Constr Approx (2009) 30: 557–597
DOI 10.1007/s00365-009-9076-9

CONSTRUCTIVE APPROXIMATION

Black Box Low Tensor-Rank Approximation Using Fiber-Crosses

Mike Espig · Lars Grasedyck · Wolfgang Hackbusch

Received: 12 September 2008 / Revised: 30 April 2009 / Accepted: 6 May 2009 /
Published online: 8 October 2009
© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract In this article we introduce a black box type algorithm for the approximation of tensors A in high dimension d. The algorithm adaptively determines 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 fiber-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 fiber-crosses and d the order of the tensor.

Keywords Low rank · Tensor · Newton · Cross approximation · Fiber-crosses · Black box

Mathematics Subject Classification (2000) 15A69 · 90C06 · 65K10

Communicated by Christoph Schwab.

M. Espig · L. Grasedyck (✉) · W. Hackbusch
Max Planck Institute for Mathematics in the Sciences, Inselstrasse 22-26, 04103 Leipzig, Germany
e-mail: lgr@mis.mpg.de

M. Espig
e-mail: espig@mis.mpg.de

W. Hackbusch
e-mail: wh@mis.mpg.de

L. Grasedyck
Berlin Mathematical School (BMS), TU-Berlin, Berlin, Germany

1 Introduction

In general, computations with tensors $A \in \mathbb{R}^{n^d}$ require 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

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

which allows the tensor X to be stored in $\mathcal{O}(kdn)$. The minimal number k of addends required for such a representation 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

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

is not a linear space. In particular, the sum of two tensors of tensor rank k 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 analyzed in detail, and it is highly non-trivial. It is neither the case that a best approximation of rank k' always exists [6], nor that a polynomial-time algorithm to compute a best approximation is known (provided it exists). Nonetheless, available state-of-the-art methods are quite efficient and have proven to be reliable in many practical cases [1, 4, 7].

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

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

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: a straight-forward multivariate interpolation of order m requires m^{d-1} interpolation points, i.e., the number of addends in the initial approximation is $k = m^{d-1}$. Already for dimension $d = 10$ and order $m = 5$ this is more than one million.

The only reference—to our knowledge—for an alternative approach in the literature is the three-dimensional cross approximation [13] (see also [2] for low rank approximation on given point sets and [8] for an application). The construction is explicit and does not involve a minimization step.

We propose a direct minimization of the distance between the tensor A and an element $X \in \mathcal{T}(k,d,n)$, 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 indices. The choice of the indices is adapted to the tensor A and will be determined

on the fly. The restriction to a small set of indices is reasonable because the low rank format involves only kdn data, whereas the whole tensor contains n^d data.

We have several possible applications in mind:

(1) *Fast evaluation of multiparametric expensive functions:* In many applications, e.g., when computing boundary integrals involving singular kernel functions or when investigating the behavior of solutions of linear or nonlinear equations, the computation of the result $u(\alpha_1, \alpha_2, \ldots)$ for a fixed set of parameters $(\alpha_1, \alpha_2, \ldots)$ is in principle possible but only with a nontrivial complexity. When the dependency on the parameters is smooth and the number of parameters larger than 2, then it is reasonable to seek a simpler representation of u (in terms of the parameters) in low rank format and to evaluate this low rank representation for many parameter combinations instead of computing u anew for each of them. Also, the low rank format allows for a further analysis (maxima, minima etc. [7]) avoiding the computation of all parameter combinations entirely.

(2) *Initial low rank approximation:* In [9] a method for the fast solution of special high-dimensional partial differential equations (PDE) is presented which requires the right-hand side of the PDE to be in low rank format. If this is not the case, then our black box algorithm can produce an initial approximation of the right-hand side in this format.

(3) *Investigation of new functions:* The possibility of approximating a high-dimensional function by a sum of few separable functions has only been investigated for a small number of functions (typically variants of $1/\|x\|$). Our black box algorithm provides a tool for investigating the approximability numerically, which might then lead to a deeper analysis of the underlying function in order to prove the desired approximability.

These applications, however, are beyond the scope of this article and will be covered in forthcoming articles.

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 indices. Also, samples in a random selection of indices will typically not lead to acceptable results.

The article is structured as follows. In Sect. 2 we introduce some notation and summarize results on the approximation of tensors in subspace bases. The minimization problem for the approximation in low rank format is presented in Sect. 3. In Sects. 4 and 5 we specify the choice of the pivot indices and define the initial guess necessary for the partial minimization. The minimization problem is then treated in Sect. 6 by Newton's method. Finally, we present numerical results in Sect. 7.

2 Approximation of Tensors in Subspaces

2.1 Basic Definitions

We begin 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:

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

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

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

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

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

The elements of $\mathcal{T}(k, d)$ are called *rank* k tensors. The representation (2) is the low rank representation of elements from $\mathcal{T}(k, d)$.

Notation 3 For the rest of the article, we fix the integers $d, n_1, \ldots, n_d$ and the index set

$$\mathcal{I} := \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_d\}.$$

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 $d \ge 3$, i.e., the sum of k rank one tensors (elementary tensors). In the literature the terms *PARAFAC* (parallel factor) rank, *CANDECOMP* (canonical decomposition) rank or *Kronecker* rank are sometimes used.

Lemma 4 *The storage complexity $N_{\mathrm{St},K}(d,k)$ for a tensor $X \in \mathcal{T}(k,d)$ in the form (2) is*

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

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 $\|x_{i,\mu}\|_2 = 1$ $(\mu = 1, \ldots, d,\ i = 2, \ldots, k)$:

$$\lambda_1 x_1 \otimes \lambda_2 x_2 \otimes \cdots \otimes \lambda_d x_d = \big((\lambda_1 \cdots \lambda_d) x_1\big) \otimes x_2 \otimes \cdots \otimes x_d. \qquad \square$$

A different kind of rank will be introduced next. If we write an $n \times m$ matrix R of rank k as the linear combination of basis vectors U_i for the column span times basis vectors V_j for the row span

$$R = \sum_{i=1}^{k} \sum_{j=1}^{k} C_{ij} U_i V_j^T,$$

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

Definition 5 (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 orthonormal 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

$$T = \sum_{i_1=1}^{k_1} \cdots \sum_{i_d=1}^{k_d} C_{(i_1,\ldots,i_d)} \bigotimes_{\mu=1}^{d} u_{i_\mu,\mu}, \qquad \langle u_{i,\mu}, u_{j,\mu} \rangle = \delta_{i,j}. \tag{3}$$

The representation of the form (3) is called the Tucker format, or in short we say T is a Tucker tensor. The set of tensors of Tucker rank at most $(k_1, \ldots, k_d)$ is denoted by

$$\mathrm{Tucker}(k_1, \ldots, k_d) := \big\{ X \in \mathbb{R}^{n_1 \times \cdots \times n_d} \mid \mathrm{Tucker\ rank}(X) \leq (k_1, \ldots, k_d) \big\}.$$

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

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

Proof The storage complexity involves two parts: 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. A related approach for the combined approximation using the orthonormal vectors u from the Tucker format and a rank k representation of the core tensor is presented in [10].

2.2 Decomposition and Approximation of Tensors in Tucker Format

Definition 7 (Fiber) Let $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mu \in \{1, \ldots, d\}$. For a multi-index $(i_1, \ldots, i_d) \in \mathcal{I}$ we define

$$I(\mu, j) := (i_1, \ldots, i_{\mu-1}, j, i_{\mu+1}, \ldots, i_d), \quad j = 1, \ldots, n_\mu.$$

The set of *fibers* of A in direction μ is defined by

$$\mathcal{W}(A, \mu) := \left\{ w \in \mathbb{R}^{n_\mu} \;\middle|\; w_j = A_{I(\mu,j)},\ j = 1, \ldots, n_\mu,\ I \in \mathcal{I} \right\}.$$

Example 8 In the matrix case $d = 2$, the set $\mathcal{W}(A, 1)$ is the set of column vectors of A, and $\mathcal{W}(A, 2)$ is the set of row vectors of A.

