Numer. Math. (2012) 122:489–525
DOI 10.1007/s00211-012-0465-9

Numerische
Mathematik

A regularized Newton method for the efficient approximation of tensors represented in the canonical tensor format

Mike Espig · Wolfgang Hackbusch

Received: 2 September 2011 / Revised: 25 January 2012 / Published online: 13 May 2012

Abstract In the present survey, we consider a rank approximation algorithm for tensors represented in the canonical format in arbitrary pre-Hilbert tensor product spaces. It is shown that the original approximation problem is equivalent to a finite dimensional ℓ_2 minimization problem. The ℓ_2 minimization problem is solved by a regularized Newton method which requires the computation and evaluation of the first and second derivative of the objective function. A systematic choice of the initial guess for the iterative scheme is introduced. The effectiveness of the approach is demonstrated in numerical experiments.

Mathematics Subject Classification 15A69 · 90C06 · 65F10 · 65N22

1 Introduction

Let $d \geq 3$ and $\mathcal{T} := \bigotimes_{\mu=1}^{d} V_\mu$ be the algebraic tensor space constructed from arbitrary pre-Hilbert spaces V_μ. In algebraic tensor spaces every tensor can be written as a finite sum of elementary tensors, where an elementary tensor is of the form $w = \bigotimes_{\mu=1}^{d} w_\mu$, $w_\mu \in V_\mu$. Since by definition of $\mathcal{T}$ only linear combinations of finitely many terms are allowed, we have

$$\mathcal{T} = \operatorname{span}\{v \in \mathcal{T}_r : r \in \mathbb{N}\},$$

where $\mathcal{T}_r$ is the set of tensors which can be represented as a sum of r elementary tensors, see Definition 2.2. In this article, we consider the canonical tensor format with representation rank r, where for variable r the format is explained by the

M. Espig (✉) · W. Hackbusch
Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany
e-mail: mike.espig@alopax.de

mapping

$$\mathfrak{C} : \bigtimes_{\mu=1}^{d} V_\mu^r \to \mathcal{T}_r$$

$$\hat{v} := (v_{i\mu} : 1 \leq i \leq r,\ 1 \leq \mu \leq d) \mapsto \mathfrak{C}(\hat{v}) := \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu}.$$

Note that the representation rank refers to the representation system $(v_{i\mu} : 1 \leq i \leq r, 1 \leq \mu \leq d)$, not to the represented tensor. The following two approximation problems in the canonical tensor format are discussed in this paper. In the first problem, for a given $v \in \mathcal{T}_R$ we fix the representation rank $r < R$ and look for approximations in $\mathcal{T}_r$, i.e. we want to determine a representation system $\hat{x}^* \in \bigtimes_{\mu=1}^{d} V_\mu^r$ such that

$$\varepsilon(r) := \varepsilon(\mathfrak{C}(\hat{x}^*), r) := \|v - \mathfrak{C}(\hat{x}^*)\| = \text{dist}\ (v, \mathcal{T}_r) = \inf_{\hat{x} \in \bigtimes_{\mu=1}^{d} V_\mu^r} \|v - \mathfrak{C}(\hat{x})\|.$$

In the second problem the roles of r and $\varepsilon(r)$ are reversed. For given $v \in \mathcal{T}_R$ and $\varepsilon > 0$ we are looking for minimal $r_\varepsilon \leq R$ and $\hat{x}^* \in \bigtimes_{\mu=1}^{d} V_\mu^{r_\varepsilon}$ such that:

(i) $\|v - \mathfrak{C}(\hat{x}^*)\| \leq \varepsilon \|v\|$,
(ii) $\|v - \mathfrak{C}(\hat{x}^*)\| = \text{dist}\ \big(v, \mathcal{T}_{r_\varepsilon}\big)$.

We briefly summarize existing approaches for solving the first approximation problem in finite dimensional tensor spaces. One of the most popular minimization method for solving the lower rank approximation problem with fixed representation rank is the alternating least squares (ALS) algorithm. In [11], the ALS method was applied for principle component analysis of order three tensors and in [1,2] for tensors presented in the canonical format. Furthermore, the minimization problem was also solved by a Gauss-Newton method in [15,16] and by Newton method in [14].

2 Pre-Hilbert tensor product spaces and the canonical tensor format

In the present paper, we are focusing on the algebraic tensor space $\mathcal{T} := \bigotimes_{\mu=1}^{d} V_\mu$ of arbitrary real-valued pre-Hilbert spaces $(V_1, \langle \cdot, \cdot \rangle_1), \ldots, (V_d, \langle \cdot, \cdot \rangle_d)$ for $d \geq 3$.

Definition 2.1 (*Elementary Tensor, Representation System*) We call a tensor $v \in \mathcal{T}$ an *elementary tensor* if there exits $(v_\mu \in V_\mu : 1 \leq \mu \leq d)$ such that

$$v = \bigotimes_{\mu=1}^{d} v_\mu. \tag{1}$$

Further, we call the d-tuple of vectors $(v_1, \ldots, v_d)$ from Eq. (1) a *representation system* of v.

Due to the multilinearity of the mapping $\otimes$, a representation system of a given elementary tensor is not uniquely determined.

The following notations and definitions will be useful. Since $(V_1, \langle\cdot,\cdot\rangle_1), \dots, (V_d, \langle\cdot,\cdot\rangle_d)$ are pre-Hilbert spaces, we equip the tensor product space $\mathcal{T}$ with the induced scalar product, i.e. the induced scalar product of $\mathcal{T}$ is defined on elementary tensors $v = \bigotimes_{\mu=1}^{d} v_\mu$ and $w = \bigotimes_{\mu=1}^{d} w_\mu$ by

$$\langle v, w\rangle := \prod_{\mu=1}^{d} \left\langle v_\mu, w_\mu \right\rangle_\mu .$$

This definition introduces a bilinear form and has a unique extension $\langle\cdot,\cdot\rangle : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$. It is known from the text books that this bilinear form is a scalar product in $\mathcal{T}$. Further, we equip $\mathcal{T}$ with the norm $\|\cdot\|$ associated with the induced scalar product.

We recall that $L(V, W)$ is the space of linear maps from V to W. For given $A_\mu \in L(V_\mu, W_\mu)$ we define the elementary tensor

$$A := \bigotimes_{\mu=1}^{d} A_\mu : \bigotimes_{\mu=1}^{d} V_\mu \to \bigotimes_{\mu=1}^{d} W_\mu$$

by

$$A\left(\bigotimes_{\mu=1}^{d} v_\mu\right) := \bigotimes_{\mu=1}^{d} A_\mu v_\mu . \tag{2}$$

The mapping defined by (2) extends uniquely to a linear mapping $A : \bigotimes_{\mu=1}^{d} V_\mu \to \bigotimes_{\mu=1}^{d} W_\mu$. In tensor product spaces we embed the vector spaces $\bigotimes_{\mu=1}^{d} L(V_\mu, W_\mu)$ into $L\left(\bigotimes_{\mu=1}^{d} V_\mu, \bigotimes_{\mu=1}^{d} W_\mu\right)$, where the embedding is described by (2), i.e Eq. (2) introduces a unique defined injective linear mapping $T : \bigotimes_{\mu=1}^{d} L(V_\mu, W_\mu) \to L\left(\bigotimes_{\mu=1}^{d} V_\mu, \bigotimes_{\mu=1}^{d} W_\mu\right)$. This mapping T is called the canonical homomorphism, see [8] for more details. Obviously, T is bijective if $\dim V_\mu < \infty$ and $\dim W_\mu < \infty$ since we have $\dim \bigotimes_{\mu=1}^{d} L(V_\mu, W_\mu) = \dim L\left(\bigotimes_{\mu=1}^{d} V_\mu, \bigotimes_{\mu=1}^{d} W_\mu\right)$.

By definition of the tensor space, every tensor $v \in \mathcal{T}$ can be written as a sum of elementary tensors. Hence the question of the minimal number of producing elementary tensors arises. This question leads us to the definition of the tensor rank of v.

Definition 2.2 (*r-Terms, Tensor Rank, Canonical Tensor Format, Representation System*) The set $\mathcal{T}_r$ of tensors which can be represented in $\mathcal{T}$ with *r-terms* is defined as

$$\mathcal{T}_r := \left\{ \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{T} : v_{i\mu} \in V_\mu \right\} . \tag{3}$$

Let $v \in \mathcal{T}$. The *tensor rank of* v *in* $\mathcal{T}$ is

$$\operatorname{rank}_{\mathcal{T}}(v) := \min\left\{ r \in \mathbb{N}_0 : v \in \mathcal{T}_r \right\} . \tag{4}$$

The *canonical tensor format in $\mathcal{T}$* for variable r is defined by the mapping

$$\mathfrak{C} : \bigtimes_{\mu=1}^{d} V_\mu^r \to \mathcal{T}_r, \tag{5}$$

$$\hat{v} := (v_{i\mu} : 1 \leq i \leq r,\ 1 \leq \mu \leq d) \mapsto \mathfrak{C}(\hat{v}) := \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu}.$$

We call the sum of elementary tensors $v = \sum_{i=1}^{r} \otimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{T}_r$ a tensor represented in the canonical tensor format with r terms. The system of vectors $\left(v_{i\mu} : 1 \leq i \leq r,\ 1 \leq \mu \leq d\right)$ is a *representation system* of v with *representation rank* r.

Note that the symbol of the set of tensors with r-terms $\mathcal{T}_r$ is combined with the symbol which we use for the tensor product space $\mathcal{T}$. Later, if there is a tensor product space $\mathcal{S}$ defined, then the symbol $\mathcal{S}_r$ identifies the set of tensors with r-terms in $\mathcal{S}$.

Similarly to the case of elementary tensors, a representation system of a sum of elementary tensors is not uniquely determined.

Remark 2.3 Note that the definition of the tensor rank of v is associated with the tensor space $\mathcal{T}$. This classification is imported since the tensor rank of v depends on the considered tensor space. For instance, let $\{a, b\} \subset \mathbb{R}^n$ be linearly independent and define the tensor

$$v := a \otimes b \otimes b + b \otimes a \otimes b + b \otimes b \otimes a = a \otimes (b \otimes b) + b \otimes (a \otimes b + b \otimes a).$$

We have for the two isomorphic tensor spaces $\mathcal{V} := \mathbb{R}^n \otimes \mathbb{R}^n \otimes \mathbb{R}^n$ and $\mathcal{W} := \mathbb{R}^n \otimes (\mathbb{R}^n \otimes \mathbb{R}^n)$

$$\mathrm{rank}_{\mathcal{V}}(v) = 3 \neq 2 = \mathrm{rank}_{\mathcal{W}}(v).$$

In this article, the tensor product of finite dimensional subspaces is of importance. Let U_μ be a subspace of V_μ with finite dimension $t_\mu := \dim U_\mu$ and $\mathcal{B}_\mu := (u_{l\mu} : 1 \leq l \leq t_\mu)$ be a basis of U_μ. Every tensor $v \in \mathcal{U} := \bigotimes_{\mu=1}^{d} U_\mu$ has a representation

$$v = \sum_{l_1=1}^{t_1} \cdots \sum_{l_d=1}^{t_d} a_{(l_1,\ldots,l_d)} \bigotimes_{\mu=1}^{d} u_{l_\mu \mu} \tag{6}$$

with uniquely defined coefficients $a_{(l_1,\ldots,l_d)}$. In the following let

$$\mathcal{S} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{t_\mu}. \tag{7}$$

Definition 2.4 (*Coefficient Tensor*) The coefficients $a_{(l_1,\ldots,l_d)}$ from Eq. (6) create a tensor $a \in \mathcal{S}$. We call $a \in \mathcal{S}$ the *coefficient tensor* of v with respect to the basis $\mathcal{B} := (\otimes_{\mu=1}^{d} u_{l_\mu \mu} : (l_1, \ldots, l_d) \in \times_{\mu=1}^{d} \{l_\mu \in \mathbb{N} : 1 \leq l_\mu \leq t_\mu\})$. If it is clear from the context which basis $\mathcal{B}$ is considered, we simply say a is the *coefficient tensor* of v.

To store the coefficient tensor a of v, one needs $\prod_{\mu=1}^{d} t_\mu$ memory entries, i.e. the memory requirement grows exponentially with d. But for $v \in \mathcal{T}_r$ one needs only $r \cdot \sum_{\mu=1}^{d} t_\mu$ entries, see Lemma 2.5.

Lemma 2.5 *Let $v \in \bigotimes_{\mu=1}^{d} U_\mu$ and $a \in \mathcal{S}$ the coefficient tensor of v. We have* $\operatorname{rank}_{\mathcal{S}}(a) = \operatorname{rank}_{\mathcal{T}}(v)$.

Proof Let $v := \sum_{i=1}^{r} \otimes_{\mu=1}^{d} v_{i\mu} \in \bigotimes_{\mu=1}^{d} U_\mu$ with $\operatorname{rank}_{\mathcal{T}}(v) = r$ and $\mathcal{B}_\mu$ as described above. Since $t_\mu = \dim U_\mu$, there is the canonical isomorphism $\Phi_\mu : \mathbb{R}^{t_\mu} \to U_\mu$ introduced by the canonical basis of $\mathbb{R}^{t_\mu}$ and $\mathcal{B}_\mu$. Furthermore, there exist $a_{i\mu} \in \mathbb{R}^{t_\mu}$ such that $v_{i\mu} = \Phi_\mu a_{i\mu}$. We have

$$v = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu} = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} \Phi_\mu a_{i\mu} = \left(\bigotimes_{\mu=1}^{d} \Phi_\mu \right) \left(\sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} a_{i\mu} \right) = \Phi\, a,$$

where $\Phi := \bigotimes_{\mu=1}^{d} \Phi_\mu$. Consequently, we have $\operatorname{rank}_{\mathcal{S}}(a) \leq \operatorname{rank}_{\mathcal{T}}(v)$ and $\operatorname{rank}_{\mathcal{T}}(v) \leq \operatorname{rank}_{\mathcal{S}}(a)$ since $a = \Phi^{-1} v$ and the tensor rank of Φ is one. □

3 Characteristics of best approximations

Let $\langle \cdot, \cdot \rangle$ be the induced inner product of $\mathcal{T} = \bigotimes_{\mu=1}^{d} V_\mu$ and $\| \cdot \| := \sqrt{\langle \cdot, \cdot \rangle}$.

