d k_\mu)$, and altogether this yields the desired estimate $\mathcal{O}(Pk^2\sum_{\mu=1}^d k_\mu)$. ■

So far we have observed that the matrix D, the first part of the Hessian, can be stored and evaluated efficiently. At last we consider the second part of the Hessian, the matrix C with entries

$$C_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} = \sum_{p=1}^{P}\sum_{\mu=1}^{d} \delta_{i^*,\ell^*}\bar{\delta}_{\mu^*,\nu^*}\langle X - A, \quad Z^{(\mu^*,j^*,\nu^*,m^*)}_{i^*}\rangle_{p,\mu}. \tag{15}$$

According to the definition of Z we obtain

$$\langle X - A, \quad Z^{(\mu^*,j^*,\nu^*,m^*)}_{i^*}\rangle_{p,\mu} = \begin{cases} \prod\limits_{\substack{\nu=1\\ \nu\neq\mu^*,\nu^*}}^{d} (x_{i^*,\nu})_{I^p_\nu}(v_{m^*,\nu^*})_{I^p_{\nu^*}} s(p,\mu,j^*) & \mu = \mu^* \\ \prod\limits_{\substack{\nu=1\\ \nu\neq\mu^*,\nu^*}}^{d} (x_{i^*,\nu})_{I^p_\nu}(v_{j^*,\mu^*})_{I^p_{\mu^*}} s(p,\mu,m^*) & \mu = \nu^* \\ \prod\limits_{\substack{\nu=1\\ \nu\neq\mu,\mu^*,\nu^*}}^{d} (x_{i^*,\nu})_{I^p_\nu}(v_{m^*,\nu^*})_{I^p_{\nu^*}}(v_{j^*,\mu^*})_{I^p_{\mu^*}} t(p,\mu,i^*) & \text{otherwise.} \end{cases}$$

Lemma 23 (Computation and Evaluation of C) *The matrix C can be stored in data-sparse form requiring $\mathcal{O}(Pdk)$ units of memory, $\mathcal{O}(Pdk)$ basic arithmetic operations for the setup and $\mathcal{O}(Pk\sum_{\mu=1}^d k_\mu)$ for a subsequent matrix-vector multiplication.*

Proof: Let $p \in \{1,\dots,P\}$ and $i^* \in \{1,\dots,k\}$ be fixed. We distinguish three cases corresponding to the number of zero factors $X_{i^*,\mu}, \mu = 1,\dots,d$: up to one, exactly two or at least three.

For at least three zero factors $\mu = q, a, b$ there are at most 6 combinations $\mu^* \neq \nu^* \in \{q,a,b\}$ that lead to non-zero entries, and for each of these combinations the summation over μ consists of at most one non-zero term $\mu \in \{q,a,b\} \setminus \{\mu^*,\nu^*\}$. For each of the combinations the variables j^*, m^* are separated, i.e., the matrix block is of rank 1. In total this is a complexity of $\mathcal{O}(Pk\max_{\mu=1,\dots,d} k_\mu)$ for the matrix-vector product.

For exactly two zero factors $j = q, a$ there are $\mathcal{O}(d)$ combinations of μ, μ^*, ν^* that lead to non-zero entries in C:

1. $\mu^* \neq \nu^* \wedge \mu^*, \nu^* \in \{q, a\}$ and full summation over all μ or
2. $\#(\{\mu^*, \nu^*\} \cap \{q, a\}) = 1$ where the summation collapses to a single term.

Each of the blocks corresponding to one of the combinations is of rank one, in total the complexity is $\mathcal{O}(Pk \sum_{\mu=1}^{d} k_\mu)$ for the matrix-vector product.

Now let there be at most one zero-entry $x_{i^*,q}$. Then we can split the products as above:

$$\prod_{\substack{\nu=1 \\ \nu \neq \mu, \mu^*, \nu^*}}^{d} (x_{i^*,\nu})_{I_\nu^p} = \zeta_1(p, i^*, \mu)\zeta_2(p, i^*, \mu^*)\zeta_3(p, i^*, \nu^*).$$

We split the summation (for fixed $i^* = \ell^*, p = 1, \ldots, P$) into three parts:

$$\begin{aligned}
C^{1,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} &:= \bar{\delta}_{\mu^*,\nu^*} \langle X - A, \; Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} \rangle_{p,\mu^*}, \\
C^{2,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} &:= \bar{\delta}_{\mu^*,\nu^*} \langle X - A, \; Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} \rangle_{p,\nu^*}, \\
C^{3,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} &:= \sum_{\substack{\mu=1 \\ \nu \neq \mu^*, \nu^*}}^{d} \bar{\delta}_{\mu^*,\nu^*} \langle X - A, \; Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} \rangle_{p,\mu}.
\end{aligned}$$