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

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

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 fibers that will be chosen adaptively. We will describe the construction later; for now, let us assume that for each direction μ a set of orthonormal vectors

$$\mathcal{V}(X, \mu) = \{v_{1,\mu}, v_{2,\mu}, \ldots, v_{k_\mu,\mu}\} \subset \operatorname{span} \mathcal{W}(X, \mu), \quad \langle v_{i,\mu}, v_{j,\mu} \rangle = \delta_{i,j}, \tag{4}$$

is given.

Remark 9 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 $\widetilde{T}$, then the vectors $u'_{i,\mu}$ used for the representation cannot be described easily (for $d = 2$ they are the singular vectors corresponding to the largest singular values). They are in general not the left dominant singular vectors of the matrix of fibers from $\mathcal{W}(T, \mu)$. 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 [5]

$$\|T - \widetilde{T}\|^2 \leq \sum_{\mu=1}^{d} \sum_{i=k'_\mu+1}^{k_\mu} \left(\sigma_i^{(\mu)}\right)^2 \leq d \min_{S \in \mathrm{Tucker}(k'_1,\ldots,k'_d)} \|T - S\|^2,$$

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

We conclude that the restriction to subspaces $\mathcal{V}(A, \mu)$ 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 (2) where each vector is of the special structure

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

This means that we seek the low rank tensor approximation in the subspace spanned by the vectors $v_{j,\mu}$. Note that we are only interested in the basis vectors $v_{j,\mu}$ ($\mathcal{O}(\sum_{\mu=1}^{d} k_\mu n_\mu)$ data) and not in the core tensor C ($\mathcal{O}(\prod_{\mu=1}^{d} k_\mu)$ data). We might as well choose a complete basis $(v_{j,\mu})_{j=1}^{n_\mu}$ of $\mathbb{R}^{n_\mu}$ in order to avoid any approximation error due to the choice of the subspaces. This complete basis would not require any a priori knowledge of the tensor.

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, where the vectors belong to the span of selected orthonormal bases $\mathcal{V}(A, \mu) = \{v_{1,\mu}, \ldots, v_{k_\mu,\mu}\}$:

$$A \approx X = X(\alpha) := \sum_{i=1}^{k} \bigotimes_{\mu=1}^{d} \sum_{j=1}^{k_\mu} \alpha_{i,j,\mu} v_{j,\mu}. \tag{6}$$

Each of the addends $X_i := \bigotimes_{\mu=1}^{d} \sum_{j=1}^{k_\mu} \alpha_{i,j,\mu} v_{j,\mu}$ is an elementary tensor.

Remark 10 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 [11, 17]. For each rank k a different set of elementary tensors $X_1, \ldots, X_k$ might be necessary.

We keep in mind that we have to declare how the basis vectors $v_{i,\mu}$ should be chosen; this will be done in Sect. 4. The unknowns to be determined are the coefficients $\alpha_{i,j,\mu}$ which are in total $\sum_{\mu=1}^{d} k_\mu k$. This defines the trial or ansatz manifold in which we seek the approximation X of the tensor A.

3.1 Full Minimization

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

$$f(\alpha) := \|A - X\|_2^2 = \langle A - X, A - X\rangle, \qquad \langle x, y\rangle := \sum_{I\in\mathcal{I}} x_I y_I$$

is minimized. For this minimization of f we would have to access all entries of A. If the tensor A is already given in low rank format, then this minimization problem can be solved by a Gauss–Newton method [14], by Newton's method [12], by a modified trust region Newton method [7], or by an alternating least squares algorithm [1, 16]. In our setting, however, the tensor A is not yet given in low rank format, and we want to avoid the evaluation of A for too many indices $I = (i_1, \ldots, i_d)$. Instead, we pick a suitable subset of the fibers and minimize the difference between A and $X(\alpha)$ only on this subset of fibers, similar to a collocation scheme for finite elements.

3.2 Partial Minimization

Let $\mathcal{P} := \{I^1, \ldots, I^P\} \subset \mathcal{I}$ be a set of multi-indices. We define the index sets corresponding to fibers through these indices by

$$J^{p,\mu} := \left\{ I = (i_1, \ldots, i_d) \in \mathcal{I} \;\mid\; \forall \nu \in \{1, \ldots, d\} \setminus \{\mu\} : i_\nu = I^p_\nu \right\}.$$

The fibers $J^{p,1}, \ldots, J^{p,d}$ form a so-called fiber-cross, see Fig. 1.

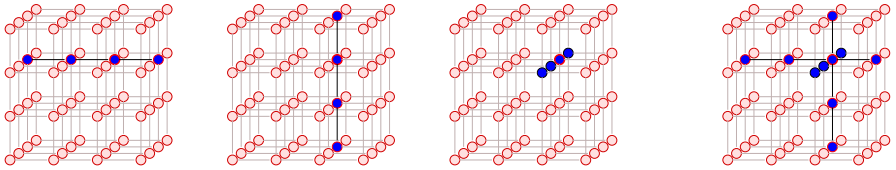


Fig. 1 The index sets $J^{p,1}, J^{p,2}, J^{p,3}$ for the pivot index $I^p := (3,3,3)$, and the whole fiber-cross (in $d = 3$)

We replace the full scalar product $\langle \cdot, \cdot \rangle$ by the bilinear form

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

The function to be minimized is now

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

The bilinear form $\langle \cdot, \cdot \rangle_{\mathcal{P}}$ requires only the evaluation of the tensor A at a small set of indices $I \in J^{p,\mu}$ corresponding to p fibers per direction $\mu = 1, \ldots, d$, each fiber being of length n_μ.

The choice of the set of pivot indices $\mathcal{P}$ will be described in Sect. 4. We start with an initial set $\mathcal{P}^{(0)}$ of p_0 pivot elements. In order to define the set of pivots for a rank k approximation, we use the set of pivots $\mathcal{P}^{(k-1)}$ from a rank $k-1$ approximation and add $c_{\mathcal{P}}$ new indices to obtain $\mathcal{P} = \mathcal{P}^{(k)}$. The vectors $v_{i,\mu}$ are defined as the dominant left singular vectors of the matrix of fibers through pivot indices in direction μ. For the minimization of $f_{\mathcal{P}}$ over the parameters $\alpha_{i,j,\mu}$ we employ a Newton type iteration (cf. Sect. 6). The complete procedure is described in short in Algorithm 1.

Algorithm 1 (Black Box Approximation—Overview)

1: Start with a small or empty set of pivot indices $\mathcal{P}^{(0)} = \{I^1, \ldots, I^{p_0}\}$
2: **for** $i = 1, \ldots, k$ **do**
3: **for** $p = p_0 + (i-1)c_{\mathcal{P}} + 1, \ldots, p_0 + i c_{\mathcal{P}}$ **do**
4: Enlarge the set of pivot indices:

$$\mathcal{P}^{(p)} := \{I^1, \ldots, I^p\} \supset \mathcal{P}^{(p-1)}$$

5: **for** $\mu = 1, \ldots, d$ **do**
6: Compute the (new) fibers $(w_{\ell,\mu})_j := A_{I^\ell(\mu,j)}$ in direction μ for all $I^\ell \in \mathcal{P}^{(p)}$ (cf. Definition 7).
7: Compute the dominant $k_\mu \le p$ left singular vectors $v_{1,\mu}, \ldots, v_{k_\mu,\mu}$ of the matrix

$$\tilde{A}^{(\mu)} := \big[w_{1,\mu} \mid \cdots \mid w_{p,\mu} \big]$$

8: **end for**
9: Minimize

$$f_{\mathcal{P}^{(p)}}(\alpha) := \langle A - X(\alpha), A - X(\alpha) \rangle_{\mathcal{P}^{(p)}}$$

for the representation

$$X(\alpha) := \sum_{\ell=1}^{i} \bigotimes_{\nu=1}^{d} \sum_{j=1}^{k_\nu} \alpha_{\ell,j,\nu} v_{j,\nu}$$

10: **end for**
11: **end for**

Lemma 11 *The complexity for Algorithm 1 is*

(1) *the cost for choosing the pivot elements* (*cf. Lemmata 12, 13*),
(2) *the cost for evaluating the tensor A on $P = p_0 + c_{\mathcal{P}} k$ crosses* (*assumed to be $\mathcal{O}(P \sum_{\mu=1}^{d} n_\mu)$*),
(3) *$\mathcal{O}(k P^2 \sum_{\mu=1}^{d} n_\mu)$ for the computation of the singular vectors, and*
(4) *the cost for the minimization step* (*cf. Lemma 15*).