Lemma 3.1 *Let $U_\mu \subseteq V_\mu$ be a linear subspace of V_μ, $P_\mu : V_\mu \to U_\mu$ the orthogonal projection of V_μ onto U_μ and define $\mathcal{U} := \bigotimes_{\mu=1}^{d} U_\mu$. Then $P := \bigotimes_{\mu=1}^{d} P_\mu : \mathcal{T} \to \mathcal{U}$ is the orthogonal projection of $\mathcal{T}$ onto $\mathcal{U}$. If $U_\mu \neq \{0\}$, the tensor rank of P is exactly one.*

Proof Obviously, we have $P^2 = P$ and $P = P^t$. □

Corollary 3.2 *Let $v = \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{T}_r$ and $U_\mu \subseteq V_\mu$ be a linear subspace of V_μ such that $\{v_{i\mu} : 1 \leq i \leq r\} \subset U_\mu$. Furthermore, let $P_\mu : V_\mu \to U_\mu$ be the orthogonal projection of V_μ onto U_μ and $P := \bigotimes_{\mu=1}^{d} P_\mu$. Then*

$$\|v - Px\| \leq \|v - x\| \tag{8}$$

holds for all $x \in \mathcal{T}$. Moreover, if additionally $x \in \mathcal{T} \backslash \mathcal{U}$, then the inequality

$$\|v - Px\| < \|v - x\| \tag{9}$$

is satisfied.

Theorem 3.3 *Let $v = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{T}_R$ and $x^* = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} x^*_{j\mu} \in \mathcal{T}_r$, with $r \leq R$, such that $\|v - x^*\| = \inf_{x \in \mathcal{T}_r} \|v - x\|$. Then we have for all $\mu \in \{1, \ldots, d\}$ and all $j \in \{1, \ldots, r\}$*

$$x^*_{j\mu} \in U_\mu := \operatorname{span}\{v_{i\mu} : 1 \le i \le R\}, \tag{10}$$

i.e. $x^* \in \mathcal{U} := \bigotimes_{\mu=1}^d U_\mu$.

Proof Let $P_\mu : V_\mu \to U_\mu$ the orthogonal projection of V_μ onto U_μ and $P := \bigotimes_{\mu=1}^d P_\mu : \mathcal{T} \to \mathcal{U}$ as in Lemma 3.1. Assuming that there are $\mu' \in \{1, \ldots, d\}$ and $j \in \{1, \ldots, r\}$ with $x^*_{j\mu'} \notin U_{\mu'}$, i.e. $x^* \notin \mathcal{U}$, we have

$$\hat{x} := Px^* = \left(\bigotimes_{\mu=1}^d P_\mu\right) \sum_{j=1}^r \bigotimes_{\mu=1}^d x^*_{j\mu} = \sum_{j=1}^r \bigotimes_{\mu=1}^d \underbrace{P_\mu x^*_{j\mu}}_{\in U_\mu} \in \mathcal{U}_r \subset \mathcal{U}.$$

With Corollary 3.2 the inequality $\|v - \hat{x}\| < \|v - x^*\|$ holds true, which contradicts the fact that $\|v - x^*\| = \inf_{x \in \mathcal{T}_r} \|v - x\|$. □

Using Theorem 3.3, we can restrict our search to the finite dimensional tensor subspace $\mathcal{U} = \bigotimes_{\mu=1}^d U_\mu$ as explained in Remark 3.4 and Corollary 3.5.

Remark 3.4 Under the notations and premises of Theorem 3.3, let $\Phi_\mu : \mathbb{R}^{t_\mu} \to U_\mu$ be the canonical isomorphism introduced by the canonical basis of $\mathbb{R}^{t_\mu}$ and an orthonormal basis of U_μ, where $t_\mu := \dim U_\mu$. According to Theorem 3.3, we have $v,\ x^* \in \mathcal{U}$. Therefore, there exist $\alpha_{i\mu}, \xi^*_{j\mu} \in \mathbb{R}^{t_\mu}$ such that

$$v_{i\mu} = \Phi_\mu \alpha_{i\mu} \quad \text{and} \quad x^*_{j\mu} = \Phi_\mu \xi^*_{j\mu}.$$

Further, let $\Phi := \bigotimes_{\mu=1}^d \Phi_\mu$, $\alpha := \sum_{i=1}^R \bigotimes_{\mu=1}^d \alpha_{i\mu} \in \mathcal{S}_R$ and $\xi^* := \sum_{j=1}^{r'} \bigotimes_{\mu=1}^d \xi^*_{j\mu} \in \mathcal{S}_r$, where $\mathcal{S}$ is defined in Eq. (7). We have

$$v = \Phi\alpha \quad \text{and} \quad x^* = \Phi\xi^*$$

and consequently

$$\|v - x\|^2 = \langle \Phi(\alpha - \xi), \Phi(\alpha - \xi)\rangle = \left\langle \alpha - \xi, \Phi^t \Phi(\alpha - \xi)\right\rangle_{\ell_2} = \|\alpha - \xi\|^2_{\ell_2}. \tag{11}$$

Corollary 3.5 *Let the notations and premises of Theorem 3.3 and Remark 3.4 hold. We have*

$$\|v - x^*\| = \operatorname{dist}(v, \mathcal{T}_r) \quad \Leftrightarrow \quad \|\alpha - \xi^*\|_{\ell_2} = \operatorname{dist}(\alpha, \mathcal{S}_r) \tag{12}$$

and

$$\|v - x^*\| = \|\alpha - \xi^*\|_{\ell_2}. \tag{13}$$

Note that Eq. (12) states that the best approximation problem in a tensor space $\left(\bigotimes_{\mu=1}^d V_\mu, \|\cdot\|\right)$ of arbitrary pre-Hilbert spaces V_μ is equivalent to a fixed finite dimensional ℓ_2 minimization problem in $\left(\bigotimes_{\mu=1}^{d} \mathbb{R}^{t_\mu}, \|\cdot\|_{\ell_2}\right)$. For this reason, it is sufficient to consider the original approximation only on $\mathcal{S}_r$. Hereby we have to assume that the computation of an orthonormal basis of U_μ and the coefficient tensor α of v is reasonable, i.e. the numerical cost for the computation of the inner product in V_μ is not expensive.

Theorem 3.3 can be used in different applications, for instance let $\Omega \subseteq \mathbb{R}$, $W^{k_\mu,2}(\Omega)$ be a Sobolev space and $\mathcal{T} := \bigotimes_{\mu=1}^{d} W^{k_\mu,2}(\Omega)$. Theorem 3.3 states that a representation system of a lower rank approximation $\varphi^* = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \varphi_{j\mu}^* \in \mathcal{T}_r$ of $\varphi = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} \varphi_{i\mu} \in \mathcal{T}_R$ satisfying the following regularity property:

$$\varphi_{j\mu}^* \in \operatorname{span}\{\varphi_{i\mu} \in W^{k_\mu,2}(\Omega) : 1 \leq i \leq R\} \quad \text{for all } 1 \leq \mu \leq d,\ 1 \leq j \leq r,$$

i.e. if the origin function φ possesses a special regularity property in the μ-th direction, then the same regularity property holds true for every lower rank approximation φ^*.

In a second example, let $\mathcal{M} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{n\times n}$ be equipped with the Frobenius norm and $A = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} A_{i\mu} \in \mathcal{M}_R$ such that $[A_{i_1\mu}, A_{i_2\mu}] = 0_{\mathbb{R}^{n\times n}}$ for $1 \leq i_1, i_2 \leq R$, where $[A_{i_1\mu}, A_{i_2\mu}]$ denotes the commutator of $A_{i_1\mu}$ and $A_{i_2\mu}$. Using Theorem 3.3 it is obvious to see that the eigenvectors of A are eigenvectors of all lower rank approximations of A. A prominent example of such a tensor is the discretized Laplace operator on a hypercube with Dirichlet boundary conditions.

4 The canonical tensor format with bounded terms and existence of best approximations

In the following let $d \geq 3$ and $\mathcal{S} := \bigotimes_{\mu=1}^{d} \mathbb{R}^{t_\mu}$. The existence of a best approximation is only expected for closed sets. Unfortunately, the set of sums of elementary tensors is in general not closed. This fact was analysed in [6]. In order to discuss the arising difficulties, we will focus on the following tensor:

$$t := a \otimes b \otimes b + b \otimes a \otimes b + b \otimes b \otimes a,$$

where the set $\{a, b\} \subset \mathbb{R}^n$ is linearly independent. Lim and de Silva showed in [6] that the tensor rank of t is exactly three. We define a sequence $(t^k)_{k\in\mathbb{N}}$ of tensors with tensor rank at most two by

$$t^k := \left(\frac{1}{k}a + b\right) \otimes \left(b + \frac{1}{k}a\right) \otimes kb + b \otimes b \otimes (a - kb).$$

Then

$$\|t - t^k\| = \frac{1}{k}\|a \otimes a \otimes b\| \xrightarrow[k\to\infty]{} 0$$

follows. The specified counter-example is of particular importance and should not be neglected, since the Laplacian in 3 dimensions is of this form. A simple analysis of the counterexample shows that for $k \to \infty$ the summands of t^k are unbounded, i.e.

$$\left\| \left(\frac{1}{k}a + b\right) \otimes \left(b + \frac{1}{k}a\right) \otimes kb \right\| \xrightarrow[k\to\infty]{} \infty,$$
$$\|b \otimes b \otimes (a - kb)\| \xrightarrow[k\to\infty]{} \infty.$$

In summary, we can say that the existence of the best approximation is not guaranteed and moreover, because of the unboundedness of the terms, the numerical treatment is not practicable. One obvious approach for solving this problem is given in the following definition.

Definition 4.1 (*Sums of Elementary Tensors with Bounded Terms*) Let $c > 0$ and $r \in \mathbb{N}$. The set of tensors which can be represented with *bounded r-terms* is defined as follows:

$$\mathcal{S}_r^c := \left\{ v = \sum_{i=1}^{r} \underbrace{\bigotimes_{\mu=1}^{d} v_{i\mu}}_{v_i :=} \in \mathcal{S}_r : \sum_{i=1}^{r} \|v_i\|^2 \leq c \right\}. \tag{14}$$

Lemma 4.2 *For fixed $c > 0$ and $r \in \mathbb{N}_0$, the set S_r^c is closed.*

Proof Let $(t^k)_{k\in\mathbb{N}} \subset \mathcal{S}_r^c$ and $t \in \mathcal{S}$ with $\lim_{k\to\infty} t^k = t$, where $t^k = \sum_{i=1}^r \bigotimes_{\mu=1}^d t_{i\mu}^k$. We can assume without loss of generality that $\|t_{i1}^k\| = \cdots = \|t_{id}^k\|$ holds true and consequently the sequence $(t_{i\mu}^k)_{k\in\mathbb{N}}$ is bounded. Hence there exist a subsequence $(t_{i\mu}^{k(l)})_{l\in\mathbb{N}}$ convergent to $\tilde{t}_{i\mu} \in \mathbb{R}^{t_\mu}$. Moreover, we have $t^{k(l)} \xrightarrow[l\to\infty]{} \tilde{t}$ and $\tilde{t} = t$ due to the fact that the canonical tensor format $\mathfrak{C}$ is a continuous mapping, where $\tilde{t} := \sum_{i=1}^r \bigotimes_{\mu=1}^d \tilde{t}_{i\mu} \in \mathcal{S}_r^c$. □

Corollary 4.3 *Let $\lim_{k\to\infty} t^k = t$, where $t^k = \sum_{i=1}^r \bigotimes_{\mu=1}^d t_{i\mu}^k$ and $r < \operatorname{rank}_{\mathcal{S}}(t)$. There exists $i \in \{1, \ldots, r\}$ such that the corresponding term sequence $(t_i^k)_{k\in\mathbb{N}}$ is not bounded.*

Proof Let us assume that for all $i \in \{1, \ldots, r\}$ the sequence $(t_i^k)_{k\in\mathbb{N}}$ is bounded by $c > 0$. According to Lemma 4.2, we have $t \in \mathcal{S}_r^{rc^2} \subset \mathcal{S}_r$, but this contradicts the fact that $\operatorname{rank}_{\mathcal{S}}(t) > r$. □

Corollary 4.4 *$\mathcal{S}_1$ is closed.*

Proof Let $(t^k)_{k\in\mathbb{N}} \subset \mathcal{S}_1$ with $\lim_{k\to\infty} t^k = t \in \mathcal{S}$. Then we have for almost all k that t^k is bounded by $\|t\|$. The rest of the proof is similar to the one of Lemma 4.2. □

Corollary 4.5 *From the Weierstrass theorem it follows that there exists a best approximation of $\alpha \in \mathcal{S}_R$ in $\mathcal{S}_1$ and $\mathcal{S}_r^c$.*

5 Approximation problem and objective function

In this section, the two approximation problems are formulated and the objective function is defined. Furthermore, for the regularized Newton method the first and second derivatives of the objective function are stated. In the following, let $c > 0, \mathcal{S} := \bigotimes_{\mu=1}^d \mathbb{R}^{t_\mu}$, and

$$\mathfrak{C} : \bigtimes_{\mu=1}^{d} \left(\mathbb{R}^{t_\mu}\right)^r \to \mathcal{S}_r$$

the canonical tensor format in $\mathcal{S}$, see Definition 2.2.