For $C^{1,i^*,p}$ we obtain

$$\begin{aligned}
C^{1,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} &= \bar{\delta}_{\mu^*,\nu^*} \prod_{\substack{\nu=1 \\ \nu \neq \mu^*, \nu^*}}^{d} (X_{i^*,\nu})_{I_\nu^p} (v_{m^*,\nu^*})_{I_{\nu^*}^p} s(p, \mu, j^*) \\
&= \bar{\delta}_{\mu^*,\nu^*} \zeta_1(p, i^*, \mu^*)\zeta_2(p, i^*, \mu^*)\zeta_3(p, i^*, \nu^*)(v_{m^*,\nu^*})_{I_{\nu^*}^p} s(p, \mu^*, j^*) \\
&= (\zeta_1(p, i^*, \mu^*)\zeta_2(p, i^*, \mu^*)s(p, \mu^*, j^*)) \times \\
&\quad \times \left(\zeta_3(p, i^*, \nu^*)(v_{m^*,\nu^*})_{I_{\nu^*}^p}\right) \\
&\quad + \delta_{\mu^*,\nu*} (\zeta_1(p, i^*, \mu^*)\zeta_2(p, i^*, \mu^*)s(p, \mu^*, j^*)) \times \\
&\quad \times \left(\zeta_3(p, i^*, \mu^*)(v_{m^*,\mu^*})_{I_{\mu^*}^p}\right),
\end{aligned}$$

which is a rank one matrix plus a block-diagonal matrix with rank one diagonal blocks. Both allow for a matrix-vector product in $\mathcal{O}(\sum_{\nu=1}^{d} k_\mu)$, for all $i^*.p$ this sums up to $\mathcal{O}(Pk \sum_{\nu=1}^{d} k_\mu)$. In the same way we derive the complexity for $C^{2,i^*,p}$. At last we consider the matrices $C^{3,i^*,p}$:

$$\begin{aligned}
C^{3,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} &= \bar{\delta}_{\mu^*,\nu^*} \sum_{\substack{\mu=1 \\ \mu \neq \mu^*, \nu^*}}^{d} \zeta_1(p, i^*, \mu)\zeta_2(p, i^*, \mu^*)\zeta_3(p, i^*, \nu^*) \times \\
&\quad \times (v_{m^*,\nu^*})_{I_{\nu^*}^p} (v_{j^*,\mu^*})_{I_{\mu^*}^p} t(p, \mu, i^*).
\end{aligned}$$

For their efficient representation we define

$$\begin{aligned}\psi(p,i^*,\nu^*,\mu^*) &:= \sum_{\substack{\mu=1\\ \mu\neq\mu^*,\nu^*}}^{d} \zeta_1(p,i^*,\mu)t(p,\mu,i^*)\\ &= \sum_{\mu=1}^{d}\zeta_1(p,i^*,\mu)t(p,\mu,i^*) \quad +\delta_{\mu^*,\nu^*}\zeta_1(p,i^*,\mu^*)t(p,\mu^*,i^*)\\ &\quad -\zeta_1(p,i^*,\mu^*)t(p,\mu^*,i^*)-\zeta_1(p,i^*,\nu^*)t(p,\nu^*,i^*)\\ &= \psi_1(p,i^*)+\delta_{\mu^*,\nu^*}\psi_2(p,i^*,\mu^*)+\psi_3(p,i^*,\mu^*)+\psi_4(p,i^*,\nu^*).\end{aligned}$$

The three terms ψ_1,ψ_3,ψ_4 are separable with respect to μ^*,ν^* and the term $\delta_{\mu^*,\nu^*}\psi_2(p,i^*,\mu^*)$ leads to a diagonal matrix with respect to μ^*,ν^*. Inserting this representation into the definition of $C^{3,i^*,p}$ we obtain

$$\begin{aligned}C^{3,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} &= \bar{\delta}_{\mu^*,\nu^*}\psi(p,i^*,\nu^*,\mu^*)\zeta_2(p,i^*,\mu^*)\zeta_3(p,i^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}(v_{j^*,\mu^*})_{I^p_{\mu^*}}\\ &= \psi(p,i^*,\nu^*,\mu^*)\zeta_2(p,i^*,\mu^*)\zeta_3(p,i^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}(v_{j^*,\mu^*})_{I^p_{\mu^*}}\\ &\quad -\delta_{\mu^*,\nu^*}\psi(p,i^*,\nu^*,\mu^*)\zeta_2(p,i^*,\mu^*)\zeta_3(p,i^*,\nu^*)(v_{m^*,\nu^*})_{I^p_{\nu^*}}(v_{j^*,\mu^*})_{I^p_{\mu^*}}.\end{aligned}$$

The first term separates the variables and leads to a rank two matrix plus a block diagonal matrix with blocks of rank one, the second term leads to a block diagonal matrix with blocks of rank one. Therefore, the matrix-vector product is of complexity $\mathcal{O}(\sum_{\nu=1}^{d} k_\mu)$ for each $C^{3,i^*,p}$ and thus in total $\mathcal{O}(Pk\sum_{\nu=1}^{d} k_\mu)$ for C^3.