In the following sections and the Appendix A we will specify the complexity for (1), (2), (4). Eventually, the complexity is dominated by the minimization step (4). For large P it is advisable to replace the determination of the dominant singular vectors in (3) by a complete basis of unit vectors.

4 Choice of the Pivot Elements

In this section we specify how one can find the pivot indices that define the fibers used for the construction of the basis vectors $v_{i,\mu}$ and for the partial scalar product $\langle \cdot, \cdot \rangle_{\mathcal{P}(p)}$.

The construction differs for the initial or first pivot elements and all other pivot elements.

4.1 Initial Pivot Elements

Often one has a priori information where the entries of the tensor A are large, e.g., positions of atoms in case of electron densities or reasonable parameter combinations for the approximation of multivariate functions. This can be used to define the initial pivot element or even a set $I^1, \ldots, I^{p_0}$ of initial pivot elements.

If there is no a priori information available, then we apply a greedy search. This greedy search need not be very accurate; it is sufficient that it give a good indication where the remainder $A - X(\alpha)$ of an approximation $X(\alpha)$ is relatively large. When choosing only a single pivot index—and this is the usual case that we are interested in—one could (theoretically) set

$$I^q := \operatorname*{argmax}_{I \in \mathcal{I}} \big| \big(A - X(\alpha)\big)_I \big|.$$

If we replace this full search by a partial (greedy) search, then we obtain a single pivot element I^q, cf. Algorithm 2.

Lemma 12 *Under the assumption that any entry of the tensor A can be obtained in $\mathcal{O}(1)$, the greedy pivot search from Algorithm 2 has a complexity of $\mathcal{O}(\ell_{\max} k \sum_{\mu=1}^{d} n_\mu)$, where $\ell_{\max}$ is the number of pivot search steps and k is the representation rank of the approximation X.*

Proof For each step ℓ in the greedy pivot search we loop over the dimension index μ and compute for all n_μ indices an entry of $A - X$. For A this complexity is

Algorithm 2 Greedy Initial Pivot Search

Given: a set of pivot indices $\mathcal{P}$, the tensor A and an approximation $X = X(\alpha)$.
Start with a (random) multi-index

$$I = (i_1, \ldots, i_d) \in \mathcal{I} \setminus \mathcal{P}.$$

for $\ell = 1, 2, \ldots, \ell_{\max}$ **do**
  for $\mu = 1, \ldots, d$ **do**
    Modify the index I in the μ-th component by

$$i_\mu := \operatorname*{argmax}_{j \in \{1,\ldots,n_\mu\}} \left|(A - X)_{(i_1,\ldots,i_{\mu-1},j,i_{\mu+1},\ldots,i_d)}\right|$$

    but ensure $I \notin \mathcal{P}$.
  end for
end for
Returnvalue: the pivot index I.

$\mathcal{O}(\sum_{\mu=1}^{d} n_\mu)$, but the evaluation of X has the dominating complexity of $\mathcal{O}(k)$ per entry, which sums up to the given bound. The efficient evaluation of X is based on the function $\gamma(p, j, \mu) = \prod_{\nu=1,\nu\neq\mu}^{d} (x_{j,\nu})_{I_\nu^p}$, which can be updated when changing one of the indices I_ν^p (see also the Appendix A, Remark 21). □

4.2 Noninitial Pivot Elements

For the noninitial pivot elements we will severely restrict the set of indices where we seek large entries of the remainder $A - X$. Again, if a priori information is available one should use it as in the previous subsection. Otherwise, a random search is possible. Numerically this is unsatisfactory, since it is nondeterministic and thus not reproducible.

We propose to search for large entries of the remainder $A - X$ on all previous fibers followed by $\ell_{\max}$ steps of the simple greedy search, cf. Algorithm 3. If the evaluation of A is rather expensive, then it is advisable to set $\ell_{\max} := 0$. This is both efficient, because new entries of the tensor A need not be computed (for $\ell_{\max} = 0$), and reliable, because we use the given information of the remainder on the fibers where we have optimized the distance $A - X$.

In the numerical experiments of the last section, we will observe that it is not sufficient to have k pivot elements (crosses) for a rank k approximation, i.e., we need to generate more than one pivot element in each step. We denote the number of pivot elements that we generate for each rank one term by

$$c_{\mathcal{P}}.$$

Together with the initial p_0 pivot elements, we obtain $p_0 + k c_{\mathcal{P}}$ pivot elements, for a rank k approximation.

Algorithm 3 Pivot Search On Fiber-Crosses

Given: a set of pivot indices $\mathcal{P} = \{I^1, \ldots, I^p\}$, the tensor A and an approximation $X = X(\alpha)$.
for $\ell = 1, 2, \ldots, p$ **do**
  for $\mu = 1, \ldots, d$ **do**

$$I := \operatorname*{argmax}_{J \in J^{\ell,\mu} \setminus \{I^\ell_\mu\}} \left|(A - X)_J\right|$$

  end for
end for
for $\ell = 1, 2, \ldots, \ell_{\max}$ **do**
  for $\mu = 1, \ldots, d$ **do**
    Modify the index $I = (i_1, \ldots, i_d)$ in the μ-th component by

$$i_\mu := \operatorname*{argmax}_{j \in \{1,\ldots,n_\mu\}} \left|(A - X)_{(i_1,\ldots,i_{\mu-1},j,i_{\mu+1},\ldots,i_d)}\right|$$

    but ensure $I \notin \mathcal{P}$.
  end for
end for
Returnvalue: the pivot index I.

Lemma 13 *The fiber-cross based pivot search from Algorithm 3 has a complexity of* $\mathcal{O}(\#\mathcal{P} \sum_{\mu=1}^d n_\mu k)$, *where* k *is the representation rank of the approximation* X.

Proof In addition to Algorithm 2 and Lemma 12, we have to compute the entries of $A - X$ for all indices of all crosses. This is of complexity $\mathcal{O}(pk \sum_{\nu=1}^d n_\nu)$ (cf. Lemma 22). □

The pivot search restricted to the previous fibers can fail already in dimension $d = 2$ [3], although it is in many cases one of the best choices. Therefore, we extend Algorithm 3 by an additional greedy (random) search of Algorithm 2 (see also Sect. 7.4).

5 Rank One Cross Approximation

Based on a single cross $(J^{p,\mu})_{\mu=1}^d$ of a tensor A one can define a rank one interpolation X on the cross, i.e.,

$$\forall I \in J^{p,1} \cup \cdots \cup J^{p,d}: \quad A_I = X_I,$$

which yields a reasonable rank one approximation for the whole tensor.

Definition 14 A rank one cross approximation $X = \bigotimes_{\mu=1}^{d} x_\mu$ of a tensor A in a pivot index $I = (i_1, \ldots, i_d) \in \mathcal{I}$, where $A_I \neq 0$, is defined by

$$(x_1)_i := A_{I(1,i)} \quad (i \in \{1, \ldots, n_1\}),$$
$$(x_\mu)_i := A_{I(\mu,i)}/A_I \quad (i \in \{1, \ldots, n_\mu\}), \quad \text{for } \mu = 2, \ldots, d.$$

The rank one cross approximation from Definition 14 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 defined as 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 $d > 2$, as the following example shows.

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

$$a(x_1, \ldots, x_d) = \sum_{i=1}^{4} \prod_{\mu=1}^{d} (x_\mu)^i \tag{7}$$

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

$$A_{(i_1,\ldots,i_d)} = a(i_1/20, \ldots, i_d/20), \quad i_\mu = 0, \ldots, 20, \ \mu = 1, \ldots, d. \tag{8}$$

We successively apply the rank one cross approximation from Definition 14 (pivot index is the index of the maximal element in modulus of the remainder) in order to obtain a low rank approximation X of the tensor A.

The results in Fig. 2 show that the pure cross approximation without minimization does not give good results. Whereas in $d = 2$ the convergence is good, it slows down considerably for dimension $d = 3$, and becomes even worse as d increases. We conclude that the cross approximation itself is only suitable for finding the initial guess for a subsequent minimization.