Notation 5.1 For short notations we define $\mathfrak{R}_{d,r,\underline{t}} := \bigtimes_{\mu=1}^{d} \left(\mathbb{R}^{t_\mu}\right)^r$, where $\underline{t} := (t_1, \ldots, t_d)$. Further, elements from $\mathfrak{R}_{d,r,\underline{t}}$ are always marked by the hat symbol, e.g. $\hat{\xi} \in \mathfrak{R}_{d,r,\underline{t}}$. If it is obvious from the context, we use the following notation: for given $\hat{\xi} = (\xi_{j\mu} \in \mathbb{R}^{t_\mu} : 1 \leq \mu \leq d,\ 1 \leq j \leq r) \in \mathfrak{R}_{d,r,\underline{t}}$ the corresponding sum of elementary tensors $\mathfrak{C}(\hat{\xi})$ is denoted by ξ, i.e.

$$\xi = \mathfrak{C}(\hat{\xi}) = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu}.$$

The set $\mathfrak{R}_{d,r,\underline{t}}^c$ is defined in analogy to sums of elementary tensors with bounded terms as follows:

$$\mathfrak{R}_{d,r,\underline{t}}^c := \left\{ \hat{\xi} \in \mathfrak{R}_{d,r,\underline{t}} : \sum_{j=1}^{r} \prod_{\mu=1}^{d} \|\xi_{j\mu}\|^2 \leq c \right\}. \tag{15}$$

5.1 Formulation of the approximation problem

Definition 5.2 (*Approximation Problem*) Let

$$\alpha = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} \alpha_{i\mu} \in \mathcal{S}_R \tag{16}$$

and $r < R$ be given. We are looking for $\hat{\xi} = (\xi_{j\mu} \in \mathbb{R}^{t_\mu} : 1 \leq \mu \leq d,\ 1 \leq j \leq r) \in \mathfrak{R}_{d,r,\underline{t}}^c$ such that

$$\|\alpha - \mathfrak{C}(\hat{\xi}^*)\| = \min_{\hat{\xi} \in \mathfrak{R}_{d,r,\underline{t}}^c} \|\alpha - \mathfrak{C}(\hat{\xi})\|, \tag{17}$$

under the constraints[1]

$$\|\xi^*_{j\mu}\| = \|\xi^*_{j\nu}\|, \quad \text{for all } \mu, \nu \in \{1, \ldots, d\},\ j \in \{1, \ldots, r\}. \tag{18}$$

In applications it is a priori not obvious how to choose the size or the representation rank r. Rather, a desired approximation accuracy ε is given. Hence, the following extended approximation problem is stated.

Definition 5.3 (*Extended Approximation Problem*) Given $\alpha \in \mathcal{S}_R$ and $\varepsilon > 0$. Find minimal $r_\varepsilon \leq R$ and $\hat{\xi}_\varepsilon \in \mathfrak{R}^c_{d,r_\varepsilon,\underline{t}}$ such that

$$\|\alpha - \mathfrak{C}(\hat{\xi}_\varepsilon)\| \leq \varepsilon, \tag{19}$$

$$\|\alpha - \mathfrak{C}(\hat{\xi}_\varepsilon)\| = \min_{\hat{\xi} \in \mathfrak{R}^c_{d,r_\varepsilon,\underline{t}}} \|\alpha - \mathfrak{C}(\hat{\xi})\|, \tag{20}$$

under the constraints from Eq. (18).

5.2 Definition of the objective function

Later, we will see that the solution of the extended approximation problem is reduced to a finite sequence of approximation problems from Definition 5.2. Therefore, we first introduce the objective function of the approximation problem with fixed representation rank r. The minimization operates on $\mathfrak{R}_{d,r,\underline{t}}$ with respect to the function

$$\frac{1}{2}\|\alpha - \cdot\|^2 \circ \mathfrak{C} : \mathfrak{R}_{d,r,\underline{t}} \to \mathbb{R}_{\geq 0},$$

where α is defined in Eq. (16). We can neglect constant terms, $\frac{1}{2}\|\alpha - \xi\|^2 = \underbrace{\frac{1}{2}\|\alpha\|^2}_{=const.} - \langle \xi, \alpha \rangle + \frac{1}{2}\|\xi\|^2$. Therefore, the main part of our objective function is

$$f_1 : \mathfrak{R}_{d,r,\underline{t}} \to \mathbb{R}_{\geq -\frac{1}{2}} \tag{21}$$

$$\hat{\xi} \mapsto f_1(\hat{\xi}) := \frac{1}{\|\alpha\|^2}\left[-\langle \alpha, \xi \rangle + \frac{1}{2}\|\xi\|^2\right], \tag{22}$$

[1] Especially in high dimensions it is a good advise to balance the representation system of every tensor which is stored in real computer implementations. Otherwise, one cannot avoid a number overflow in some entries of the representation system. For example consider the rank-one tensor $v := \bigotimes_{\mu=1}^d v_\mu \in \bigotimes_{\mu=1}^d \mathbb{R}^n$ with $(v_\mu)_l = 1$ for all $l \in \{1, \ldots, n\}$. If one introduces the following constrains for the representation system: $\|v_\mu\| = 1$ for all $\mu \in \{2, \ldots, d\}$, we have $\|v_1\| = n^d$. If n and d are large enough, we will produce a number overflow on computer systems. With (18) we have a balanced and solid representation system. Furthermore, with (18) we avoid unnecessary scaling influences in our objective function and consequently their second derivative with respect to the representation system.

where we normalized the objective function for numerical reasons. After manipulations, we get

$$f_1(\hat{\xi}) = \frac{1}{\|\alpha\|^2}\left[-\sum_{j=1}^{r}\sum_{i=1}^{R}\prod_{\mu=1}^{d}\langle\alpha_{i\mu},\xi_{j\mu}\rangle + \frac{1}{2}\sum_{j=1}^{r}\sum_{j'=1}^{r}\prod_{\mu=1}^{d}\langle\xi_{j\mu},\xi_{j'\mu}\rangle\right]. \tag{23}$$

Normally, in constrained minimization we have to satisfy the Karush–Kuhn–Tucker conditions (KKT), but our constraints are of very simple structure. Therefore we can avoid the KKT conditions and treat the constraints as penalty terms in our objective function. Thus we construct the following function g_1 such that $g_1(\hat{\xi}) = 0$, if $\xi \in \mathcal{S}_r$ meets the constraints (18), i.e. we have

$$g_1 : \mathfrak{R}_{d,r,\underline{t}} \to \mathbb{R}_{\geq 0} \tag{24}$$

$$\hat{\xi} \mapsto g_1(\hat{\xi}) := \frac{1}{8\sqrt[d]{\|\alpha\|^4}}\sum_{j=1}^{r}\sum_{1\leq\nu<\mu\leq d}\left(\|\xi_{j\mu}\|^2 - \|\xi_{j\nu}\|^2\right)^2. \tag{25}$$

With g_1 we pay attention to the non-uniqueness of the tensor representation. According to Lemma 4.2, the approximation problem is well-defined only on $\mathcal{S}_r^c$. If the minimum of the minimization problem does not exist, it follows from Corollary 4.3 that a minimizing sequence has an unbounded representation system. One can bound the norm of the terms by using an additional term g_2:

$$g_2 : \mathfrak{R}_{d,r,\underline{t}} \to \mathbb{R}_{\geq 0} \tag{26}$$

$$\hat{\xi} \mapsto g_2(\hat{\xi}) := \frac{1}{2\|\alpha\|^2}\sum_{j=1}^{r}\|\xi_j\|^2 = \frac{1}{2\|\alpha\|^2}\sum_{j=1}^{r}\prod_{\mu=1}^{d}\|\xi_{j\mu}\|^2. \tag{27}$$

This approach has the advantage that the constant c need not to be selected in advance and we avoid the difficult treatment of the KKT conditions. Thus, the complete objective function is described by

$$f : \mathfrak{R}_{d,r,\underline{t}} \to \mathbb{R}_{\geq -\frac{1}{2}} \tag{28}$$

$$\hat{\xi} \mapsto f(\hat{\xi}) := f_1(\hat{\xi}) + \lambda_1 g_1(\hat{\xi}) + \lambda_2 g_2(\hat{\xi}), \tag{29}$$

where $\lambda_1, \lambda_2 > 0$. In practice, we choose the parameter λ_2 such small that the impact on the main part f_1 is not significant, i.e. we want that for $\xi \in \mathcal{S}_r^{c\cdot\|\alpha\|^2}$ the influence of g_2 on f_1 is (much) smaller than ε. We choose $\kappa \in (0, \varepsilon)$ and $\lambda_2 \leq \kappa/2c$, since we have for $\xi \in \mathcal{S}_r^{c\cdot\|\alpha\|^2}$

$$|f_1(\hat{\xi}) - (f_1(\hat{\xi}) + \lambda_2 g_2(\hat{\xi}))| = \lambda_2 g_2(\hat{\xi}) \leq \frac{\kappa}{c}\, g_2(\hat{\xi}) \leq \kappa < \varepsilon.$$

5.3 The first and second derivatives of the objective function

In this section, we are specifying the first and second order derivatives of the objective function. These derivatives are important for the regularized Newton Method.

Notation 5.4 For $1 \leq \mu_1,\ \mu_2 \leq d$ we define $\langle \cdot, \cdot \rangle_{\mu_1 \mu_2}$ by

$$\langle \cdot, \cdot \rangle_{\mu_1 \mu_2} \ : \ \bigtimes_{\mu=1}^{d} \mathbb{R}^{t_\mu} \times \bigtimes_{\mu=1}^{d} \mathbb{R}^{t_\mu} \to \mathbb{R}, \tag{30}$$

$$(\hat{v}, \hat{w}) \mapsto \left\langle \hat{v}, \hat{w} \right\rangle_{\mu_1 \mu_2} := \prod_{\mu \in \{1,\ldots,d\} \setminus \{\mu_1, \mu_2\}} \left\langle v_\mu, w_\mu \right\rangle. \tag{31}$$

We write shorter $\langle v, w \rangle_{\mu_1 \mu_2} := \left\langle \hat{v}, \hat{w} \right\rangle_{\mu_1 \mu_2}$ if $v = \bigotimes_{\mu=1}^{d} v_\mu$ and $w = \bigotimes_{\mu=1}^{d} w_\mu$. For $\mu_1 = \mu_2$ we define $\left\langle \hat{v}, \hat{w} \right\rangle_{\mu_1} := \left\langle \hat{v}, \hat{w} \right\rangle_{\mu_1 \mu_2}$. We define further

$$\bar{\delta}_{ij} := 1 - \delta_{ij}, \tag{32}$$

where δ_{ij} is the Kronecker-delta, i.e.

$$\delta_{ij} := \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \tag{33}$$

Note that the value of $\langle v, w \rangle_{\mu_1 \mu_2}$ depends on the representation system of v and w.

Lemma 5.5 ([7, Lemma 3.3.3, Lemma 3.3.4 and Lemma 3.3.5]) *Let* $1 \leq \mu_1 \leq d$, $1 \leq j_1 \leq r$, f_1, g_1 *and* g_2 *defined as in* (21), (24) *and* (26) *respectively. For the first derivatives we have*

$$f'_{1\,j_1\mu_1}(\hat{\xi}) = \frac{1}{\|\alpha\|^2} \left[-\sum_{i=1}^{R} \left\langle \alpha_i, \xi_{j_1} \right\rangle_{\mu_1} \alpha_{i\mu_1} + \sum_{j=1}^{r} \left\langle \xi_j, \xi_{j_1} \right\rangle_{\mu_1} \xi_{j\mu_1} \right], \tag{34}$$

$$g'_{1\,j_1\mu_1}(\hat{\xi}) = \frac{1}{\sqrt[d]{\|\alpha\|^4}} \left[\sum_{\mu=1, \mu \neq \mu_1}^{d} \left(\|\xi_{j_1\mu_1}\|^2 - \|\xi_{j_1\mu}\|^2 \right) \right] \xi_{j_1\mu_1}, \tag{35}$$

$$g'_{2\,j_1\mu_1}(\hat{\xi}) = \frac{1}{\|\alpha\|^2} \left\langle \xi_{j_1}, \xi_{j_1} \right\rangle_{\mu_1} \xi_{j_1\mu_1}. \tag{36}$$

Furthermore, the first order derivative of the objective function f *from Eq.* (28) *is*

$$f'_{\ j_1\mu_1}(\hat{\xi}) = f'_{1\,j_1\mu_1}(\hat{\xi}) + \lambda_1 g'_{1\,j_1\mu_1}(\hat{\xi}) + \lambda_2 g'_{2\,j_1\mu_1}(\hat{\xi}). \tag{37}$$

The second derivative of f describes in a natural way a block matrix of block matrices.