■

Theorem 24 *The Hessian H allows for a matrix-vector multiplication in $\mathcal{O}(Pk^2\sum_{j=1}^{d} k_j)$. The storage complexity is $\mathcal{O}(Pkd+Pk^2+P\sum_{j=1}^{d} k_j)$ in addition to the storage requirements needed for the computation of the gradient.*

Proof: Combine Remark 20 and Lemmata 21,22,23. ■

3.5 Constraints in the Minimization

For stability reasons it is (sometimes) necessary to bound the norm of the addends X_i. This can be accomplished by adding an additional term

$$f_{\text{norm}}(\alpha) := c_{\text{norm}}\sum_{i=1}^{k}\sum_{\mu=1}^{d}\sum_{j=1}^{k_\mu}\alpha^2_{i,j,\mu}$$

to the target functional f. The penalty method (parameter c_{norm}) is best suited because we do not need a strict bound on the norm. Furthermore, we choose the full ℓ_2-norm in order to obtain a simple matrix structure in the Hessian.

Lemma 25 *The first partial derivative of f_{norm} with respect to the variable α_{i^*,j^*,μ^*} is*

$$\partial_{\alpha_{i^*,j^*,\mu^*}} f_{\text{norm}}(\alpha) = 2c_{\text{norm}}\alpha_{i^*,j^*,\mu^*}$$

The second partial derivative with respect to the variable $\alpha_{\ell^*,m^*,\nu^*}$ is

$$\partial_{\alpha_{\ell^*,m^*,\nu^*}}\partial_{\alpha_{i^*,j^*,\mu^*}} f_{\text{norm}}(\alpha) = 2c_{\text{norm}}\delta_{i^*,\ell^*}\delta_{j^*,m^*}\delta_{\mu^*,\nu^*}.$$

In Lemma 3 we have pointed out that the representation system for a rank k tensor is not unique, which is due to the fact that for an elementary tensor $\bigotimes_{\mu=1}^{d} x_\mu$ one can scale all but one factor by a nonzero constant and divide the one factor by the product of the constants without changing the tensor:

$$\bigotimes_{\mu=1}^{d} x_\mu = \frac{x_1}{\lambda_2 \cdots \lambda_d} \otimes \bigotimes_{\mu=2}^{d} \lambda_\mu x_\mu, \quad \lambda_\mu \in \mathbb{R} \setminus \{0\}.$$

This non-uniqueness can be remedied by an additional penalty term

$$f_{\mathrm{rep}}(\alpha) := c_{\mathrm{rep}} \sum_{i=1}^{k} \sum_{\mu=1}^{d} \sum_{\nu=1}^{d} \left(\|x_{i,\mu}\|^2 - \|x_{i,\nu}\|^2 \right)^2,$$

which enforces an equilibration of the norms of the factors for each elementary tensor.

Lemma 26 *The first partial derivative of f_{rep} with respect to the variable α_{i^*,j^*,μ^*} is*

$$\partial_{\alpha_{i^*,j^*,\mu^*}} f_{\mathrm{rep}}(\alpha) = 8 c_{\mathrm{rep}} \alpha_{i^*,j^*,\mu^*} \sum_{\mu=1}^{d} \left(\|x_{i^*,\mu^*}\|^2 - \|x_{i^*,\mu}\|^2 \right)$$

The second partial derivative with respect to the variable $\alpha_{\ell^*,m^*,\nu^*}$ is

$$\begin{aligned} \partial_{\alpha_{\ell^*,m^*,\nu^*}} \partial_{\alpha_{i^*,j^*,\mu^*}} f_{\mathrm{rep}}(\alpha) \;=\; & 8 c_{\mathrm{rep}} \delta_{i^*,\ell^*} \delta_{j^*,m^*} \delta_{\mu^*,\nu^*} \sum_{\mu=1}^{d} \left(\|x_{i^*,\mu^*}\|^2 - \|x_{i^*,\mu}\|^2 \right) \\ & + 16 c_{\mathrm{rep}} d \delta_{i^*,\ell^*} \delta_{\mu^*,\nu^*} \alpha_{i^*,j^*,\mu^*} \alpha_{i^*,m^*,\mu^*} \\ & - 16 c_{\mathrm{rep}} \delta_{i^*,\ell^*} \alpha_{i^*,j^*,\mu^*} \alpha_{i^*,m^*,\nu^*} \end{aligned}$$

Proof: Elementary calculation and using the fact that the $v_{\cdot,\mu}$ are orthonormal:

$$\|x_{i,\mu}\|^2 = \left\langle \sum_{j=1}^{k_\mu} \alpha_{i,j,\mu} v_{j,\mu}, \sum_{j=1}^{k_\mu} \alpha_{i,j,\mu} v_{j,\mu} \right\rangle = \sum_{j=1}^{k_\mu} \alpha_{i,j,\mu}^2.$$

■

A rebalancing of a rank k tensor so that for each i the factors of the i-th elementary tensor are of the same norm will ensure that the gradient of f_{rep} vanishes and the Hessian of f_{rep} becomes

$$H^{\mathrm{rep}}_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} = 16 c_{\mathrm{rep}} \delta_{i^*,\ell^*} \alpha_{i^*,j^*,\mu^*} (d \delta_{\mu^*,\nu^*} - 1) \alpha_{i^*,m^*,\nu^*}$$

which allows a matrix-vector multiplication in $\mathcal{O}(k \sum_{\mu=1}^{d} k_\mu)$.