6 Modified Newton's Method for Partial Minimization

All Newton methods are based on approximating the objective function locally by a quadratic model and then minimizing that function approximately, e.g., by Krylov subspace methods. The quadratic model of the objective function f at α^k along p is given by the expansion

$$f(\alpha^k + p) \approx q_k(p) := f(\alpha^k) + \langle \nabla f(\alpha^k), p \rangle + \frac{1}{2} \langle H_f(\alpha^k) p, p \rangle$$

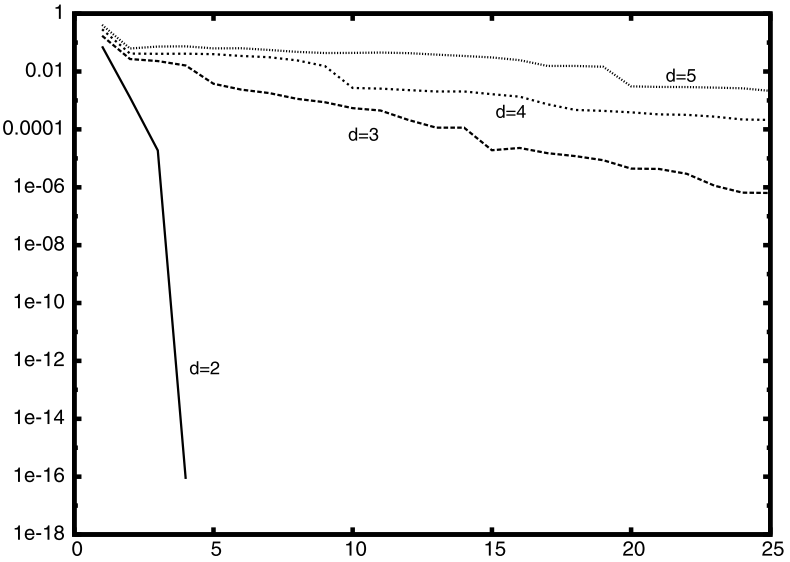


Fig. 2 The relative error $\epsilon = \|A - X\|/\|A|$ in log-scale of a pure cross approximation with $k = 1, \ldots, 25$ terms in $d = 2, \ldots, 5$

($H_f(\alpha^k)$ is the Hessian of f at α^k). The successor α^{k+1} is the minimizer of the following minimization problem:

$$\min_{p \in \mathbb{R}^n} q_k(p)$$

if and only if

$$\alpha^{k+1} = \alpha^k - d^k,$$

and the Hessian matrix $H_f(\alpha^k)$ is positive definite, where d^k solves the Newton equation

$$H_f\big(\alpha^k\big)d^k = \nabla f\big(\alpha^k\big).$$

Computational difficulties arise with the above mentioned method when the function f is highly nonlinear. These difficulties usually result in an ill-conditioned Hessian matrix, making the inversion process numerically unstable. Since our target function $f(\alpha) := f_{\mathcal{P}}(\alpha) = \langle A - X(\alpha), A - X(\alpha)\rangle_{\mathcal{P}}$ (with or without additional constraint functions $f_{\text{norm}}(\alpha)$ and $f_{\text{rep}}(\alpha)$, cf. Appendix A.5) is nonconvex and the Hessian is in general not positive definite, Newton's method cannot converge in general. In order to overcome this difficulty, in [7] a modified Newton's method is introduced which converges globally to a stationary point (it will not necessarily converge to a global optimum).

6.1 Finding a Descent Direction by Trust Region

In the following, we briefly repeat the procedure introduced in [7] for the approximation of tensors in the low rank format. For the modified Newton's method the trust

region subproblem is of vital importance. The trust region subproblem is

$$\begin{aligned} \min \quad & q(p) := f\left(\alpha^k\right) + \left\langle \nabla f\left(\alpha^k\right), p\right\rangle + \frac{1}{2}\left\langle H_f\left(\alpha^k\right)p, p\right\rangle \\ \text{s.t.} \quad & \|p\|_{T(\alpha^k)} \le r \end{aligned}$$

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 $H_f(\alpha^k)$ 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

$$\alpha^{k+1}(r) = \alpha^k - \left(\lambda T\left(\alpha^k\right) + H_f\left(\alpha^k\right)\right)^{-1} \nabla f\left(\alpha^k\right),$$

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

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

see [7] for more details. The substitution $\omega := 1/(1+\lambda)$ leads to

$$\begin{aligned} \alpha^{k+1}(\omega) &:= \alpha^k - \hat{H}_f\left(\alpha^k, \omega\right)^{-1} \nabla f\left(\alpha^k\right), \\ \hat{H}_f\left(\alpha^k, \omega\right) &:= \omega H_f\left(\alpha^k\right) + (1-\omega) T\left(\alpha^k\right), \end{aligned}$$

with a parameter $\omega \in [0, 1]$ that has to be determined adaptively.

The standard choice would be $T(\alpha^k) := \mathbf{Id}$. This leads to the gradient direction as $\omega \to 0$, but in practice we observe that the positive definite matrix

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

see Corollary 28, gives much better results. Thus we obtain a direction

$$d^k(\omega) := \left(\hat{H}_f\left(\alpha^k, \omega\right)\right)^{-1} \nabla f\left(\alpha^k\right),$$

which is for $\omega = 1$ the Newton direction and for $\omega \to 0$ a descent direction. When solving the system $\hat{H}_f(\alpha^k, \omega)$ iteratively by the cg-iteration, one can exploit the fact that the convergence depends on the positivity of $\hat{H}_f(\alpha^k, \omega)$. As long as the iteration diverges, we decrease ω. Also, when (9) is not fulfilled, we restart and decrease ω. Thus, during the iterative solve, the parameter ω can be determined.

Lemma 15 (Complexity of the minimization) *The total complexity for the minimization using c_N Newton steps is*

$$\mathcal{O}\left(c_N P\left(k^2 \sum_{\mu=1}^{d} k_\mu + k \sum_{\mu=1}^{d} n_\mu\right)\right).$$

Proof Combine Lemma 23 and Theorem 33 in the Appendix to find that the complexity for the computation of the gradient and matrix-vector multiplication with the Hessian is $\mathcal{O}(P(k^2 \sum_{\mu=1}^{d} k_\mu + k \sum_{\mu=1}^{d} n_\mu))$. □

Algorithm 4 Modified Newton's Method

1: Choose initial α^1 and parameters $\gamma, \beta \in (0,1)$, $\varepsilon \in \mathbb{R}_{>0}$, $\sigma \in (0, \frac{1}{2})$, $\delta \in \mathbb{R}_{>0}$, $p \in \mathbb{R}_{\geq 2}$, and define $k := 1$ and $\omega_0 := 1$.
2: **while** $\|\nabla f(\alpha^k)\| > \varepsilon$ **do**
3: $\omega_k := \min\{\frac{\omega_{k-1}}{\gamma}, 1\}$.
4: Compute d^k as a solution of

$$\hat{H}(\alpha^k, \omega_k) d^k = \nabla f(\alpha^k)$$

by the cg-method. If the cg-method fails to converge within a prescribed number of steps (e.g., 100) or the condition

$$\frac{\langle \nabla f(\alpha^k), d^k \rangle}{\|\nabla f(\alpha^k)\| \|d^k\|} \geq \min\{\delta, \|\nabla f(\alpha^k)\|^2\} \tag{9}$$

is false, we set $\omega_k := \gamma \omega_k$ and continue with step 4.
5: Compute $\bar{\omega}_k \in \mathbb{R}_{>0}$ by the Armijo rule

$$\bar{\omega}_k := \max_{l \in \mathbb{N}_{\geq 0}} \{\beta^l : f(\alpha^k) - f(\alpha^k - \beta^l d^k) \geq \sigma \beta^l \langle \nabla f(\alpha^k), d^k \rangle\}. \tag{10}$$

6: Set $\alpha^{k+1} := \alpha^k - \bar{\omega}_k d^k$ and $k \leftarrow k+1$.
7: **end while**

Remark 16 In Appendix A the efficient procedures for the computation of the gradient and evaluation of the Hessian of the given target function f are described in detail.