Lemma 5.6 ([7, Lemma 3.4.3]) *Let* $1 \leq \mu_1, \mu_2 \leq d, 1 \leq j_1, j_2 \leq r$*, and* f_1 *as defined in Eq.* (21)*. For the second derivative* $f_1''(\hat{\xi})$ *we have*

$$f_1''{}_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) \\ = \frac{1}{\|\alpha\|^2}\left[A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) + B_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) + C_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) - D_{\mu_1\mu_2 j_1 j_2}(\hat{\xi})\right], \tag{38}$$

where

$$A_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \delta_{\mu_1\mu_2}\left\langle \xi_{j_1}, \xi_{j_2}\right\rangle_{\mu_1} \mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}, \tag{39}$$

$$B_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \overline{\delta}_{\mu_1\mu_2}\left\langle \xi_{j_1}, \xi_{j_2}\right\rangle_{\mu_1\mu_2} \xi_{j_2\mu_1}\xi^t_{j_1\mu_2}, \tag{40}$$

$$C_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}\sum_{j=1}^{r}\left\langle \xi_j, \xi_{j_1}\right\rangle_{\mu_1\mu_2} \xi_{j\mu_1}\xi^t_{j\mu_2}, \tag{41}$$

$$D_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) := \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}\sum_{i=1}^{R}\left\langle \alpha_i, \xi_{j_1}\right\rangle_{\mu_1\mu_2} \alpha_{i\mu_1}\alpha^t_{i\mu_2}. \tag{42}$$

Lemma 5.7 ([7, Lemma 3.3.4 and Lemma 3.3.5]) *Let* $1 \leq \mu_1, \mu_2 \leq d, 1 \leq j_1, j_2 \leq r$*. For the second derivatives* $g_1''(\hat{\xi})$ *and* $g_2''(\hat{\xi})$ *we have*

$$g_1''{}_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) = \frac{1}{\sqrt[d]{\|\alpha\|^4}}\left[G_{1\mu_1\mu_2 j_1 j_2}(\hat{\xi}) + G_{2\mu_1\mu_2 j_1 j_2}(\hat{\xi})\right], \tag{43}$$

$$g_2''{}_{\mu_1\mu_2 j_1 j_2}(\hat{\xi}) = \frac{1}{\|\alpha\|^2}\delta_{j_1 j_2}\begin{cases}\left\langle \xi_{j_1}, \xi_{j_1}\right\rangle_{\mu_1} \mathbf{Id}_{\mathbb{R}^{t_{\mu_1}}}, & \mu_1 = \mu_2;\\ 2\left\langle \xi_{j_1}, \xi_{j_1}\right\rangle_{\mu_1\mu_2} \xi_{j_1\mu_1}\xi^t_{j_1\mu_2}, & \textit{else,}\end{cases}, \tag{44}$$

where

$$G_{1\mu_1\mu_2 j_1 j_2}(\hat{\xi}) \\ = \delta_{\mu_1\mu_2}\delta_{j_1 j_2}\left[\sum_{\mu=1,\mu\neq\mu_1}^{d}\left(\|\xi_{j_1\mu_1}\|^2 - \|\xi_{j_1\mu}\|^2\right) Id_{\mathbb{R}^{t_{\mu_1}}} + 2(d-1)\xi_{j_1\mu_1}\xi^t_{j_1\mu_1}\right],$$

$$G_{2\mu_1\mu_2 j_1 j_2}(\hat{\xi}) = \overline{\delta}_{\mu_1\mu_2}\delta_{j_1 j_2}(-2)\xi_{j_1\mu_1}\xi^t_{j_1\mu_2}.$$

6 Solution of the approximation problem

6.1 Regularized Newton method

All Newton-like methods are based on approximating the objective function locally by a quadratic model and then minimizing that model approximately, often by Krylov subspace methods. The quadratic model of the objective function f at $\hat{\xi}$ in direction d is given by the Taylor polynomial of second order:

$$f(\hat{\xi}^k + d) \approx q_k(\hat{\xi}) := f(\hat{\xi}^k) + \left\langle f'(\hat{\xi}^k), \hat{\xi} - \hat{\xi}^k \right\rangle + \frac{1}{2} \left\langle f''(\hat{\xi}^k) \left(\hat{\xi} - \hat{\xi}^k\right), \left(\hat{\xi} - \hat{\xi}^k\right) \right\rangle.$$

The successor $\hat{\xi}^{k+1}$ is the minimum of the minimization problem

$$\min_{\hat{\xi} \in \mathfrak{R}_{d,r,\underline{t}}} q_k(\hat{\xi}),$$

if and only if

$$\hat{\xi}^{k+1} = \hat{\xi}^k - d^k$$

and the Hessian matrix $f''(\hat{\xi}^k)$ is positive definite, where d^k solves the Newton equation approximately, i.e.

$$\left\| f'(\hat{\xi}^k) - f''(\hat{\xi}^k) d^k \right\| = o(\|d^k\|). \tag{45}$$

It is well known from theory that $\hat{\xi}^k \xrightarrow[k \to \infty]{} \hat{\xi}^*$ converges at least locally superlinearly and $f'(\hat{\xi}^*) = 0$ if $f''(\hat{\xi}^*)$ is regular. Note that we treat the non-uniqueness of the tensor representation with the use of $g_1(\hat{\xi})$ in Eq. (28). Furthermore, if in addition f'' is local Lipschitz continuous, we have that $(\hat{\xi}^k)_{k \in \mathbb{N}}$ converges to $\hat{\xi}^*$ at least quadratically. Nevertheless, computational difficulties arise with the above mentioned method when the function f is strongly nonlinear. These difficulties usually result from a ill-conditioned Hessian matrix, making the inversion process numerically challenging. Since our objective function f is non-convex, the Hessian is in general not positive definite. Therefore, Newton's method will not converge in general. There are several ways to modify the Newton method for unconstrained minimization to achieve global convergence. For twice continuously differentiable and strongly convex functions, the Newton direction is a descent direction. The local "quality" of the Newton direction at each point can be estimated by the condition number of the Hessian at this point. If the condition number is bounded from above uniformly, then by introducing a step-size, it is possible to guarantee global convergence of the so-called damped Newton method. By adjusting the step-size of the damped Newton method, using for example the Armijo rule, the asymptotic quadratic rate of convergence can be achieved. To guarantee global convergence of the Newton method in case when the function is not strongly convex, regularization of the Hessian is used. Although this scheme converges globally, it will not necessarily converge to a global optimum. A standard minimization method fits into the general scheme of a descent method with step-size strategy, i.e. starting from a given iterant $\hat{\xi}^k$ one first determines a descent direction d^k and then computes a successor $\hat{\xi}^{k+1}$ on the ray $\{\hat{\xi}^k - \alpha d^k : \alpha \in \mathbb{R}_{\geq 0}\}$ which is defined by $\hat{\xi}^k$ and d^k such that $f(\hat{\xi}^{k+1})$ is sufficiently small compared to $f(\hat{\xi}^k)$, where the descent direction d^k is a solution to the unconstrained quadratic optimization problem

$$\min_{d \in \mathfrak{R}_{d,r,\underline{t}}} q_k(d). \tag{46}$$

In general, the quadratic function q_k can approximate the highly non-linear objective function f only locally. Nevertheless, in standard minimization methods one takes d^k as a search direction and introduces a line search. A regularized Newton method can be viewed as an extension of this principle. But here, in contrast, the successor $\hat{\xi}^{k+1}$ is located on a arbitrary spatial curve, where the curve is not necessarily a ray like $\{\hat{\xi}^k - \alpha d^k : \alpha \in \mathbb{R}_{\geq 0}\}$. Here, the calculation of the descent direction leads directly to the successor $\hat{\xi}^{k+1}$; a subsequent step-size calculation is unnecessary. However, this coupling has an evident disadvantage. In order to ensure convergence, one has to evaluate the successor $\hat{\xi}^{k+1}$. If the successor is not admissible, $\hat{\xi}^{k+1}$ is discarded and a new system of linear equations has to be solved. This leads to high computational cost, but it can be corrected in some respects. We will also use a damped regularized Newton method and, in addition, we will adopt the ideas of the regularized Newton method using special properties of our objective function. For the regularized Newton method the following constrained minimization problem is of vital importance:

$$\min_{d(r_k)\in\mathfrak{R}_{d,r,\underline{t}}} q_k(d(r_k)) \quad \text{subject to} \quad \|d(r_k)\|_{A_k} \leq r_k,$$

for some parameter $r_k \in \mathbb{R}_+$ and a positive definite matrix A_k. Note that in regularized Newton methods the descent direction depends on the parameter r_k. For this reason, this method is also called a parameter dependent descent direction method. The parameter r_k defines a trust region and therefore this problem is also called trust region subproblem. For every r_k there exists exactly one solution of the trust region subproblem. This solution is described by

$$\hat{\xi}^{k+1}(r_k) = \hat{\xi}^k - d^k(r_k), \tag{47}$$

where

$$\left\| f'(\hat{\xi}^k) - (\lambda_k A_k + f''(\hat{\xi}^k))d^k(r_k) \right\| = o(\|d^k(r_k)\|) \tag{48}$$

and $\lambda_k \in \mathbb{R}_+$ is uniquely determined by the problem

$$\varphi(\lambda_k) := \|(\lambda_k A_k + f''(\hat{\xi}^k))^{-1} f'(\hat{\xi}^k)\|_{A_k} = r_k,$$

see Lemma A.1 for more details. The substitution $\omega_k := (1 - \lambda_k)/\lambda_k$ leads to

$$\lambda_k A_k + f''(\hat{\xi}^k) \mapsto \hat{H}(\hat{\xi}^k, \omega_k) := \omega_k f''(\hat{\xi}^k) + (1 - \omega_k) A_k \tag{49}$$

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

In the regularized Newton method the standard choice is $A_k := \mathbf{Id}$ which leads for $d^k(\omega_k)$ to the gradient direction as $\omega_k \xrightarrow[k\to\infty]{} 0$, but in practice we observe that the positive definite matrix

$$A_k := A(\hat{\xi}^k), \tag{50}$$

Algorithm 1 Regularized Newton Method (RNM)

```
1: Choose $\hat{\xi}^1 \in \mathcal{S}, \gamma, \beta \in (0,1), \varepsilon > 0, \sigma \in (0, \frac{1}{2})$ $\delta > 0$, define $k := 1$ and $\omega_0 := 1$.
2: while $\|f'(\hat{\xi}^k)\| > \varepsilon$ do
3:   $\omega_k := \min\left\{\frac{\omega_{k-1}}{\gamma}, 1\right\}$.
4:   Compute $d^k$ as a solution of
         $$\hat{H}(\hat{\xi}^k, \omega_k)d^k = f'(\hat{\xi}^k)$$
     by the cg-method. If the cg-method does not converge or the condition
         $$\frac{\left\langle f'(\hat{\xi}^k), d^k \right\rangle}{\|f'(\hat{\xi}^k)\|\|d^k\|} \geq \min\{\delta, \|f'(\hat{\xi}^k)\|^2\} \qquad (51)$$
     is false, we set $\omega_k := \gamma\omega_k$ and continue with Step 1.
5:   Compute $\bar{\omega}_k > 0$ by the Armijo rule
         $$\bar{\omega}_k := \max_{l \in \mathbb{N}_0} \left\{\beta^l : f(\hat{\xi}^k) - f(\hat{\xi}^k - \beta^l d^k) \geq \sigma\beta^l \left\langle f'(\hat{\xi}^k), d^k \right\rangle\right\}. \qquad (52)$$
6:   Set $\hat{\xi}^{k+1} := \hat{\xi}^k - \bar{\omega}_k d^k$ and $k \mapsto k+1$.
7: end while
```

see Eq. (39), gives much better results. Furthermore, from Lemma A.3 and Lemma A.2 follows that without loss of generality A_k has a condition number uniformly bounded from above, since $d^k(0)$ is a gradient-like descent direction. Note that in cases where $\lambda_{\min}(A_k) < \gamma_{\min}$ one can use the ideas described in Lemma A.3 and Lemma A.2 to reduce the tensor rank of the current iterant $\xi^k = \mathfrak{C}(\hat{\xi}^k)$. Thus we obtain a direction $d^k(\omega_k)$ which for $\omega_k \to 1$ is the Newton direction and for $\omega_k \to 0$ a descent direction. While solving system (48) iteratively by the cg-iteration, one can exploit that the convergence depends on the positivity of $\hat{H}$. As long as the iteration diverges or the residual is not properly decreased in a defined number of cg-iterations, we decrease ω_k. Also, when (51) is not fulfilled we restart the process and decrease ω_k. Thus, during the iterative solve the parameter ω_k can be adaptively determined.

Remark 6.1 In our numerical experiments with data discussed in [4,5], we noticed that the following modification of the system matrix from Eq. (49) leads to better results, see numerical results in Sect. 7. We redefine the system matrix by

$$\hat{H}(\hat{\xi}_k, \omega_k) := \begin{cases} A_k(\hat{\xi}_k) + \omega_k\left(B_k(\hat{\xi}_k) + \lambda_1 G_{1k}(\hat{\xi}_k) + \lambda_2 G_{2k}(\hat{\xi}_k)\right), & \overline{\omega}_k \neq 1 \\ A_k(\hat{\xi}_k) + \omega_k\Big(B_k(\hat{\xi}_k) + C_k(\hat{\xi}_k) - D_k(\hat{\xi}_k) & \overline{\omega}_k = 1, \\ \quad +\lambda_1 G_{1k}(\hat{\xi}_k) + \lambda_2 G_{2k}(\hat{\xi}_k)\Big), & \end{cases} \tag{53}$$

where A_k, B_k, C_k, D_k, G_{1k} and G_{2k} are stated in Lemma 5.6 and Lemma 5.7. The idea behind this definition can be explained as follows. For $\overline{\omega}_k = 1$, we expect that the current iterant is potentially located in an environment where the regularized Newton method converges quadratically. Therefore, we choose for the next iteration step the second derivative of f. If $\overline{\omega}_k \neq 1$ we choose only selected terms of the second order derivative, see Lemma 5.6. Notice that with the new definition of $\hat{H}(\hat{\xi}_k, \omega_k)$ we have that the system matrix is independent of R (the tensor rank of the given tensor α). Therefore, this choice is also favourably in cases where R is large.

Since we solve Eq. (48) by the cg-method, we have to consider the numerical complexity of a matrix vector multiplication performed by the system matrix $\hat{H}(\hat{\xi}_k, \omega_k)$. With this new definition we will see later that in cases where $\overline{\omega}_k \neq 1$, the numerical cost for solving (48) is reduced. According to Lemma A.2 the complexity for the inversion of the matrix $A_k := A_k(\hat{\xi}_k)$ is $\mathcal{O}(dr^3)$. Therefore an obvious choice of a preconditioner for the cg-method applied to the linear system (48) is A_k. This definition leads to

$$A_k^{-1}\hat{H}(\hat{\xi}_k, \omega_k) := \begin{cases} \mathbf{Id} + \omega_k A_k^{-1}\Big(B_k(\hat{\xi}_k) + \lambda_1 G_{1k}(\hat{\xi}_k) + \lambda_2 G_{2k}(\hat{\xi}_k)\Big), & \overline{\omega}_k \neq 1 \\ \mathbf{Id} + \omega_k A_k^{-1}\Big(B_k(\hat{\xi}_k) + C_k(\hat{\xi}_k) - D_k(\hat{\xi}_k) & \overline{\omega}_k = 1, \\ \quad +\lambda_1 G_{1k}(\hat{\xi}_k) + \lambda_2 G_{2k}(\hat{\xi}_k)\Big), & \end{cases}$$

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

6.2 Convergence of the regularized Newton method

In the following we repeat the results about the convergence of the regularized Newton method, see [12, Section 9.6], [17, Section 8.2], and [13, Section 14.4] for a detailed description.