4 Newton's Method for Partial Minimization

In the previous sections we have obtained efficient procedures for the computation of the gradient and evaluation of the Hessian. Since our target function $f(\alpha)$ (with or without additional constraint functions $f_{\mathrm{norm}}(\alpha)$ and $f_{\mathrm{rep}}(\alpha)$) is in general not convex and the Hessian is in general not positive definite, we have to modify Newton's method in order to find a descent direction and to obtain a globally convergent scheme. Although the scheme converges globally, it will not necessarily converge to a global optimum.

4.1 Finding a Descent Direction by Trust Region

The trust region subproblem is

$$\begin{aligned} \min \quad & g(p) := f(\alpha) + \langle \nabla f(\alpha), p\rangle + \frac{1}{2}\langle H(\alpha)p, p\rangle \\ \text{s.t.} \quad & \|p\|_{T(\alpha)} \leq r, \end{aligned}$$

for some parameter $r \in \mathbb{R}_+$ and a positive definite matrix $T(\alpha)$. The advantage of the trust region approach is partially due to the fact that $H(\alpha)$ is not required to be positive definite. So in particular the Hessian can be used even if it is singular or indefinite. For every r there exists exactly one solution of the trust region subproblem. This solution satisfies the following equation

$$p(r) = \alpha - (\lambda T(\alpha) + H(\alpha))^{-1}\nabla f(\alpha),$$

where $\lambda \in \mathbb{R}_+$ is uniquely determined by the problem

$$\varphi(\lambda) := \|(\lambda T(\alpha) + H(\alpha))^{-1}\nabla f(\alpha)\|_{T(\alpha)} = r,$$

see [5] for more details. The substitution $\omega := (1-\lambda)/\lambda$ leads to

$$\hat{H}(\alpha, \omega) := \omega H(\alpha) + (1-\omega)T(\alpha),$$

with $\omega \in [0, 1]$.

The standard choice would be $T(\alpha) := \mathbf{Id}$, the gradient direction, but in practice we observe that the positive definite matrix

$$T(\alpha) := D^1,$$

see Corollary 19, gives much better results. Thus we obtain a direction

$$d(\alpha, \omega) := (\hat{H}(\alpha, \omega))^{-1} g(\alpha)$$

which is for $\omega = 1$ the Newton direction and for $\omega \to 0$ a descent direction, where $g(\alpha) := \nabla f(\alpha)$. The parameter ω is chosen to be 0.95^j, where $j \in \mathbb{N}_0$ is the smallest integer such that the direction $d(\alpha, 0.95^j)$ fulfils

$$\langle g(\alpha), d(\alpha, 0.95^j)\rangle \geq 10^{-8}\|g(\alpha)\|\,\|d(\alpha, 0.95^j)\|. \tag{16}$$

Remark 27 *When solving the system $\hat{H}$ iteratively by the cg-iteration, one can exploit that the convergence depends on the positivity of $\hat{H}$. As long as the iteration diverges, we decrease ω. Also, when (16) is not fulfiled we restart and decrease ω. Thus, during the iterative solve the parameter ω can be determined.*

As already mentioned in Remark 20, our preconditioner for the linear system is $(D^1)^{-1}$. Where the computation of the inverse has a complexity of $\mathcal{O}(dk^3)$, see Remark 20. Because of the splitting of $H(\alpha)$ into a sum of matrices, see e.g. Definition 15, we can write

$$H(\alpha) = D^1 + R(\alpha),$$

with a matrix R defined by the splitting. It follows that

$$\begin{aligned} (D^1)^{-1}\hat{H}(\alpha, \omega) &= (D^1)^{-1}(\omega H(\alpha) + (1-\omega)T(\alpha)) \\ &= \mathbf{Id} + \omega(D^1)^{-1}R(\alpha). \end{aligned}$$

As a consequence, this choice improves the condition number of $(D^1)^{-1}\hat{H}(\alpha, \omega)$ especially in problematic cases, i.e. if we have $\omega \to 0$.

4.2 Line Search by Armijo Rule

Let $d(\alpha)$ be a suitable descent direction, i.e., a descent direction that is close to the Newton direction and fulfils (16). Our aim is to perform a line search

$$\text{minimize} \qquad \lambda \mapsto f(\alpha) - f(\alpha + \lambda d(\alpha))$$

which requires the evaluation of the one-dimensional function $\lambda \mapsto f(\alpha + \lambda d(\alpha))$ for several $\lambda \in \mathbb{R}$. This can best be done by changing the representation of $X(\alpha - \lambda d(\alpha))$ to the form (7) and then computing