As mentioned in Remark 29, our preconditioner for the linear system is $(D^1)^{-1}$ which can be computed in $\mathcal{O}(dk'^3)$ (Remark 29). Because of the splitting of $H_f(\alpha^k)$ into a sum of matrices (cf. Definition 24), we can write

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

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

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

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

6.2 Line Search by Armijo Rule

The Armijo rule is characterized by (10). Let $d(\omega_k)$ be a suitable descent direction, i.e., a descent direction that is close to the Newton direction and fulfills (9). Our aim

is to perform a line search

$$\text{minimize} \quad \bar{\omega} \mapsto f(\alpha) - f\big(\alpha - \bar{\omega} d(\omega_k)\big)$$

which requires the evaluation of the one-dimensional function $\bar{\omega} \mapsto f(\alpha - \bar{\omega} d(\omega_k))$ for several $\bar{\omega} \in \mathbb{R}$. This can best be done by changing the representation of $X(\alpha - \bar{\omega} d(\omega_k))$ to the form (13) and then computing

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

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.

7 Numerical Examples

7.1 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, i.e., the constant $c_{\mathcal{P}}$ for the number $P = p_0 + c_{\mathcal{P}} k$ of pivots in Algorithm 1. The minimal reasonable choice is $c_{\mathcal{P}} = 1$, which means that we read $P(d(n-1)+1)$ 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 (8) of 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.

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 17 (Approximate Interpolation) *For any tensor A of order d, any $\varepsilon > 0$ and any $p > 0$ pivot indices $\mathcal{P} = \{I^1, \ldots, I^p\}$ there exists a tensor X of rank $k = p$ such that $\|A - X\|_{\mathcal{P}} \leq \varepsilon$, i.e., the full tensor can almost be interpolated on p crosses by a tensor of rank $k = p$.*

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

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}

Remark 18 The previous conjecture is not true for the exact interpolation of A, as the example (due to Aram Khachatryan) shows: a $2 \times 2 \times 2$ tensor of order $d = 3$ and rank $k = 3$ exists [15] and cannot be interpolated on two crosses by rank 2 because two nonintersecting crosses cover the whole index set.

There are two conclusions from our observation. First, it is not sufficient to read the data from $k(d(n-1)+1)$ indices in order to find a rank k tensor approximation. In particular, from k crosses one cannot 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 indices 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 $c_{\mathcal{P}}$ and P, we perform a numerical test. We approximate a tensor A of order $d = 4$ by a low rank tensor using $P = c_{\mathcal{P}} k$ pivots, $c_{\mathcal{P}} \in \{1, 2, 3, 4, 5, 10\}$. The tensor A is the evaluation of the function

$$a(x_1, \ldots, x_d) = \left(\sum_{\mu=1}^{d} x_\mu^2 \right)^{-1/2} \tag{11}$$

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

$$A_{i_1,\ldots,i_d} = a(1 + i_1/20, \ldots, 1 + i_d/20), \quad i_\mu = 0, \ldots, 20, \ \mu = 1, \ldots, d. \tag{12}$$

For $P = k$, i.e., $c_{\mathcal{P}} = 1$, the results in Fig. 3 are not convincing since the approximation accuracy stagnates. All other values yield a reasonably good approximation. The difference in the accuracies is not because of the number of crosses ($c_{\mathcal{P}} = 2$ gives almost the same results as $c_{\mathcal{P}} = 5$), but due to the fact that for each additional cross the minimization in Algorithm 1 is restarted with a slightly different target functional. Thereby, a local minimum from the previous step can be left for a better approximation and increases the probability of finding the global minimizer. We observed that a value of $c_{\mathcal{P}} = 5$ is a reasonably good choice that we will use for the rest of this article. For problems where the evaluation of A is expensive, one should use $c_{\mathcal{P}} = 2$. For problems where the evaluation of A is inexpensive, one should use $c_{\mathcal{P}} \geq 5$.

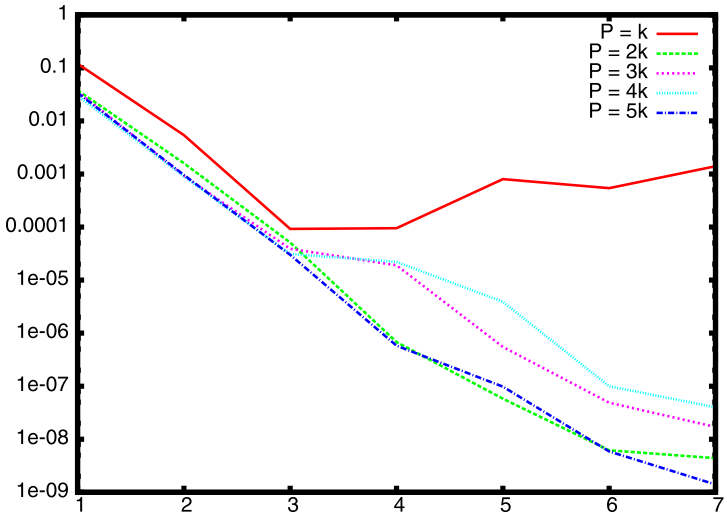


Fig. 3 Approximation error (in log-scale) for different numbers of pivot indices $P = c_{\mathcal{P}} k$ in dimension $d = 4$

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 $P = 5k$ pivots

k	$d=3$	$d=4$	$d=5$	$d=6$
1	9.6×10^{-2}	1.8×10^{-1}	2.8×10^{-1}	3.7×10^{-1}
2	2.7×10^{-3}	5.4×10^{-3}	1.0×10^{-2}	1.7×10^{-2}
3	2.4×10^{-5}	6.7×10^{-5}	1.6×10^{-4}	2.2×10^{-4}
4	2.3×10^{-13}	5.3×10^{-5}	1.2×10^{-12}	1.0×10^{-11}
5	–	(1.7×10^{-5})	–	–

7.2 Reconstruction of Low Rank Tensors

Now that we have obtained a suitable value for the number of pivot indices P, we can try to reconstruct the rank $k_A = 4$ tensor A from Sect. 5. The results for $d = 3, 4, 5, 6$ are contained in Table 2. In three of the cases the tensor is reconstructed, and in the case $d = 4$ we have computed a local minimum where the gradient of $f(\alpha)$ is almost zero and the Hessian $H(\alpha)$ is positive but $f(\alpha) \neq 0$. Depending on the initial guess and the pivots, this can happen as well for other dimensions.

Our conclusion is that local minima do exist and cannot be easily 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 for avoiding local minima is necessary and will be performed in a follow-up article.

7.3 Approximation of a Smooth Nonseparable Function

In this section we consider the d-variate function a from (11),

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

Table 3 The relative error $\frac{\|A-X(\alpha)\|}{\|A\|}$ and the target function value $f(\alpha)$ for the rank $k = 1, \ldots, 7$ approximation of an order $d = 3, 4, 5$ tensor

k	$d=3$		$d=4$		$d=5$	
	$\frac{\|A-X(\alpha)\|}{\|A\|}$	$f(\alpha)$	$\frac{\|A-X(\alpha)\|}{\|A\|}$	$f(\alpha)$	$\frac{\|A-X(\alpha)\|}{\|A\|}$	$f(\alpha)$
1	2.4×10^{-2}	6.7×10^{-2}	3.4×10^{-2}	7.3×10^{-2}	3.8×10^{-2}	1.3×10^{-1}
2	7.7×10^{-4}	7.3×10^{-5}	9.6×10^{-4}	1.1×10^{-4}	1.0×10^{-3}	1.2×10^{-4}
3	2.1×10^{-5}	7.1×10^{-8}	3.0×10^{-5}	7.8×10^{-8}	2.8×10^{-5}	1.0×10^{-7}
4	5.1×10^{-7}	4.5×10^{-11}	5.8×10^{-7}	6.0×10^{-11}	6.5×10^{-7}	6.1×10^{-11}
5	1.3×10^{-8}	3.5×10^{-14}	9.8×10^{-8}	2.3×10^{-12}	1.5×10^{-8}	4.3×10^{-14}
6	2.8×10^{-9}	2.3×10^{-15}	5.9×10^{-9}	5.6×10^{-15}	1.3×10^{-8}	3.3×10^{-14}
7	5.0×10^{-10}	7.5×10^{-17}	1.4×10^{-9}	4.0×10^{-16}	5.3×10^{-9}	3.9×10^{-15}

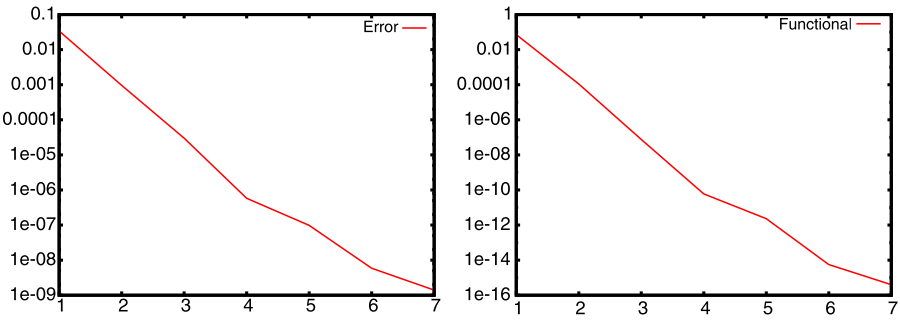


Fig. 4 The decay of the relative error (*left*) and the decay of the target function (*right*) for the rank $k = 1, \ldots, 7$ approximation of an order $d = 4$ tensor

and the corresponding tensor A of order d defined by

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

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 if either the gradient is close to zero or if after ten iterations there is almost no progress. The results of the test for dimension $d = 3, 4, 5$ are reported in Table 3, and the error decay is depicted in Fig. 4.