Theorem 6.2 *Let the sequence $(\hat{\xi}^k)_{k\in\mathbb{N}}$ generated by Algorithms 1. Then every accumulation point $\hat{\xi}^*$ of $(\hat{\xi}^k)_{k\in\mathbb{N}}$ is a critical point, i.e. $f'(\hat{\xi}^*) = 0$. Moreover, if $f''(\hat{\xi}^*)$ is positive definite and $\|\hat{H}(\hat{\xi}_k, \omega_k) - f''(x^k)\| \xrightarrow[k\to\infty]{} 0$ then we have quadratic convergence of $(\hat{\xi}^k)_{k\in\mathbb{N}}$ to $\hat{\xi}^*$ and $\hat{\xi}^*$ is a strict local minimum of f from Eq. (28).*

6.3 Complexity analysis

In the following, the numerical complexity of the previously described regularized Newton method applied to our objective function is studied. The main part of the numerical cost is due to the solution of (48). For (48) we have to compute the first and second order derivative of f from Eq. (28). Furthermore, for the computation of the descent direction we use the cg-method, therefore one needs the matrix vector multiplication of the system matrix from (53). Finally, with the use of the Armijo rule, we have to compute the step-size parameter $\overline{\omega}_k$. Obviously, this cost is similar to that for the computation of the gradient and is negligible compared to the complexity of the inversion of the system matrix. According to Lemma 5.6 and Lemma 5.7, we only need to compute the inner products $\langle\cdot,\cdot\rangle_{\mu_1\mu_2}$ from Notation 5.4 for the system matrix.

Since the other parts of $\hat{H}(\hat{\xi}_k, \omega_k)$ consist of vectors from the given representation system of α and ξ^k. In Corollary A.5 it is shown that the complexity for the computation of the inner products $\langle\cdot,\cdot\rangle_{\mu_1\mu_2}$ is $\mathcal{O}\left(r\cdot(r+R)\cdot\left(d^2+\sum_{\mu=1}^d t_\mu\right)\right)$. Below, we will analyse the numerical cost for the computation of the gradient and the matrix vector multiplication. Let $\mathcal{S} := \bigotimes_{\mu=1}^d \mathbb{R}^{t_\mu}$, $\xi \in \mathcal{S}_r$, and $\alpha \in \mathcal{S}_R$ like in Sect. 5.

Lemma 6.3 ([7, Lemma 5.3.2]) *The complexity of the computation of the first order derivative at $\hat{\xi}^k$ is*

$$\mathcal{O}\left(r\cdot(r+R)\cdot\sum_{\mu=1}^d t_\mu\right). \tag{54}$$

Lemma 6.4 ([7, Corollary 5.3.14]) *The complexity of the matrix vector multiplication by the system matrix from Eq. (53) is*

$$\mathcal{O}\left(r\cdot(d+r)\cdot\sum_{\mu=1}^d t_\mu\right) \tag{55}$$

and

$$\mathcal{O}\left(r\cdot(d+r+R)\cdot\sum_{\mu=1}^d t_\mu\right) \tag{56}$$

for $\overline{\omega}_k \neq 1$ and $\overline{\omega}_k = 1$, respectively.

For the global convergence of a minimization method, the calculation of the step-size parameter $\overline{\omega}_k$ by Armijo's rule is important. For the Armijo rule, one has to evaluate the function $\beta \mapsto f(\hat{\xi}^k - \beta d^k)$. The numerical cost for one function evaluation for given $\hat{\xi}^k$ and d^k is stated below.

Lemma 6.5 ([7, Corollary 5.3.16]) *The number of arithmetic operations in order to evaluate the function $\beta \mapsto f(\hat{\xi}^k - \beta d^k)$ scales as*

$$\mathcal{O}\left(r\cdot\left[(r+R)\cdot\sum_{\mu=1}^d t_\mu + d\cdot(r+R+d)\right]\right). \tag{57}$$

Corollary 6.6 *The overall complexity of a minimization step is*

$$\mathcal{O}\left(r\cdot(r+R)\cdot d^2 + d\cdot r^3 + r\cdot(r+R+d)\cdot\sum_{\mu=1}^d t_\mu\right). \tag{58}$$

6.4 Solution of the extended approximation problem and systematic choice of initial guesses

The solution of the extended approximation problem

$$\begin{aligned} \|\alpha - \mathfrak{C}(\hat{\xi}_\varepsilon)\| &\leq \varepsilon, \\ \|\alpha - \mathfrak{C}(\hat{\xi}_\varepsilon)\| &= \min_{\hat{\xi} \in \mathfrak{R}^c_{d,r_\varepsilon,\underline{t}}} \|\alpha - \mathfrak{C}(\hat{\xi})\| \end{aligned}$$

from Definition 5.3 is closely related to the choice of the initial guess for the regularized Newton method. We are introducing a scheme which solves the extended approximation problem by the successive use of the regularized Newton method. With the concrete definition of our initial guess described below, we ensure that the approximation error will not increase. Before we start with the description of the solution of the extended approximation problem, we will explain some simple but in practice very useful methods in order to create and improve the initial guess. As we already mentioned, the choice of initial guesses is very important for iterative methods, in particular, for the regularized Newton method since every iteration step is relatively expensive.

Definition 6.7 (*Fibre and Cross*) Let $\underline{i} := (i_1, \ldots, i_d) \in \mathop{\times}_{\mu=1}^{d} \{1, \ldots, t_\mu\}$ be a multi-index and $1 \leq \mu \leq d$. The *fibre* of $\underline{i}$ in direction μ is defined as the following set:

$$\underline{i}^\mu := \left(\mathop{\times}_{\nu=1}^{\mu-1} \{i_\nu\} \right) \times \{1, \ldots, t_\mu\} \times \left(\mathop{\times}_{\nu=\mu+1}^{d} \{i_\nu\} \right). \tag{59}$$

The *cross* $\kappa^{\underline{i}}$ of $\underline{i}$ is the union of the fibres in all directions, i.e.

$$\kappa^{\underline{i}} := \bigcup_{\mu=1}^{d} \underline{i}^\mu. \tag{60}$$

Algorithm 2 Successive Cross Approximation (SCA)

```
Choose $r \in \mathbb{N}$ and $\xi_0 := 0 \in \mathcal{S}$
for $i = 1$ to r do
    $\varrho_i := \alpha - \xi_{i-1}$
    Compute a rank-one cross approximation $\xi$ of $\varrho_i$ as defined in Lemma 6.8
    $\xi_i := \xi_{i-1} + \frac{\langle \varrho_i, \xi \rangle}{\|\xi\|^2} \xi$
end for
```

Lemma 6.8 *Let* $k \in \mathbb{N}$, $\beta := \sum_{j=1}^{k} \bigotimes_{\mu=1}^{d} \beta_{j\mu} \in \mathcal{S}_k$, *and* $\underline{i} := (i_1, \ldots, i_d) \in \mathop{\times}_{\mu=1}^{d} \{1, \ldots, t_\mu\}$, *with* $\beta_{\underline{i}} \in \mathbb{R} \backslash \{0\}$. *Moreover let*

$$\xi^{\underline{i}} := \frac{1}{\left[\beta_{\underline{i}}\right]^{d-1}} \bigotimes_{\mu=1}^{d} \xi^{\underline{i}}_{\mu} \in \mathcal{S}_1, \ \textit{where} \ \xi^{\underline{i}}_{\mu} := \sum_{j=1}^{k} \left(\prod_{\nu=1,\nu\neq\mu}^{d} \left(\beta_{j\nu}\right)_{i_\nu} \right) \beta_{j\mu} \in \mathbb{R}^{t_\mu}. \quad (61)$$

For all $\underline{m} \in \kappa^{\underline{i}}$ *we have*

$$\xi^{\underline{i}}(\underline{m}) = \beta(\underline{m}). \quad (62)$$

Proof Let $\underline{m} \in \kappa^{\underline{i}}$ and without loss of generality $\underline{m} := (i_1, \ldots, i_{d-1}, l_d)$, $1 \leq l_d \leq t_d$. We have

$$\begin{aligned}\xi^{\underline{i}}_{\underline{m}} &= \frac{1}{\left[\beta_{\underline{i}}\right]^{d-1}} \prod_{\mu=1}^{d-1} \left[\sum_{j=1}^{k} \prod_{\nu=1,\nu\neq\mu}^{d} \left(\beta_{j\nu}\right)_{i_\nu} \left(\beta_{j\mu}\right)_{i_\mu} \right] \left[\sum_{j=1}^{k} \prod_{\nu=1}^{d-1} \left(\beta_{j\nu}\right)_{i_\nu} \left(\beta_{jd}\right)_{l_d} \right] \\ &= \frac{1}{\left[\beta_{\underline{i}}\right]^{d-1}} \left[\beta_{\underline{i}}\right]^{d-1} \left[\sum_{j=1}^{k} \prod_{\nu=1}^{d-1} \left(\beta_{j\nu}\right)_{i_\nu} \left(\beta_{jd}\right)_{l_d} \right] = \sum_{j=1}^{k} \prod_{\nu=1}^{d-1} \left(\beta_{j\nu}\right)_{i_\nu} \left(\beta_{jd}\right)_{l_d} = \beta_{\underline{m}}.\end{aligned}$$

□

Remark 6.9 We call the elementary tensor from (61) a rank-one cross interpolation of β. For the cross approximation it is possible that $\|\beta - \xi^{\underline{i}}\| \geq \|\beta\|$. In this case the zero tensor $0_{\mathcal{S}}$ is a better approximation of β. This fact can be avoided by a simple scaling of $\xi^{\underline{i}}$ with $\left\langle \beta, \xi^{\underline{i}} \right\rangle / \|\xi^{\underline{i}}\|^2$, since we have

$$\left\| \beta - \frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle}{\|\xi^{\underline{i}}\|^2} \xi^{\underline{i}} \right\|^2 = \|\beta\|^2 - 2 \frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle^2}{\|\xi^{\underline{i}}\|^2} + \frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle^2}{\|\xi^{\underline{i}}\|^4} \|\xi^{\underline{i}}\|^2 = \|\beta\|^2 - \frac{\left\langle \beta, \xi^{\underline{i}} \right\rangle^2}{\|\xi^{\underline{i}}\|^2} \leq \|\beta\|^2.$$

The first method creates an initial guess by successively computing rank-one cross approximations (SCA), see Algorithm 2. It is clear that the SCA algorithm produces only rough approximations of a given tensor. For this reason we introduce a new scheme which improves a given approximation. This approach is based on tensor rank-one approximations. We are alternating over the terms of the given approximation and improve the approximation quality by defining the corresponding residual. This residual will be approximated further, therefore we need rank-one approximation methods. A complete algorithmic description of our method is given in Algorithm 3. There are different well known methods for the rank-one approximation of a tensor, see [18]. In our implementation, we are using the ALS algorithm. The method described in Algorithm 3 has a numerical complexity of $\mathcal{O}\left(k_{\max} r \cdot R \cdot \sum_{\mu=1}^{d} t_\mu\right)$. This complexity is negligible compared with the complexity of the regularized Newton method. Hence, this algorithm is a good choice for the improvement of the initial guess for the regularized Newton method.

The method which solves the extended approximation problem from Definition 5.3 is stated in Algorithm 4. Starting from an initial guess, we compute a locally best rank-r approximation ξ_r, where r is the rank of the initial guess. Further, we compute

Algorithm 3 Improve Approximation (IA)

```
Given $\alpha \in \mathcal{S}_R$ and $\xi := \sum_{j=1}^{r} \xi_j \in \mathcal{S}_r$. Choose $k_{\max} \in \mathbb{N}$.
for $k = 1$ to $k_{\max}$ do
  for $i = 1$ to r do
    $\varrho_i := \alpha - \left(\sum_{j=1}^{i-1} \xi_j + \sum_{j=i+1}^{r} \xi_j\right)$
    Compute a rank-one approximation $\xi$ of $\varrho_i$ with $\xi_i$ as initial guess.
    $\xi_i := \xi$
  end for
end for
```

a best rank-one approximation ψ_1 of the residual $\varrho_r := \alpha - \xi_r$. Finally, we use the regularized Newton method to compute a locally best rank-$(r+1)$ approximation of α, where $\xi_{r+1}^{(0)}$ is the initial guess, see Algorithm 4 for the complete description.

Algorithm 4 Computation of an Optimal ε-Approximation

```
Given: $\varepsilon > 0$, $\alpha$ and initial guess $\tilde{\xi}$.
$r := \text{rank}_{\mathcal{S}}(\tilde{\xi})$.
Call improve approximation IA($\alpha, \tilde{\xi}$).
Compute a local best approximation $\xi_r$ of $\alpha$ in $\mathcal{S}_r$ with the use of the regularized Newton method, see
Algorithm 1, where $\tilde{\xi}$ is the initial guess.
Define $\varrho_r := \alpha - \xi_r$.
while $\|\varrho_r\| > \varepsilon$ and $r < R$ do
  Compute a rank-one cross interpolation $\zeta^{\underline{i}}$ of $\varrho_r$.
  Compute a locally best approximation $\zeta_1$ of $\varrho_r$ in $\mathcal{S}_1$ with the use of the regularized Newton method,
  where $\zeta^{\underline{i}}$ is the initial guess, where $\mathcal{S}_1$ is defined in Definition 2.2.
  if $\|\varrho_r - \zeta_1\| = \|\varrho_r\|$ then
    Fill $\zeta_1$ with random numbers and redefine $\zeta_1 \mapsto \frac{\langle \zeta_1, \varrho_r \rangle}{\|\zeta_1\|^2} \zeta_1$.
  end if
  Define $\xi_{r+1}^{(0)} := \xi_r + \zeta_1$.
  Call improve approximation IA($\alpha, \xi_{r+1}^{(0)}$).
  Compute a locally best approximation $\xi_{r+1}$ of $\alpha$ in $\mathcal{S}_{r+1}^c$ with the use of the regularized Newton
  method, where $\xi_{r+1}^{(0)}$ is the initial guess.
  Define $\varrho_{r+1} := \alpha - \xi_{r+1}$ and $r \mapsto r + 1$.
end while
if $r = R$ then
  Define $\xi_r := \alpha$.
end if
```

Remark 6.10 With the definition of the initial guess we ensure that

$$\|\alpha - \xi_{r+1}^{(0)}\| \leq \|\alpha - \xi_r\|, \tag{63}$$

even in cases where ξ_r is only a local minimum of the original approximation problem, see [7, Lemma 5.4.5].