$$\begin{aligned} f(\alpha + \lambda d(\alpha)) &= \langle A - X(\alpha + \lambda d(\alpha)), A - X(\alpha + \lambda d(\alpha))\rangle_P \\ &= \sum_{p=1}^{P}\sum_{\mu=1}^{d} \langle A - X(\alpha + \lambda d(\alpha)), A - X(\alpha + \lambda d(\alpha))\rangle_{p,\mu} \\ &= \sum_{p=1}^{P}\sum_{\mu=1}^{d} \langle \underbrace{A|_{J^{p,\mu}}}_{\in\mathbb{R}^{n_\mu}} - \underbrace{X(\alpha + \lambda d(\alpha))|_{J^{p,\mu}}}_{\in\mathbb{R}^{n_\mu}}, \underbrace{A|_{J^{p,\mu}}}_{\in\mathbb{R}^{n_\mu}} - \underbrace{X(\alpha + \lambda d(\alpha))|_{J^{p,\mu}}}_{\in\mathbb{R}^{n_\mu}} \rangle \end{aligned}$$

in $\mathcal{O}(P\sum_{\mu=1}^{d} n_\mu)$. The costs are negligible when compared to the complexity for the inversion of the Hessian and they are comparable to those for the computation of the gradient.

The Armijo rule is characterized by the choice

$$\lambda = \max\{0.5^j \mid j \in \mathbb{N}_0\} \quad s.t. \quad f(\alpha) - f(\alpha + \lambda d(\alpha)) \geq \frac{\lambda}{5}\langle g(\alpha), d(\alpha)\rangle.$$

4.3 Finding Initial Guess and Pivot Points

Newton's method will only converge *locally* quadratically, and the overall performance depends critically on the initial guess. Therefore, we require a scheme where the initial guess is constructed in a suitable way. We propose to construct successively a rank $1, \ldots, k$ approximation X_k, and use as an initial guess for the rank $i+1$ Newton iteration the tensor

$$X_{i+1}^{(0)} := X_i + R_i, \qquad R_i \in \mathcal{T}(d, 1),$$

where R_i is a rank one approximation of the remainder $A - X_i$. This rank one approximation can be chosen to be a rank one cross approximation, see the following section.

5 Successive Rank One Approximation

Definition 28 *A rank one cross approximation $X = \otimes_{\mu=1}^{d} x_\mu$ of a tensor A is defined by a pivot point $I = (I_1, \ldots, I_d)$, where $A_I \neq 0$, and*

$$(x_\mu)_i := A_J/A_I, \qquad J_\nu = \begin{cases} I_\nu & \nu \neq \mu \\ i & \nu = \mu \end{cases} \qquad \nu = 1, \ldots, d, \quad i = 1, \ldots, n_\mu, \quad \mu > 1,$$

$$(x_1)_i := A_J, \qquad J_\nu = \begin{cases} I_\nu & \nu \neq 1 \\ i & \nu = 1 \end{cases} \qquad \nu = 1, \ldots, d, \quad i = 1, \ldots, n_1.$$

The rank one cross approximation from Definition 28 can be used successively in order to construct a rank k tensor approximation of A: Let X_i be the i-th term. For $i = 1, \ldots, k$ the elementary tensor X_i is a rank one cross approximation of

$$A - \sum_{j=1}^{i-1} X_j.$$

However, we are not aware of a choice of the pivot elements such that the approximation is close to a best approximation. In dimension $d = 2$ good pivot elements are the maximal entries in modulus, which is not sufficient in dimension $d > 2$ (see the following Section on numerical examples). In order to find an element of $A - \sum_{j=1}^{i-1} X_j$ with large absolute value, we first determine a (random) position I where $A_I \neq 0$ and then we circulate through all the d fibres intersecting the position I and pick the position with largest entry in modulus. This is the pivot element. If a nonzero element is not detectable, then the algorithm stops.

6 Numerical Examples

In this section we investigate the optimal choice of parameters and the behaviour of the black box approximation for different tensors.

Our first numerical test concerns the question whether or not the minimization of the function f is necessary.

6.1 Successive rank 1 cross approximation

We consider the tensor A of order $d = 2, 3, 4, 5$ and rank $k_A = 4$ defined by the evaluation of the function

$$a(x_1, \ldots, x_d) = \sum_{i=1}^{4} \prod_{\mu=1}^{d} (x_\mu)^i \tag{17}$$

on a uniform grid in the cube $[0, 1]^d$ with $n = (21)^d$ mesh-points:

$$A_{(i_1,\ldots,i_d)} = a(i_1/20, \ldots, i_d/20), \quad i_\mu = 0, \ldots, 20, \quad \mu = 1, \ldots, d. \tag{18}$$

We successively apply the rank 1 cross approximation from Definition 28 (see previous section) in order to obtain a low rank approximation X of the tensor A.

The results in Figure 2 show that the pure cross approximation without minimization does not give good results. Whereas in $d = 2$ the convergence is good, it almost breaks down for dimension $d = 3$ and becomes even worse as d increases. We conclude that the cross approximation itself is not suitable for the approximation of tensors of order $d > 2$. Therefore, we need a different approach that is able to find better low rank approximations. The question arises whether or not a variant of the cross approximation might be suitable. This question will be answered in the next section.