(The number of Newton steps required to find a local minimum is quite large, sometimes more than 3000, so there is definitely room for improvement.)

In the previous example the fiber-cross approximation works quite well. However, there are still some open problems that we want to address. For a more pronounced

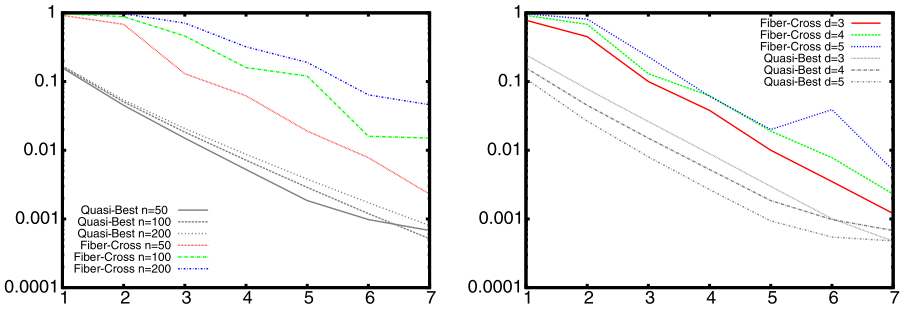


Fig. 5 The decay of the relative error for a rank $k = 1, \ldots, 7$ approximation for varying mesh-width (*left*) and varying dimension (*right*)

singularity we consider a similar tensor as before:

$$A_{i_1,\ldots,i_d} = a\left(\frac{1}{n} + i_1\frac{1}{n}, \ldots, \frac{1}{n} + i_d\frac{1}{n}\right), \quad i_\mu = 0, \ldots, n-1,\ \mu = 1, \ldots, d.$$

In the first experiment we vary the dimension $d = 3, 4, 5$ and keep $n = 50$ fixed. In the second experiment we keep the dimension $d = 4$ fixed and vary the mesh-width $n = 50, 100, 200$. The results in comparison with a quasi-best rank k approximation are depicted in Fig. 5. We observe that the change of dimension has only a mild effect on the accuracy of the fiber-cross approximation, but the change of the mesh-width has much more severe consequences: the convergence (with respect to the rank k) is delayed. Here, it will be necessary to consider (algebraic) multilevel techniques as in [10] in order to be independent of the mesh-width.

7.4 Approximation of a Tensor from Quantum Chemistry

Finally, we consider an order $d = 3$ tensor describing the electron density of the CH_4 molecule. The data were kindly provided by Sambasiva Rao Chinnamsetty.

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 if either the gradient is close to zero or if after ten iterations there is almost no progress. We compare the approximation with the best-known low tensor-rank approximation of the full tensor (obtained by a trust region Newton method).

In Fig. 6 we report the results for the (only theoretically interesting) full search pivot strategy and the partial search pivot strategy of Algorithm 3, which uses only the information on the previous $p - 1$ crosses. The results are compared to a quasi-best approximation obtained from a low rank approximation of the full tensor.

We observe that the (simple) partial pivoting fails: the approximation error stagnates from rank 3 on. This effect is known for the matrix case [3] and can be circumvented by either a priori knowledge (positions of atoms) or by an additional random

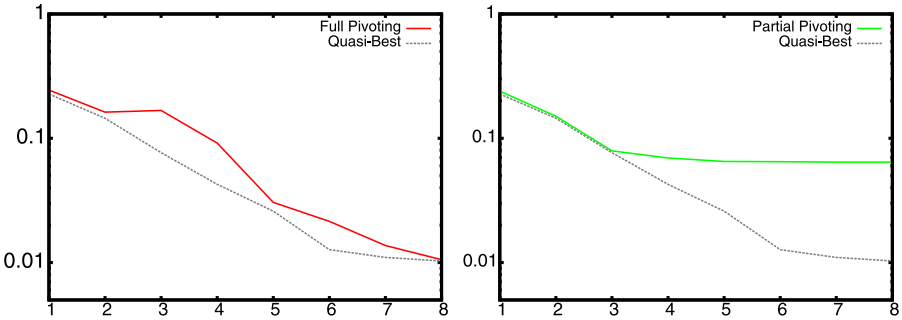
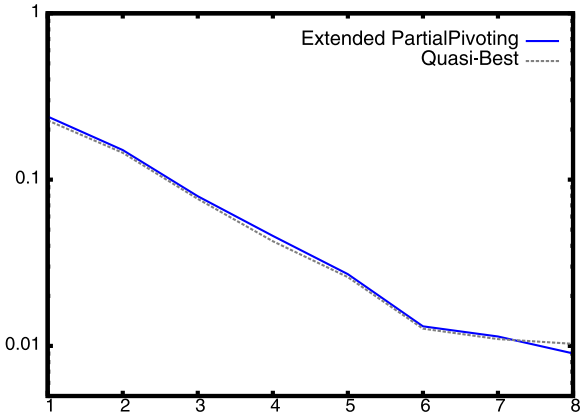


Fig. 6 The relative error in log-scale for the rank $k = 1, \ldots, 10$ approximation of an order $d = 3$ tensor. *Left*: full pivoting, *Right*: (simple) partial pivoting

Fig. 7 The relative error in log-scale for the rank $k = 1, \ldots, 10$ approximation of an order $d = 3$ tensor using extended partial pivoting



sampling (Algorithm 2) as follows. After finding a first preliminary pivot index by Algorithm 3, we also compute a second preliminary pivot index by Algorithm 2. The better of the two (with respect to $|A_I - X_I|$) is used as the next pivot index. The results of this extended partial pivoting are presented in Fig. 7.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

Appendix A: Gradient and Hessian of f

A.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 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 $\alpha_{i,j,\mu}$.

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

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

Also let $\nu^* \in \{1, \dots, d\} \setminus \{\mu^*\}$ and $m^* \in \{1, \dots, k_{\nu^*}\}$ be given, and without loss of generality $\mu^* < \nu^*$. 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 defining $Z_{i^*}^{(\mu^*, j^*, \nu^*, m^*)}$ for the case $\nu^* = \mu^*$ because later this term will 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 19 *The first partial derivatives of*

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

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

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

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

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

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)

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

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

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

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

Analogously, we obtain the second partial derivatives. □

A.2 Changing the Representation

Sometimes we require the values of $X(\alpha)$ at all indices $I \in J^{p,\mu}$, $p \in \{1,\dots,P\}$, $\mu \in \{1,\dots,d\}$. The computation of these values can be done in two steps:

1. First we compute coefficients $\tilde{\alpha}_{p,j,\mu}$ such that

$$X(\alpha)|_{J^{p,\mu}} = \sum_{j=1}^{k_\mu} \tilde{\alpha}_{p,j,\mu} v_{j,\mu}. \tag{13}$$

2. Second we compute the linear combinations in (13).

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

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

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

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

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 $\mathcal{O}(Pkd)$.