Remark 6.11 The complexity of the method described in Algorithm 4 is

$$\mathcal{O}\left(\sum_{r=\mathrm{rank}_{\mathcal{S}}(\tilde{\xi})}^{r_\varepsilon} k_r \cdot \left[r\cdot(r+R)\cdot d^2 + d\cdot r^3 + r\cdot(r+R+d)\cdot\sum_{\mu=1}^{d} t_\mu\right]\right), \tag{64}$$

where k_r is the number of iterations in the regularized Newton method for the rank-r approximation.

Notation 6.12 For a given $\alpha \in \mathcal{S}_R$ and $\varepsilon > 0$ we will denote the solution of Algorithm 4 by

$$\mathfrak{App}_\varepsilon(\alpha). \tag{65}$$

7 Numerical experiments

Our numerical experiments are based on the above algorithm's C++ implementations, where the computation itself is performed on an Intel Core 2 Duo Processor T7300 2.0 GHz, dual core.

7.1 Model problem

Our model problem is the Poisson equation in d dimensions on $\Omega := [0,1]^d$ with Dirichlet boundary conditions, i.e.

$$\begin{aligned} -\Delta u &= h,\\ u\,|_\Omega &= 0. \end{aligned}$$

The function h is defined as follows

$$\begin{aligned} h : \overline{\Omega} &\to \mathbb{R},\\ \underline{x} := (x_1,\ldots,x_d) &\mapsto \sum_{\mu=1}^{d} \prod_{\nu=1,\nu\neq\mu}^{d} \varphi(x_\nu)\left(-2 + (4-12x_\mu)\prod_{\nu=1,\nu\neq\mu}^{d} 2x_\nu\right), \end{aligned}$$

where $\varphi : [0,1] \to \mathbb{R}$, $t \mapsto \varphi(t) := (1-t)t$. The function h has been chosen such that the following function u with tensor rank equal to two is the solution of our model problem:

$$\begin{aligned} u : \overline{\Omega} &\to \mathbb{R} \\ \underline{x} := (x_1,\ldots,x_d) &\mapsto u(\underline{x}) := \prod_{\nu=1}^{d} \varphi(x_\nu)\left(1 + \prod_{\nu=1}^{d} 2x_\nu\right). \end{aligned} \tag{66}$$

A standard finite difference discretization on a uniform grid leads to a linear system $AU(\mathbf{u}) = b$ with

$$A = \mathbf{T} \otimes \mathbf{Id} \otimes \cdots \otimes \mathbf{Id} + \cdots + \mathbf{Id} \otimes \cdots \otimes \mathbf{Id} \otimes \mathbf{T}, \quad b = \sum_{i=1}^{R} \bigotimes_{\mu=1}^{d} b_{i,\mu},$$

where the matrix $\mathbf{T}$ is a discretized version of the second derivative, e.g.

$$T = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

The investigated high-dimensional partial differential equation is of relatively simple nature. Nevertheless, for a first numerical experiment this problem is a good choice, because:

- After discretization, we can easily represent all tensors in the canonical tensor format. Even Δ^{-1} can be approximated in the canonical format format. For this purpose, we need the approximation of the function $\tau \mapsto \frac{1}{\tau}$ by exponential sums as described in [3].
- We can examine the model error introduced by discretization and approximation of the inverse of the Laplacian since the solution u is given explicitly. This allows us to observe the iterative behaviour of our algorithm in particular in cases where we approximate tensors with additional noisy data.

Lemma 7.1 ([3]) *Let* $s_L(\tau) := \sum_{l=1}^{L} \omega_l \exp(-\alpha_l \tau)$ *with* $\alpha_l, \omega_l > 0$. *With the optimal choice of the parameter* α_l *and* ω_l *we have*

$$\sup_{\tau \in [1,c]} \left| \frac{1}{\tau} - s_L(\tau) \right| \leq 16 \exp\left(\frac{-L\pi^2}{\log(8c)} \right).$$

From this approximation follows that for the optimal choice of α_l and ω_l,

$$\|A^{-1} - s_L(A)\|_2 \leq \frac{16}{\lambda_{\min}(A)} \exp\left(\frac{-L\pi^2}{\log(8\kappa(A))} \right),$$

where $s_L(A) = \sum_{l=1}^{L} \omega_l \otimes_{\mu=1}^{d} \exp(-\alpha_l T)$. The parameters α_l and ω_l are precomputed for different k and c and are available at the web page [9]. The analytic solution of Eq. (66) is evaluated at the grid points. We denote the resulting tensor by $\underline{u}$. Furthermore, we define $\underline{\tilde{u}}_L := s_L(A)\underline{h}$, where $\underline{h}$ is the function generated tensor of h. The discretization error of our example is $\mathcal{E}_L := \frac{\|\underline{\tilde{u}}_k - \underline{u}\|}{\underline{\tilde{u}}_k}$. Our numerical examples are performed for $k \in \{15, 42\}$ and $d \in \{10, 20, 50, 100\}$ such that the model errors are $\mathcal{E}_{42} \leq 9.3 \times 10^{-7}$ and $\mathcal{E}_{15} \leq 1.125 \times 10^{-4}$. Moreover, for the approximations $\underline{\tilde{u}}_{42}$

and $\underline{\tilde{u}}_{15}$ we have $\underline{\tilde{u}}_{42} \in \mathcal{S}_{84 \cdot d}$ and $\underline{\tilde{u}}_{15} \in \mathcal{S}_{30 \cdot d}$, i.e. after the matrix vector multiplication, the tensor $\underline{\tilde{u}}_{42}$ is represented by $84 \cdot d$ elementary tensors and accordingly $\underline{\tilde{u}}_{15}$ is represented by $30 \cdot d$ terms. Since $\underline{u}$ has tensor rank two, we approximate $\underline{\tilde{u}}_k$ in the canonical tensor format. In the Tables 1, 2, 3, 4, 5, 6, 7 and 8, the columns are: the tensor rank r of the optimal low rank approximation $\underline{u}_r$, the relative approximation error of the initial guess $\underline{u}_r^{(0)}$, the relative approximation error of $\underline{u}_r$, the norm of the gradient of the objective function f of the limit $\underline{u}_r$, the number of minimization steps in the regularized Newton method and the used CPU time in seconds. Independently of the dimension d and the accuracy $\mathcal{E}_L$, the introduced algorithm finds in all numerical examples for $r = 2$ an approximation $\underline{u}_r$ which approximates $\underline{\tilde{u}}_L$ with the same quality as the prescribed accuracy. The regularized Newton method needs a maximum of 10 iterations in these cases. As an example, we state the progression of the iteration in Table 2. During this calculations the regularized Newton method does not need the shift described in Algorithm 1, i.e. we have $\omega_k = 1$ for all k. As long as we do not reach the model accuracy, the regularization of the Hessian is not necessary and our method works like the Newton method. Furthermore, the algorithm converges quadratically. However, the situation is different if we try to approximate the tensor $\underline{\tilde{u}}_k$ below the given accuracy $\mathcal{E}_L$. Here the shift parameters is $\omega_k < 1$ and

Table 1 Low tensor rank approximation of $\underline{\tilde{u}}_{42}$ from the model problem for $d = 10$, $R = 840$, $n = 1000$, and $\mathcal{E}_{42} = 2.508 \times 10^{-7}$

r	$\frac{\Vert \underline{\tilde{u}}_L - \underline{u}_r^{(0)} \Vert}{\Vert \underline{\tilde{u}}_L \Vert}$	$\frac{\Vert \underline{\tilde{u}}_L - \underline{u}_r \Vert}{\Vert \underline{\tilde{u}}_L \Vert}$	$\Vert f'(\underline{u}_r) \Vert$	RNM-iterations	CPU-time (s)
1	3.098×10^{-1}	1.861×10^{-1}	1.642×10^{-9}	4	0.07
2	8.722×10^{-2}	5.162×10^{-8}	1.362×10^{-9}	9	0.15

Table 2 Iteration of the RNM for $d = 10$, $R = 840$, $r = 2$, $n = 1000$, $\mathcal{E}_{42} = 2.508 \times 10^{-7}$, and k is the iteration index in the regularized Newton method

k	$\left\Vert f'(\underline{u}_r^{(k)}) \right\Vert$	ω_k	$\bar{\omega}_k$	$\frac{\left\Vert \underline{\tilde{u}}_{42} - \underline{u}_r^{(k)} \right\Vert}{\Vert \underline{\tilde{u}}_{42} \Vert}$
0	3.0571×10^{-2}	1.00	–	8.7216×10^{-2}
1	7.7305×10^{-2}	1.00	0.25	7.5917×10^{-2}
2	1.0317×10^{-1}	1.00	0.25	6.9140×10^{-2}
3	1.0359×10^{-1}	1.00	0.25	5.9843×10^{-2}
4	1.0125×10^{-1}	1.00	0.50	4.6120×10^{-2}
5	2.2740×10^{-2}	1.00	1.00	9.3248×10^{-3}
6	5.8127×10^{-4}	1.00	1.00	8.6426×10^{-4}
7	5.6276×10^{-5}	1.00	1.00	2.5567×10^{-5}
8	1.4504×10^{-8}	1.00	1.00	6.9094×10^{-8}
9	1.3624×10^{-9}	1.00	1.00	5.1619×10^{-8}

The example is the iteration from Table 1 for $r = 2$

Table 3 Low tensor rank approximation of $\underline{\tilde{u}}_{15}$ from the model problem for $d = 10$, $R = 300$, $n = 1000$, and $\mathcal{E}_{15} = 5.981 \times 10^{-5}$

r	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r^{(0)}\|}{\|\underline{\tilde{u}}_L\|}$	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r\|}{\|\underline{\tilde{u}}_L\|}$	$\|f'(\underline{u}_r)\|$	RNM-iterations	CPU-time (s)
1	2.199×10^{-1}	1.861×10^{-1}	7.447×10^{-9}	10	0.04
2	1.346×10^{-1}	1.587×10^{-5}	1.917×10^{-9}	7	0.06
3	1.344×10^{-5}	1.241×10^{-5}	9.401×10^{-9}	35	0.68
4	1.101×10^{-5}	9.789×10^{-6}	9.799×10^{-9}	146	3.29

Table 4 Iteration of the RNM for $d = 10$, $R = 300$, $r = 4$, $n = 1000$, and $\mathcal{E}_{15} = 5.981 \times 10^{-5}$

k	$\left\|f'(\underline{u}_r^{(k)})\right\|$	ω_k	$\bar{\omega}_k$	$\frac{\left\|\underline{\tilde{u}}_{42} - \underline{u}_r^{(k)}\right\|}{\|\underline{\tilde{u}}_{42}\|}$
0	2.8259×10^{-8}	1.0	–	1.1014×10^{-5}
30	2.0969×10^{-8}	$(0.8)^3$	0.50	1.0490×10^{-5}
60	1.7256×10^{-8}	$(0.8)^4$	0.25	1.0217×10^{-5}
90	2.0321×10^{-8}	$(0.8)^3$	0.50	1.0029×10^{-5}
120	1.0128×10^{-8}	$(0.8)^4$	0.25	9.8876×10^{-6}
146	9.7994×10^{-9}	$(0.8)^5$	0.25	9.7889×10^{-6}

The example is the iteration from Table 3 for $r = 4$

Table 5 Low tensor rank approximation of $\underline{\tilde{u}}_{42}$ from the model problem for $d = 20$, $R = 1680$, $n = 1000$ and $\mathcal{E}_{42} = 8.789 \times 10^{-7}$

r	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r^{(0)}\|}{\|\underline{\tilde{u}}_L\|}$	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r\|}{\|\underline{\tilde{u}}_L\|}$	$\|f'(\underline{u}_r)\|$	RNM-iterations	CPU-time (s)
1	5.163×10^{-1}	1.990×10^{-1}	5.546×10^{-9}	10	1.07
2	9.536×10^{-1}	1.490×10^{-8}	8.088×10^{-9}	4	0.97

Table 6 Low tensor rank approximation of $\underline{\tilde{u}}_{15}$ from the model problem for $d = 20$, $R = 600$, $n = 1000$ and $\mathcal{E}_{15} = 1.125 \times 10^{-4}$

r	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r^{(0)}\|}{\|\underline{\tilde{u}}_L\|}$	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r\|}{\|\underline{\tilde{u}}_L\|}$	$\|f'(\underline{u}_r)\|$	RNM-iterations	CPU-time (s)
1	2.311×10^{-1}	1.990×10^{-1}	3.847×10^{-9}	9	0.24
2	2.107×10^{-1}	4.609×10^{-5}	3.548×10^{-9}	2	0.12
3	3.131×10^{-5}	2.384×10^{-5}	9.906×10^{-9}	84	6.9
4	1.597×10^{-5}	1.183×10^{-5}	9.836×10^{-9}	102	8.3

we need appreciably more iterations, at most 146. In these cases one cannot expect that the method converges superlinearly or even quadratically. In Table 4 the iterative behaviour of the regularized Newton method is documented for such a case. Typical

Table 7 Low tensor rank approximation of $\underline{\tilde{u}}_{42}$ from the model problem for $d = 50$, $R = 4200$, $n = 1000$ and $\mathcal{E}_{42} = 9.261 \times 10^{-7}$

r	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r^{(0)}\|}{\|\underline{\tilde{u}}_L\|}$	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r\|}{\|\underline{\tilde{u}}_L\|}$	$\|f'(\underline{u}_r)\|$	RNM-iterations	CPU-time (s)
1	9.144×10^{-1}	3.540×10^{-2}	1.051×10^{-8}	5	8.21
2	3.530×10^{-3}	1.125×10^{-8}	2.187×10^{-8}	2	9.13

Table 8 Low tensor rank approximation of $\underline{\tilde{u}}_{42}$ from the model problem for $d = 100$, $R = 8400$, $n = 1000$ and $\mathcal{E}_{42} = 2.013 \times 10^{-7}$

r	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r^{(0)}\|}{\|\underline{\tilde{u}}_L\|}$	$\frac{\|\underline{\tilde{u}}_L - \underline{u}_r\|}{\|\underline{\tilde{u}}_L\|}$	$\|f'(\underline{u}_r)\|$	RNM-iterations	CPU-time (s)
1	9.644×10^{-1}	1.271×10^{-3}	2.888×10^{-8}	3	83.23
2	6.308×10^{-6}	1.577×10^{-8}	7.925×10^{-11}	1	101.28

in this case is, that with our choice of the initial guess the norm of the gradient is very small for all iterates.