6.2 The optimal choice of the number of pivots

In the following numerical test we fix the tensor A to be approximated and vary the number of pivots we use. The minimal reasonable choice is $P = k$ which means that we read

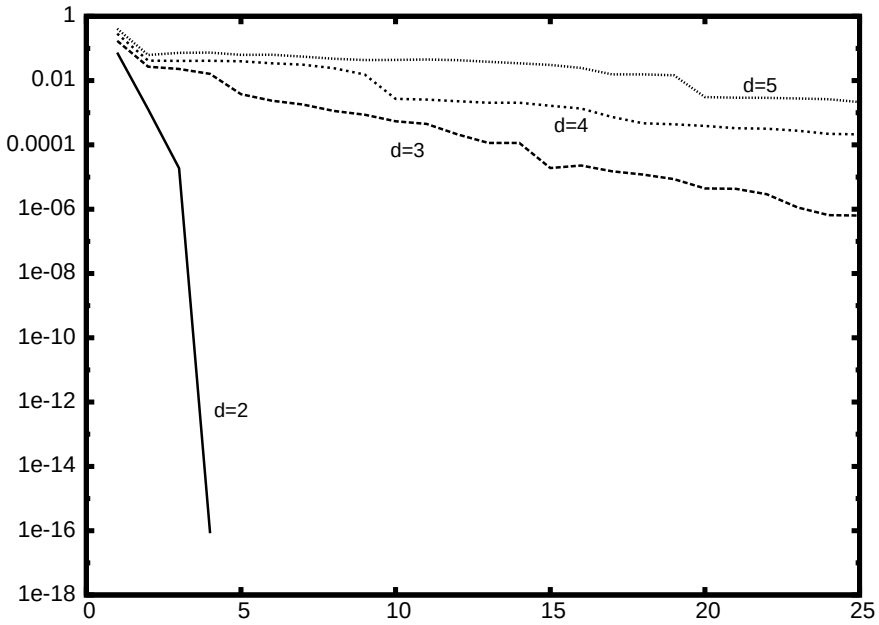


Figure 2: The relative error $\epsilon = \|A - X\|/\|A|$ of a pure cross approximation with $k = 1, \ldots, 25$ terms in $d = 2, \ldots, 5$.

$P(d(n-1)+1)$ values from the input tensor A (assuming that there is no overlap between the fibers of different crosses) and construct a tensor X with $k(d(n-1)+1)$ degrees of freedom. In Table 1 we observe that the choice $p = k$ is not suitable: the input tensor is the tensor A from the previous section (rank 4), the approximation X is of rank $k = 1, \ldots, 6$ and interpolates the tensor A on all $p = k$ (non-intersecting) crosses, but $A \neq X$. For the first three steps $k = 1, \ldots, 3$ the approximation is good, but afterwards it stagnates or becomes worse.

k	$\frac{\|A-X(\alpha)\|}{\|A\|}$	$\langle A - X(\alpha), A - X(\alpha)\rangle_{\mathcal{P}}$
1	2.9×10^{-1}	2×10^{-28}
2	1.3×10^{-2}	5×10^{-30}
3	1.1×10^{-4}	1×10^{-29}
4	7.2×10^{-4}	1×10^{-29}
5	1.7×10^{-3}	2×10^{-29}
6	8.2×10^{-3}	4×10^{-29}

Table 1: Approximation of a rank 4 tensor in $d = 4$ using $P = k$ pivots.

The results from this numerical test as well as from several other tests we performed indicate a general feature summarized in the following conjecture.

Conjecture 29 *Any tensor A of order d can be interpolated on p crosses by a tensor of rank $k = p$.*

There are two conclusions from our observation: First, it is not sufficient to read the data from $k(d(n-1)+1)$ points in order to find a rank k tensor approximation. In particular,

from k crosses one can not expect a reasonable rank k approximation no matter how the approximation is constructed from the crosses.

Second, we conclude that the number P of pivot points should be larger than the rank k of the tensor X. It is clear that for large enough P we obtain the deterministic minimization problem to approximate a full tensor by a low rank one, but P enters the complexity linearly so that we want to keep P reasonably small.

In order to find the (more or less) optimal value for P, we perform a numerical test. We approximate a tensor A of order $d \in \{3, 4\}$ by a low rank tensor using $P = ck$ pivots, $c \in \{1, 2, 3, 6\}$. The tensor A is the evaluation of the function

$$a(x_1, \ldots, x_d) = \left(\sum_{\mu=1}^{d} x_\mu^2 \right)^{-1/2} \tag{19}$$

on a uniform grid in the cube $[1, 2]^d$ with $n = (21)^d$ mesh-points:

$$A_{i_1,\ldots,i_d} = a(1 + i_1/20, \ldots, 1 + i_d/20), \quad i_\mu = 0, \ldots, 20, \quad \mu = 1, \ldots, d. \tag{20}$$