A.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 20 *Let $p \in \{1,\dots,P\}$ and $\mu \in \{1,\dots,d\}$. For any index $I \in J^{p,\mu}$ there holds*:

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

Proof The proof follows directly from the definition of $Y_{i^*}^{(\mu^*,j^*)}$. □

As a consequence of Lemma 20, we can precompute the values

$$\gamma(p,i^*,\mu) := \prod_{\nu\neq\mu} (x_{i^*,\nu})_{I_\nu^p} \tag{14}$$

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 $\gamma(p,i^*,\cdot)\equiv 0$. If one of the factors is zero, then the corresponding nonzero entries of $\gamma(p,i^*,\cdot)$ (at most Pk) can be computed in $\mathcal{O}(Pdk)$. If all factors are nonzero, then there holds:

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

Remark 21 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 5.

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

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

Lemma 22 (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

Algorithm 5 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

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

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

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 $\gamma(p,i,\mu)$. □

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 $\mathcal{O}(P\sum_{\nu=1}^{d} k_\nu n_\nu + Pk\sum_{\nu=1}^{d} n_\nu)$.

Algorithm 6 Computation of $R_I^{(p,\mu)}$

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

Lemma 23 (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*

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

Proof 1. (Setup) For the setup, we compute and store the entries of $\gamma(p,i^*,\mu)$ according to Remark 21 in $\mathcal{O}(Pkd)$. The defects $R^{(p,\mu)}$ are computable in $\mathcal{O}(Pk\sum_{\nu=1}^d n_\nu)$ (Lemma 22) 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 $N_{\mathrm{St},1}$ and complexity $N_{\mathrm{co},1}$ of the setup phase are

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

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:

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

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

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

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

$$\left\langle R^{(p,\mu)}, Y_{i^*}^{(\mu^*,j^*)}\right\rangle_{p,\mu} = \left(\prod_{\substack{\nu=1\\ \nu\neq\mu^*,\mu}}^d (x_{i^*,\nu})_{I_\nu^p}\right)(v_{j^*,\mu^*})_{I_{\mu^*}^p} t(p,\mu,i^*). \tag{15}$$

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

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

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

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 (15) is zero. The sought nonzero 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

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

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 $\mathcal{O}(Pk\sum_{\nu=1}^{d} k_\nu)$,

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

All parts together require $N_{\mathrm{St}}(\nabla) = \sum_{\iota=1}^{3} N_{\mathrm{St},\iota}$ storage and $N_{\mathrm{co}}(\nabla) = \sum_{\iota=1}^{3} N_{\mathrm{co},\iota}$ 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 7.

Algorithm 7 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., $\delta = 10^{-16}$)}
Initialize $\mathrm{grad}(i^*, j^*, \mu^*) := 0$
for $p = 1, \ldots, P$ **do**
  for $i^* = 1, \ldots, k$ **do**
    Determine the two smallest factors $\theta_1 := |(X_{i^*,q})_{I_q^p}| \le \theta_2 := |(X_{i^*,a})_{I_a^p}|$
    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 $\theta_2 \le \delta$ **then**
      $y := \prod_{\mu \ne q,a} (X_{i^*,\mu})_{I_\mu^p}$
      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 $\theta_1 \le \delta$ **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

A.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 as follows. 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

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

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

$$D_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} = \sum_{p=1}^{P} \sum_{\mu=1}^{d} \big\langle Y_{\ell^*}^{(\nu^*,m^*)},\ Y_{i^*}^{(\mu^*,j^*)} \big\rangle_{p,\mu}. \tag{16}$$

Definition 24 (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

$$D^1_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} := \sum_{p=1}^{P} \delta_{j^*,m^*} \delta_{\mu^*,\nu^*} \gamma(p, i^*, \mu^*) \gamma(p, \ell^*, \nu^*), \tag{17}$$

$$\begin{aligned} D^2_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} := \sum_{p=1}^{P} \bar{\delta}_{\mu^*,\nu^*} \big(& \gamma(p, i^*, \mu^*) \zeta(p, \ell^*, \mu^*, \nu^*) (v_{m^*,\nu^*})_{I_{\nu^*}^p} \alpha_{\ell^*,j^*,\mu^*} \\ & + \zeta(p, i^*, \mu^*, \nu^*) \gamma(p, \ell^*, \nu^*) (v_{j^*,\mu^*})_{I_{\mu^*}^p} \alpha_{i^*,m^*,\nu^*} \big), \end{aligned} \tag{18}$$

$$\begin{aligned} D^3_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} := & \sum_{p=1}^{P} \sum_{\substack{\mu=1 \\ \mu \neq \mu^*,\nu^*}}^{d} \zeta(p, i^*, \mu^*, \mu) \zeta(p, \ell^*, \nu^*, \mu) \\ & \times (v_{j^*,\mu^*})_{I_{\mu^*}^p} (v_{m^*,\nu^*})_{I_{\nu^*}^p} \langle x_{i^*,\mu}, x_{\ell^*,\mu} \rangle. \end{aligned} \tag{19}$$

Lemma 25 *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) $\mu = \mu^* = \nu^*$
- (D^2) $\mu = \mu^* \neq \nu^*$ or $\mu = \nu^* \neq \mu^*$
- (D^3) $\mu \neq \mu^*$ and $\mu \neq \nu^*$.

According to the definition of Y and Lemma 20, 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

$$\hat{D}^{1,\mu} = \left(\sum_{p=1}^{P} \gamma_{p,\mu} (\gamma_{p,\mu})^T \right) \otimes \mathbf{Id}_{k_\mu \times k_\mu}, \tag{20}$$

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

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

see (14). 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

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

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

$$X^\mu := \left(\bigotimes_{\nu \neq \mu} x_{1\mu}, \ldots, \bigotimes_{\nu \neq \mu} x_{k\mu} \right) \in \mathbb{R}^{K_\mu \times k}, \quad K_\mu := \prod_{\nu \neq \mu} k_\mu,$$

$$E_P^\mu := \bigotimes_{\nu \neq \mu} e_{I_\nu^p} \in \mathbb{R}^{K_\mu},$$

$$\mathrm{P}^{\mu} := \sum_{p=1}^{P} \bigotimes_{\nu\neq\mu} e_{I_\nu^p} \big(e_{I_\nu^p}\big)^t \in \mathbb{R}^{K_\mu \times K_\mu}, \qquad G^{\mu} := \big(X^{\mu}\big)^t P^{\mu} X^{\mu} \in \mathbb{R}^{k\times k}.$$

Remark 26 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 fibers in direction μ exactly at the same position.

Lemma 27 *Let $X \in \mathcal{T}(d,k)$ with tensor rank exactly k. Then we have for all $\mu \in \{1,\dots,d\}$ that the matrix X^{μ} has rank k.*

Proof Assume there exists $\mu \in \{1,\dots,d\}$ with $\{\bigotimes_{\nu\neq\mu} x_{1\mu},\dots,\bigotimes_{\nu\neq\mu} x_{k\mu}\}$ linear dependent. Then there are $\lambda_1,\dots,\lambda_k \in \mathbb{R}$ and $i_0 \in \{1,\dots,k\}$ 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 $\operatorname{rank}(X) = k$. □

Corollary 28 *Let $X \in \mathcal{T}(d,k)$ with tensor rank exactly k and the pivot indices constructed as mentioned in Remark 26. Then D^1 is positive definite, hence regular. Furthermore, we have*

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

Proof Since $\hat{D}^{1,\mu} = G^{\mu} \otimes \mathbf{Id}_{k_\mu \times k_\mu}$ is a sum of positive semidefinite matrices, see (20), it is positive semidefinite. It follows from Lemma 27 and Remark 26 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

$$D^1\big(D^1\big)^{-1} = \sum_{\mu=1}^{d} \sum_{\mu'=1}^{d} \mathbb{E}^{\mu}\mathbb{E}^{\mu'} \otimes G^{\mu}\big(G^{\mu'}\big)^{-1} \otimes \mathbf{Id}_{k_\mu \times k_\mu}$$

$$= \sum_{\mu=1}^{d} \mathbb{E}^{\mu} \otimes \mathbf{Id}_{k\times k} \otimes \mathbf{Id}_{k_\mu \times k_\mu} = \mathbf{Id}. \qquad \square$$

Remark 29 (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 Sect. 6.