As already mentioned, Algorithm 4 is used in [4,5] for lower rank approximations of tensors from the Hartree–Fock equation. In [4,5] we made the same observation as for the model problem. As long as we approximate the given tensor with an approximation error which is of the same level as the underlying model accuracy, our method needs only few iterations and converges quadratically. If we approximate the original tensor below the accuracy introduced by the model discretisation, the behaviour of the iteration process changes. In these cases, the regularization of the system matrix from Eq. (49) is applied.

To complete the numerical experiments for the model problem, the results for $d \in \{20, 50, 100\}$ are documented in Tables 5, 6, 7 and 8.

7.2 Inexact iterations

In this section, we want to study the numerical behaviour of Algorithm 4 in the context of inexact iterations in the canonical tensor format. Let $\mathcal{T} := \bigotimes_{\mu=1}^{d} \mathbb{R}^n$ and $(x_k)_{k\in\mathbb{N}} \subset \mathcal{T}$ be a recursively defined sequence, i.e. there is $\Phi_k : \mathcal{T}_{r_k} \to \mathcal{T}_{r_{k+1}}$ and $x_0 \in \mathcal{T}_{r_0}$ with

$$x_k := \Phi_k(x_{k-1}), \quad \lim_{k\to\infty} x_k = x^*.$$

In practice, the sequence of tensor ranks $(r_k)_{k\in\mathbb{N}}$ grows that fast such that the iterative process cannot be performed on computer systems. Therefore, one introduces inexact iterations as defined below:

$$\begin{aligned} y_0 &:= \mathfrak{App}_{\varepsilon_0}(x_0), \\ z_k &:= \Phi_k(y_{k-1}), \\ y_k &:= \mathfrak{App}_{\varepsilon_k}(z_k), \end{aligned} \tag{67}$$

where $\mathfrak{App}_{\varepsilon_k}$ is defined in Notation 6.12. If one makes sure that $\|z_k - \mathfrak{App}_{\varepsilon_k}\| \leq c\|z_k - x^*\|$ and if $x_k \xrightarrow[k\to\infty]{} x^*$ at least quadratically, we have that $y_k \xrightarrow[k\to\infty]{} x^*$ with the same rate of convergence as $x_k \xrightarrow[k\to\infty]{} x^*$. For a complete convergence analysis of inexact iterations we refer to [10].

The computation of the pointwise inverse of a tensor $u \in \mathcal{T}$ with $u_{\underline{i}} \neq 0$ for all $\underline{i} \in \{1, \dots, n\}^d$ is of interest in several applications, e.g. for the pointwise $sign$-function of u, where the pointwise $sign$-function of u has important applications in the data analysis of high-order tensors. A naive computation of this problem would have a complexity which grows exponentially with the order d of the tensor u. If u is given in some efficient tensor format like the canonical tensor format, one can compute the pointwise inverse iteratively. For the pointwise inverse, the function Φ_k from Eq. (67) is defined as follows

$$x \mapsto \Phi_k(x) := x \odot (2\mathbb{1} - u \odot x), \tag{68}$$

where $\odot$ is the pointwise Hadamard product and $\mathbb{1}$ is the constant rank-one tensor with all values equal to one. The recursion function Φ is motivated by the Newton method applied to the function $\tau \mapsto a - \tau^{-1}$, see [10] for more details. If one defines the error $e_k := \mathbb{1} - u \odot x_k$, we have

$$\begin{aligned} e_k &= \mathbb{1} - ux_k = \mathbb{1} - ux_{k-1}\left(\mathbb{1} + e_{k-1}\right) = e_{k-1} - ux_{k-1}e_{k-1} \\ &= \left(\mathbb{1} - ux_{k-1}\right) e_{k-1} = e_{k-1}^2 = e_0^{2^k}, \end{aligned}$$

where we set for simplicity $ab := a \odot b$ for any $a, b \in \mathcal{T}$. For $\|e_0\| < 1$ it follows that $x_k \xrightarrow[k\to\infty]{} u^{-1}$, where u^{-1} is defined pointwise. For the numerical test we used the following two function-generated tensors u_1 and u_2 defined by the evaluation of the functions

$$\varphi_1 : [0,1]^d \to \mathbb{R}, \quad \underline{x} \mapsto \varphi_1(\underline{x}) := 1 + \sum_{l=1}^{2} \prod_{\mu=1}^{d} (x_\mu)^{\frac{l}{d}},$$

$$\varphi_2 : [0,1]^d \to \mathbb{R}, \quad \underline{x} \mapsto \varphi_1(\underline{x}) := 1 + \frac{9}{d} \sum_{\mu=1}^{d} x_\mu$$

on the uniform grid

$$\Gamma_n := \left\{ \eta(i_1 - 1, i_2 - 1, \dots, i_d - 1)^t \in [0,1]^d : (i_1, \dots, i_d) \in \{1, \dots, n\}^d \right\}, \\ \eta := 1/(n-1).$$

The approximation error is determined by $\varrho_i := \frac{\|\mathbb{1}-u_i y\|}{\|\mathbb{1}\|}$ for $i \in \{1, 2\}$. The computation of an initial guess $y_0^{(i)}$ with $\varrho_i(y_0^{(i)}) < 1$ is necessary for the convergence of the method. In the present case it is sufficient to find a locally best rank-one approximation of u_i and to compute the pointwise inverse of this rank-one approximation. The pointwise inverse of an elementary tensor $v = \bigotimes_{\mu=1}^{d} v_\mu$ is easy to compute, since we have $v^{-1} = \bigotimes_{\mu=1}^{d} v_\mu^{-1}$. In inexact iterations as described in (67), one can expect that the predecessor y_{k-1} is a good initial guess for the low rank approximation of z_k. Therefore, we choose y_{k-1} as the initial guess for the low rank approximation of z_k in Algorithm 4. The numerical tests are documented in Tables 9, 10 and 11 for $d \in \{100, 150\}$ and $n := 100$. The proposed Algorithm 1 needs only few iterations in order to compute a local best approximation of z_k in all examples.

7.3 Comparison of the descent directions

In regularized Newton method, the standard choice for the system matrix method is

$$\begin{aligned}\hat{H}^{(1)}(\hat{\xi}_k, \omega_k) &:= (1-\omega_k)A_k(\hat{\xi}_k) + \omega_k f''(\hat{\xi}_k)\\ &= A_k(\hat{\xi}_k) + \omega_k \left(B_k(\hat{\xi}_k)+C_k(\hat{\xi}_k)-D_k(\hat{\xi}_k)+\lambda_1 G_{1k}(\hat{\xi}_k)+\lambda_2 G_{2k}(\hat{\xi}_k)\right).\end{aligned}$$

In Eq. (53) we have introduced the following new system matrix

Table 9 Computation of u_1^{-1} with $d = 100$, $n = 100$, and $\varrho_1(y_{(3)}) = 2.137 \times 10^{-6}$

k	$\operatorname{rank}_{\mathcal{T}}(z_k)$	$\operatorname{rank}_{\mathcal{T}}(y_{r_k})$	$\frac{\left\|z_k-y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\left\|z_k-y_{r_k}\right\|}{\|z_k\|}$	$\left\|f'(y_{r_k})\right\|$	RNM-iterations	CPU-time (s)
1	102	1	6.63×10^{-6}	6.63×10^{-6}	1.15×10^{-7}	1	0.33
1	102	2	5.61×10^{-6}	4.43×10^{-6}	3.54×10^{-7}	25	11.05
2	406	2	4.43×10^{-6}	4.17×10^{-6}	4.50×10^{-7}	21	23.28
2	406	3	2.14×10^{-6}	2.13×10^{-6}	3.44×10^{-7}	2	6.07

Table 10 Computation of u_1^{-1} with $d = 150$, $n = 100$, and $\varrho_1(y_{(2)}) = 3.141 \times 10^{-6}$

k	$\operatorname{rank}_{\mathcal{T}}(z_k)$	$\operatorname{rank}_{\mathcal{T}}(y_{r_k})$	$\frac{\left\|z_k-y_{r_k}^{(0)}\right\|}{\|z_k\|}$	$\frac{\left\|z_k-y_{r_k}\right\|}{\|z_k\|}$	$\left\|f'(y_{r_k})\right\|$	RNM-iterations	CPU-time (s)
1	152	1	4.44×10^{-6}	4.44×10^{-6}	4.26×10^{-7}	1	0.81
1	152	2	3.77×10^{-6}	3.15×10^{-6}	3.79×10^{-7}	11	20.08

Table 11 Computation of u_2^{-1} with $d = 150$, $n = 100$, and $\varrho_2(y_{(3)}) = 1.384 \times 10^{-6}$

k	$\text{rank}_{\mathcal{T}}(z_k)$	$\text{rank}_{\mathcal{T}}(y_{r_k})$	$\frac{\left\lVert z_k - y_{r_k}^{(0)} \right\rVert}{\lVert z_k \rVert}$	$\frac{\left\lVert z_k - y_{r_k} \right\rVert}{\lVert z_k \rVert}$	$\lVert f'(y_{r_k}) \rVert$	RNM-iterations	CPU-time (s)
1	4	1	1.66×10^{-3}	1.66×10^{-3}	1.02×10^{-7}	1	0.04
1	4	2	9.33×10^{-4}	2.34×10^{-6}	2.93×10^{-7}	5	2.03
2	14	2	3.72×10^{-6}	1.88×10^{-6}	1.86×10^{-7}	2	1.12
3	14	2	1.89×10^{-6}	1.89×10^{-6}	2.96×10^{-7}	2	0.51
3	14	3	1.57×10^{-6}	1.41×10^{-6}	2.33×10^{-7}	2	2.23

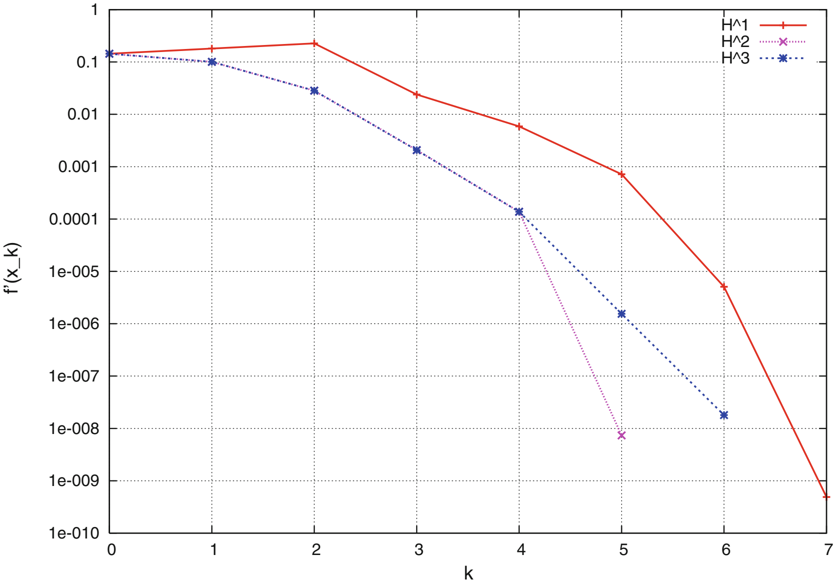

Fig. 1 Comparison of the descent directions on the model problem from Sect. 7.1 with $d = 25$, $r = 2$, $R = 2100$ and $n = 1000$

$$\hat{H}^{(2)}(\hat{\xi}_k, \omega_k) := \begin{cases} A_k(\hat{\xi}_k) + \omega_k \left(B_k(\hat{\xi}_k) + \lambda_1 G_{1k}(\hat{\xi}_k) + \lambda_2 G_{2k}(\hat{\xi}_k) \right), & \overline{\omega}_k \neq 1 \\ A_k(\hat{\xi}_k) + \omega_k \left(B_k(\hat{\xi}_k) + C_k(\hat{\xi}_k) - D_k(\hat{\xi}_k) \right. & \overline{\omega}_k = 1. \\ \quad \left. + \lambda_1 G_{1k}(\hat{\xi}_k) + \lambda_2 G_{2k}(\hat{\xi}_k) \right), & \end{cases}$$

The method with $\hat{H}^{(2)}(\hat{\xi}_k, \omega_k)$ as system matrix is a combination of $\hat{H}^{(1)}(\hat{\xi}_k, \omega_k)$ for $\overline{\omega}_k \neq 1$ and $\hat{H}^{(3)}(\hat{\xi}_k, \omega_k)$ for $\overline{\omega}_k = 1$, where

$$\hat{H}^{(3)}(\hat{\xi}_k, \omega_k) := A_k(\hat{\xi}_k) + \omega_k \left(B_k(\hat{\xi}_k) + C_k(\hat{\xi}_k) - D_k(\hat{\xi}_k) + \lambda_1 G_{1k}(\hat{\xi}_k) + \lambda_2 G_{2k}(\hat{\xi}_k) \right).$$