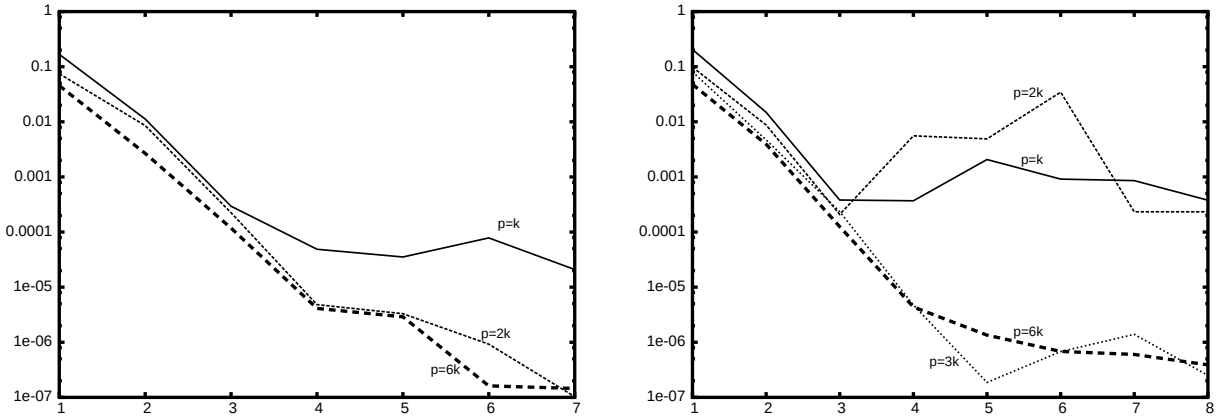


Figure 3: Different number of pivot points p in $d = 3, 4$.

For $c = 1$ and $c = 2$ the results in Figure 3 are not convincing since in dimension $d = 4$ the approximation accuracy stagnates. The value $c = 3$ seems to be sufficient. In order to be more conservative we choose a larger value for P:

$$P := 3 + 3k + \lfloor k^2/2 \rfloor \tag{21}$$

The constant 3 is chosen in order to provide enough information for the initial crosses and the quadratic term gives more information for higher ranks.

6.3 Reconstruction of low rank tensors

Now that we have obtained a suitable value for the number of pivot points P we can try to reconstruct the rank $k_A = 4$ tensor A from Section 6.1. The results for $d = 3, 4, 5, 6$ are contained in Table 2. In the first three cases the tensor is reconstructed, in the last case $d = 6$ we have computed a local minimum where the gradient of $f(\alpha)$ is zero, the Hessian

k	$d=3$	$d=4$	$d=5$	$d=6$
1	1.4×10^{-1}	2.3×10^{-1}	3.5×10^{-1}	4.5×10^{-1}
2	4.6×10^{-3}	9.5×10^{-3}	1.6×10^{-2}	2.7×10^{-2}
3	3.4×10^{-5}	7.5×10^{-5}	1.9×10^{-4}	3.6×10^{-4}
4	4.0×10^{-16}	9.2×10^{-16}	2.3×10^{-12}	2.1×10^{-4}

Table 2: The table contains the relative error $\|A - X(\alpha)\|/\|A\|$ of the approximation of the rank 4 order d tensor A by a tensor X of rank k using P pivots (P as in (21)).

$H(\alpha)$ is positive, but $f(\alpha) \neq 0$. Depending on the initial guess and the pivots, this happens as well for other dimensions.

Our conclusion is that local minima do exist and cannot be trivially avoided. The relative difference in the target function f between the local minimum and the global minimum can be arbitrarily large. A more involved investigation is necessary and will be performed in a followup article.

6.4 Approximation of a smooth non-separable function

At last we consider the d-variate function a from (19),

$$a(x_1, \ldots, x_d) = \left(\sum_{\mu=1}^{d} x_\mu^2 \right)^{-1/2}$$

and the corresponding tensor A of order d defined by

$$A_{i_1,\ldots,i_d} = a(1 + i_1/14, \ldots, 1 + i_d/14), \quad i_\mu = 0, \ldots, 20, \quad \mu = 1, \ldots, d.$$

We perform the black box approximation for this tensor with the parameters described above. In the Newton iteration we use the stopping criterion $\|\nabla f(\alpha^i)\| < 10^{-10}$ or

$$\frac{f(\alpha^{i-10}) - f(\alpha^i)}{f(\alpha^{i-10})} < 10^{-6},$$

i.e., we stop of either the gradient is close to zero or if after ten iterations there is almost no progress. The results of the test for dimension $d = 3, 4$ are reported in Table 3 and the error decay is depicted in Figure 4.

The number of Newton steps that are required to find a local minimum is quite large, and there is definitely more work necessary to enhance the convergence.

References

[1] G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

[2] S. R. Chinnamsetty, M. Espig, B. N. Khoromskij, W. Hackbusch, and H. J. Flad. Tensor product approximation with optimal rank in quantum chemistry. Technical Report 41/2007, Max Planck Institute for Mathematics in the Sciences, 2007. Submitted to Journal of Chemical Physics.