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 30 (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 (18))

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

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

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

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 represen-

tation 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 $(x_{\ell^*,q})_{I_q^p} = 0$. Then the term $\zeta(p, \ell^*, \mu^*, \nu^*)$ is nonzero only for $\mu^* = q$ or $\nu^* = q$. Consequently,

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

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 $(x_{\ell^*,\iota})_{I_\iota^p} = 0$ for $\iota \in \{q, a\}$. This yields nonzero entries only for the case $\mu^*, \nu^* \in \{a, q\}$. For these $\mathcal{O}(1)$ combinations, the sub-matrix is of rank one because the variables j^* and m^* factorize. □

Lemma 31 (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

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

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

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

Two cases have to be distinguished:

- there exists $a \neq q \in \{1, \ldots, d\}$ such that either $(x_{i^*,\mu})_{I_\mu^p} = 0$ for $\mu = a, q$ or $(x_{\ell^*,\mu})_{I_\mu^p} = 0$ for $\mu = a, q$;

- there exist $a, q \in \{1, \dots, d\}$ such that $(x_{i^*,\mu})_{I_\mu^p} \neq 0$ for $\mu \neq q$ and $(x_{\ell^*,\mu})_{I_\mu^p} \neq 0$ for $\mu \neq a$;

Case 1 There exists $a \neq q \in \{1, \dots, d\}$ such that $(X_{i^*,\mu})_{I_\mu^p} = 0$ 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

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

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

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

$$
\begin{aligned}
\zeta(p, i^*, \mu^*, \mu) &= \zeta_1(p, i^*, \mu^*)\zeta_2(p, i^*, \mu), \\
\zeta(p, \ell^*, \nu^*, \mu) &= \zeta_3(p, \ell^*, \nu^*)\zeta_4(p, \ell^*, \mu),
\end{aligned}
$$

because at most one of the factors $(x_{i^*,\mu})_{I_\mu^p}$ (respectively $(x_{\ell^*,\mu})_{I_\mu^p}$) 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

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

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

$$
\sum_{\substack{\mu=1 \\ \mu\neq\mu^*,\nu^*}}^{d} \zeta_2(p, i^*, \mu)\zeta_4(p, \ell^*, \mu)\langle x_{i^*,\mu}, x_{\ell^*,\mu}\rangle
$$

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

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 representation

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

The first two terms lead to a rank 2 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. Finally, we consider the second part of the Hessian, the matrix C with entries

$$
C_{(i^*,j^*,\mu^*),(\ell^*,m^*,\nu^*)} = \sum_{p=1}^{P}\sum_{\mu=1}^{d} \delta_{i^*,\ell^*}\bar{\delta}_{\mu^*,\nu^*}\big\langle X - A,\ Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)}\big\rangle_{p,\mu}. \tag{21}
$$

According to the definition of Z, we obtain

$$\left\langle X - A,\ Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} \right\rangle_{p,\mu}$$

$$= \begin{cases} \prod_{\substack{\nu=1 \\ \nu\neq\mu^*,\nu^*}}^{d} (x_{i^*,\nu})_{I_\nu^p} (v_{m^*,\nu^*})_{I_{\nu^*}^p} s(p,\mu,j^*), & \mu = \mu^*, \\ \prod_{\substack{\nu=1 \\ \nu\neq\mu^*,\nu^*}}^{d} (x_{i^*,\nu})_{I_\nu^p} (v_{j^*,\mu^*})_{I_{\mu^*}^p} s(p,\mu,m^*), & \mu = \nu^*, \\ \prod_{\substack{\nu=1 \\ \nu\neq\mu,\mu^*,\nu^*}}^{d} (x_{i^*,\nu})_{I_\nu^p} (v_{m^*,\nu^*})_{I_{\nu^*}^p} (v_{j^*,\mu^*})_{I_{\mu^*}^p} t(p,\mu,i^*), & \text{otherwise.} \end{cases}$$

Lemma 32 (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 nonzero entries, and for each of these combinations the summation over μ consists of at most one nonzero 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 one. 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 nonzero 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 $i^* = \ell^*$, $p = 1,\dots,P$) into three parts:

$$C^{1,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} := \bar{\delta}_{\mu^*,\nu^*} \left\langle X - A,\ Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} \right\rangle_{p,\mu^*},$$

$$C^{2,i^*,p}_{(j^*,\mu^*),(m^*,\nu^*)} := \bar{\delta}_{\mu^*,\nu^*} \left\langle X - A,\ Z_{i^*}^{(\mu^*,j^*,\nu^*,m^*)} \right\rangle_{p,\nu^*},$$

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

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

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

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 $\mathcal{O}(Pk\sum_{\nu=1}^{d} k_\mu)$. We derive the complexity for $C^{2,i^*,p}$ in the same way. Finally, we consider the matrices $C^{3,i^*,p}$:

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

For their efficient representation we define

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

The three terms ψ_1, ψ_3, and 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

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

$$- \delta_{\mu^*,\nu^*}\psi(p,i^*,\nu^*,\mu^*)\zeta_2(p,i^*,\mu^*)\zeta_3(p,i^*,\nu^*)(v_{m^*,\nu^*})_{I_{\nu^*}^p}(v_{j^*,\mu^*})_{I_{\mu^*}^p}.$$

The first term separates the variables and leads to a rank 2 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 33 *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 29 and Lemmata 30, 31, 32. □

A.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

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

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

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

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

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

In Lemma 4, 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 nonuniqueness can be remedied by an additional penalty term

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

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

Lemma 35 *The first partial derivative of f_{rep} with respect to the variable α_{i^*,j^*,μ^*} is*

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

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

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

Proof By 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

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

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

References

1. Beylkin, G., Mohlenkamp, M.J.: Algorithms for numerical analysis in high dimensions. SIAM J. Sci. Comput. **26**(6), 2133–2159 (2005)
2. Beylkin, G., Garcke, J., Mohlenkamp, M.: Multivariate regression and machine learning with sums of separable functions. SIAM J. Sci. Comput. **31**, 1840–1857 (2009)
3. Börm, S., Grasedyck, L.: Hybrid cross approximation of integral operators. Numer. Math. **101**, 221–249 (2005)
4. Chinnamsetty, S.R., Espig, M., Khoromskij, B.N., Hackbusch, W., Flad, H.J.: Tensor product approximation with optimal rank in quantum chemistry. J. Chem. Phys. **127**(8) (2007)
5. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
6. de Silva, V., Lim, L.-H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. Technical Report SCCM-06-06, Stanford University (2006)
7. Espig, M.: Effiziente Bestapproximation mittels Summen von Elementartensoren in hohen Dimensionen. PhD thesis, Universität Leipzig (2008)
8. Flad, H.-J., Khoromskij, B.N., Savostyanov, D., Tyrtyshnikov, E.E.: Verification of the cross 3D algorithm on quantum chemistry data. Russ. J. Numer. Anal. Math. Model. **23**, 329–344 (2008)
9. Grasedyck, L.: Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure. Computing **72**, 247–265 (2004)
10. Khoromskij, B.N., Khoromskaia, V.: Multigrid accelerated approximation of function related multidimensional arrays. Preprint 40/2008, Max Planck Institute for Mathematics in the Sciences (2008)

11. Kolda, T.G.: Orthogonal tensor decompositions. SIAM J. Matrix Anal. Appl. **23**(1), 243–255 (2001)
12. Oseledets, I.V., Savost'yanov, D.V.: Minimization methods for approximating tensors and their comparison. Comput. Math. Math. Phys. **46**(10), 1641–1650 (2006)
13. Oseledets, I.V., Tyrtyshnikov, E.E., Savost'yanov, D.V.: Tucker dimensionality reduction of three-dimensional arrays in linear time. SIAM J. Matrix Anal. Appl. (2008, to appear)
14. Paatero, P.: A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. Chemometrics Intel. Lab. Syst. **38**, 223–242 (1997)
15. Ten Berge, J.M.F.: Kruskal's polynomial for $2 \times 2 \times 2$ arrays and a generalization to $2 \times n \times n$. Psychometrika **56**, 631–636 (1991)
16. Young, F.W., de Leeuw, J., Takane, Y.: Additive structure in qualitative data: an alternating least squares method with optimal scaling features. Psychometrika **41**, 471–503 (1976)
17. Zhang, T., Golub, G.H.: Rank-one approximation to high order tensors. SIAM J. Matrix Anal. Appl. **23**(2), 534–550 (2001)