	$d=3$				$d=4$			
k	$\frac{\|A-X(\alpha)\|}{\|A\|}$	$f(\alpha)$	$\|\nabla f(\alpha)\|$	Steps	$\frac{\|A-X(\alpha)\|}{\|A\|}$	$f(\alpha)$	$\|\nabla f(\alpha)\|$	Steps
1	4.5×10^{-2}	1.8×10^{-1}	2.1×10^{-12}	5	7.4×10^{-2}	2.5×10^{-1}	1.3×10^{-14}	7
2	3.4×10^{-3}	4.2×10^{-4}	3.4×10^{-14}	25	4.6×10^{-3}	1.5×10^{-4}	5.0×10^{-14}	28
3	1.2×10^{-4}	7.4×10^{-7}	1.6×10^{-12}	100	1.7×10^{-4}	6.3×10^{-7}	1.2×10^{-14}	85
4	4.0×10^{-6}	1.9×10^{-9}	5.4×10^{-11}	257	4.9×10^{-6}	1.4×10^{-9}	9.8×10^{-14}	182
5	1.7×10^{-7}	4.9×10^{-12}	6.0×10^{-12}	1109	1.5×10^{-7}	2.8×10^{-12}	4.5×10^{-14}	872
6	1.8×10^{-8}	9.3×10^{-14}	1.4×10^{-8}	2642	1.4×10^{-8}	4.6×10^{-14}	9.7×10^{-8}	1711

Table 3: The relative error, the target function value, the norm of the gradient and the number of Newton iterations for the rank $k = 1, \ldots, 6$ approximation of an order $d = 3, 4$ tensor.

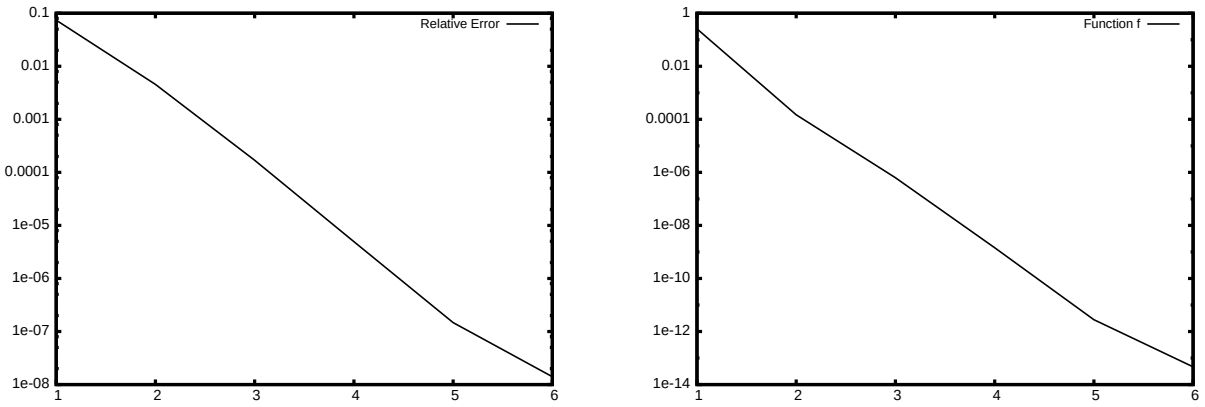


Figure 4: The decay of the relative error (left) and the decay of the target function (right) for the rank $k = 1, \ldots, 6$ approximation an order $d = 4$ tensor.

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

[4] V. de Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. Technical Report SCCM-06-06, Stanford University, 2006.

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

[6] F. W. Young J. de Leeuw, Y. Takane. Additive structure in qualitative data: an alternating least squares method with optimal scaling features. *Psychometrika*, 41:471–503, 1976.

[7] T. G. Kolda. Orthogonal tensor decompositions. *SIAM J. Matrix Anal. Appl.*, 23(1):243–255, 2001.

[8] I. V. Oseledets and E. E. Tyrtyshnikov D. V. Savost'yanov. Tucker dimensionality reduction of three-dimensionala arrays in linear time. *SIAM J. Matrix Anal. Appl.*, 2008. to appear.

[9] I. V. Oseledets and D. V. Savost'yanov. Minimization methods for approximating tensors and their comparison. *Computational Mathematics and Mathematical Physics*, 46(10):1641–1650, 2006.

[10] P. Paatero. A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 38:223–242, 1997.

[11] T. Zhang and G. H. Golub. Rank-one approximation to high order tensors. *SIAM J. Matrix Anal. Appl.*, 23(2):534–550, 2001.

Preprint Series DFG-SPP 1324

http://www.dfg-spp1324.de

Reports

[1] R. Ramlau, G. Teschke, and M. Zhariy. A Compressive Landweber Iteration for Solving Ill-Posed Inverse Problems. Preprint 1, DFG-SPP 1324, September 2008.

[2] G. Plonka. The Easy Path Wavelet Transform: A New Adaptive Wavelet Transform for Sparse Representation of Two-dimensional Data. Preprint 2, DFG-SPP 1324, September 2008.

[3] E. Novak and H. Woźniakowski. Optimal Order of Convergence and (In-) Tractability of Multivariate Approximation of Smooth Functions. Preprint 3, DFG-SPP 1324, October 2008.

[4] M. Espig, L. Grasedyck, and W. Hackbusch. Black Box Low Tensor Rank Approximation Using Fibre-Crosses. Preprint 4, DFG-SPP 1324, October 2008.