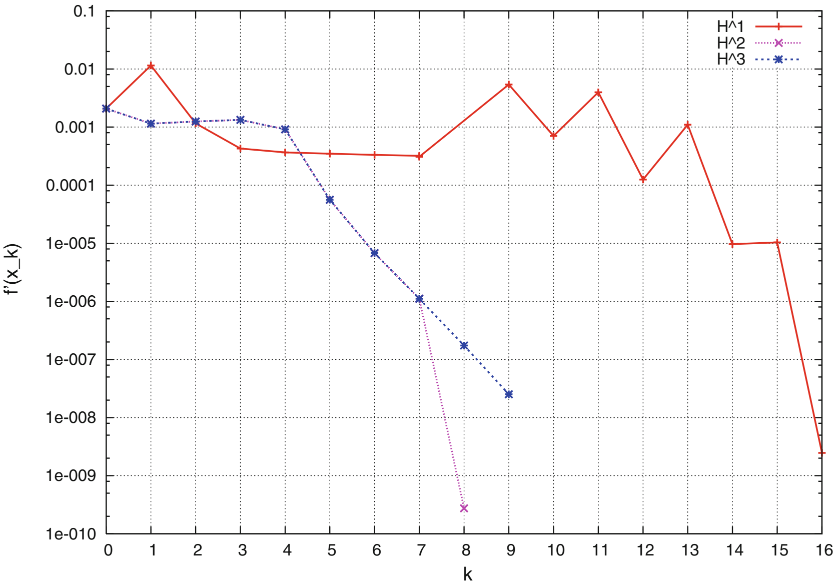

Fig. 2 Comparison of the descent directions on example u_1^{-1} from Sect. 7.2 with $d = 50$, $r = 2$, $R = 462$ and $n = 100$

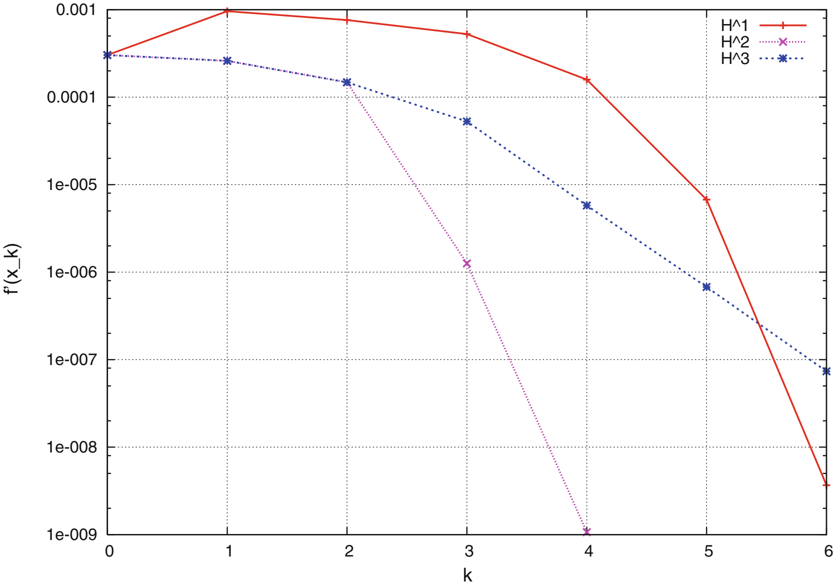

Fig. 3 Comparison of the descent directions on example u_2^{-1} from Sect. 7.2 with $d = 50$, $r = 4$, $R = 52$ and $n = 100$

These three different choices lead to different minimization methods. We want to compare the iterative behaviour of the different methods applied to randomly chosen examples from the previous sections. In Figs. 1, 2 and 3 the iteration process of the regularized Newton method is plotted. In all calculations the descent direction with $\hat{H}^{(2)}(\hat{\xi}_k, \omega_k)$ leads to the best result. Note that this choice leads to an even better complexity for the computation of the descent direction, see Lemma 6.4.

A Appendix

Lemma A.1 *Let $a \in \mathbb{R}, 0 \neq b \in \mathbb{R}^n, x \in \mathbb{R}^n$, $A, B \in \mathbb{R}^{n \times n}$ be positive definite, symmetric and*

$$d \mapsto q(d) := a + \langle b, d - x\rangle + \frac{1}{2}\langle B(d-x), d-x\rangle \tag{69}$$

for all $d \in \mathbb{R}^n$. For every $r \in \left(0, \|B^{-1}b\|_A\right)$ there exist a unique minimizer d^* of q in $\overline{K}_A(x, r) := \{d \in \mathbb{R}^n : \|d - x\|_A \leq r\}$. Further we have

$$d^* = x - \left(\lambda^* A + B\right)^{-1} b, \tag{70}$$

where $\lambda^* > 0$ is uniquely determined as the solution of the non-linear equation

$$\varphi(\lambda^*) := \|(\lambda^* A + B)^{-1} b\|_A = r. \tag{71}$$

Proof Let $\lambda > 0, r \in \left(0, \|B^{-1}b\|_A\right)$ and $d \in \mathbb{R}^n$. The constraint $\|d - x\|_A^2 \leq r^2$ induces the Lagrange function

$$L_\lambda(d) := q(d) + \frac{1}{2}\lambda\left(\|d - x\|_A^2 - r^2\right).$$

Since $\lambda > 0$ and B is positive definite, L_λ is convex. Therefore, d^* is a minimizer of L_λ in $\mathbb{R}^n$ if and only if

$$\begin{aligned} 0 &= L'_\lambda(d^*) = q'(d^*) + \lambda A(d^* - x) \\ &= b + B(d^* - x) + \lambda A(d^* - x) \quad \Leftrightarrow \quad d^* = x - (\lambda A + B)^{-1} b. \end{aligned}$$

Since $\varphi(\lambda) \xrightarrow[\lambda \to 0]{} \|B^{-1}b\|_A$ and $\varphi(\lambda) \xrightarrow[\lambda \to \infty]{} 0$, there exists a solution $\lambda^* > 0$ of Eq. (71). It remains to show uniqueness. We have

$$\begin{aligned} \varphi^2(\lambda) &= \|(\lambda A + B)^{-1} b\|_A^2 = \|A^{-\frac{1}{2}}(\lambda \mathbf{Id} + A^{-\frac{1}{2}} B A^{-\frac{1}{2}})^{-1} A^{-\frac{1}{2}} b\|_A^2 \\ &= \|(\lambda \mathbf{Id} + \tilde{B})^{-1} \tilde{b}\|^2, \end{aligned}$$

where we define $\tilde{B} := A^{-\frac{1}{2}} B A^{-\frac{1}{2}}$ and $\tilde{b} := A^{-\frac{1}{2}} b$. $\tilde{B}$ is symmetric and has only positive eigenvalues. Therefore $\tilde{B}$ is positive definite. Accordingly, we have

$$\varphi^2(\lambda) = \|U\left(\lambda \mathbf{Id} + D\right)^{-1} U^t \tilde{b}\|^2 = \|\left(\lambda \mathbf{Id} + D\right)^{-1} \hat{b}\|^2 = \sum_{i=1}^{n} (\lambda + d_i)^{-2} \hat{b}_i^2,$$

where $\tilde{B} = UDU^t$ and $\hat{b} := U^t\tilde{b}$. Furthermore we have

$$(\varphi^2)'(\lambda) = -2\sum_{i=1}^{n} \underbrace{(\lambda + d_i)^{-3}}_{>0} \hat{b}_i^2 < 0,$$

since $\hat{b} \neq 0$ and $d_i > 0$, where $D = \operatorname{diag}(d_i)_{i=1}^n$. Therefore, φ is a strictly monotonically decreasing function, consequently injective. □

Lemma A.2 *Let $c > 0$ and $(\xi^k)_{k\in\mathbb{N}} \subset \mathcal{S}_r^c$ by a sequence of tensors in the canonical format with bounded terms and $\hat{A}_k := A(\hat{\xi}^k)$ as defined in Lemma 5.6. Furthermore, let $\operatorname{rank}_{\mathcal{T}}(\xi^k) = r$. We have:*

(i) *$\hat{A}_k$ is symmetric and positive definite, consequently regular.*

(ii)

$$\hat{A}_k^{-1} = \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \left[G_\mu^{(k)}\right]^{-1} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}, \tag{72}$$

where $(G_\mu^{(k)})_{j_1 j_2} := \left\langle \xi_{j_1}^k, \xi_{j_2}^k \right\rangle_\mu$ for $1 \leq j_1, j_2 \leq r$.

(iii) *There exists $M > 0$ such that for all $u \in \mathfrak{R}_{d,r,\underline{t}}$ and all $k \in \mathbb{N}$ we have the inequality*

$$\left\langle u, \hat{A}_k u \right\rangle \leq M\|u\|^2. \tag{73}$$

Proof (i) We have

$$\hat{A}_k^t = \sum_{\mu=1}^{d} \mathbb{E}_\mu^t \otimes G_\mu^{(k)^t} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}^t = \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes G_\mu^{(k)} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}} = \hat{A}_k.$$

$G_\mu^{(k)}$ is a Gram matrix and positive definite, see Lemma A.3. Consequently, $\hat{A}_k$ is a positive definite matrix.

(ii) Let $\widetilde{A}_k := \sum_{\mu=1}^d \mathbb{E}_\mu \otimes \left[G_\mu^{(k)}\right]^{-1} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}$. We have

$$\begin{aligned}\hat{A}_k \cdot \widetilde{A}_k &= \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \left(G_\mu^{(k)} \cdot \left[G_\mu^{(k)}\right]^{-1}\right) \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}} \\ &= \sum_{\mu=1}^{d} \mathbb{E}_\mu \otimes \mathbf{Id}_{\mathbb{R}^r} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}} = \mathbf{Id}_{\mathbb{R}^{r|\underline{t}|}}.\end{aligned}$$

(iii) Define $M := rc^{2\frac{d-1}{d}} > 0$ and let $k \in \mathbb{N}$ and $u \in \mathfrak{R}_{d,r,\underline{t}}$. The following sequence of inequalities

$$\left\langle u, \hat{A}_k u \right\rangle \leq \|\hat{A}_k\| \|u\|^2 \leq \max_{1\leq\mu\leq d} \|G_\mu^{(k)} \otimes \mathbf{Id}_{\mathbb{R}^{t_\mu}}\| \|u\|^2$$
$$\leq \max_{1\leq\mu\leq d} \|G_\mu^{(k)}\| \|u\|^2 \leq \max_{1\leq\mu\leq d} |\lambda_{\max}(G_\mu^{(k)})| \|u\|^2$$

is valid. Further, we have for $\xi^k \in \mathcal{S}_r^c$ that $\|\xi_j^k\|_\mu \leq c^{\frac{d-1}{d}}$ holds. From Gerschgorin's Theorem it follows that

$$|\lambda_{\max}(G_\mu^{(k)})| \in \bigcup_{j=1}^{r} \{\lambda \in \mathbb{R} : |\|\xi_j^k\|_\mu^2 - \lambda| \leq \rho_j\},$$

where

$$\rho_j := \sum_{j'=1,\, j'\neq j}^{r} \left|\left\langle \xi_j^k, \xi_{j'}^k \right\rangle_\mu\right| \leq \sum_{j'=1,\, j'\neq j}^{r} \|\xi_j^k\|_\mu \|\xi_{j'}^k\|_\mu \leq (r-1)c^{2\frac{d-1}{d}}.$$

It follows

$$|\lambda_{\max}(G_\mu^{(k)})| \leq rc^{2\frac{d-1}{d}} = M$$

and finally we have

$$\left\langle u, \hat{A}_k u \right\rangle \leq M\|u\|^2.$$

□

The following two Lemmata A.3 and A.4 are playing an important role in several direct minimization methods, e.g. the ALS method [1] and the regularized Newton method [7]. In particular, Lemma A.3 assures that the system matrix in the ALS method and the preconditioner in [7] are without loss of generality regular, see Lemma A.4.

Lemma A.3 *Let* $v := \sum_{i=1}^r v_i \in \mathcal{T}$ *with* $\operatorname{rank}_{\mathcal{T}}(v) = r \in \mathbb{N}$, *where* $v_i := \bigotimes_{\mu=1}^d v_{i\mu}$, $v_{i\mu} \in A_\mu \backslash \{0\}$. *Then we have that* $\{v_i^\mu : 1 \leq i \leq r\}$ *is linearly independent for all* $\mu \in \{1, \ldots, d\}$, *where we defined* $v_i^\mu := \bigotimes_{\nu=1,\, \nu\neq\mu}^d v_{i\nu}$.

Proof Assume that there is $1 \leq \mu_0 \leq d$ with $\{v_i^{\mu_0} : 1 \leq i \leq r\}$ linearly dependent. Then there are $\lambda_1, \ldots, \lambda_r \in \mathbb{R}$ and $1 \leq i_0 \leq r$ with $\lambda_{i_0} \neq 0$ and $\sum_{i=1}^r \lambda_i v_i^{\mu_0} = 0$. Without loss of generality let $i_0 = r$. We have $v_r^{\mu_0} = \sum_{i=1}^{r-1} \underbrace{\frac{-\lambda_i}{\lambda_r}}_{\tilde{\lambda}_i :=} v_i^{\mu_0}$. Further, it follows that

$$\begin{aligned} v = \sum_{i=1}^{r} v_i &= \left(\sum_{i=1}^{r-1} v_i^{\mu_0}(v_{i\mu_0}) \right) + v_r^{\mu_0}(v_{r\mu_0}) \\ &= \sum_{i=1}^{r-1} \left(v_i^{\mu_0}(v_{i\mu_0}) + \tilde{\lambda}_i v_i^{\mu_0}(v_{r\mu_0}) \right) = \sum_{i=1}^{r-1} \tilde{v}_i, \end{aligned}$$

where $\tilde{v}_i := v_i^{\mu_0}(v_{i\mu_0} + \tilde{\lambda}_i v_{r\mu_0})$. But this contradicts the fact that $\operatorname{rank}_{\mathcal{T}}(v) = r$. □

The case discussed in Lemma A.3 is an idealized situation. In practice, the set $\{v_i^{\mu} : 1 \le i \le r\}$ is almost linearly dependent, i.e. some singular values of the Gram matrix $G_\mu := (\langle v_{i_1}, v_{i_2} \rangle_\mu)_{i_1,i_2=1}^{r}$ are small. Let $l \in \{1, \ldots, r\}$ and

$$G_\mu = \sum_{i=1}^{r} \sigma_{i,\mu} \, u_{i,\mu} u_{i,\mu}^t \tag{74}$$

be the singular value decomposition of G_μ. Since $u_l^{\mu} \neq 0_{\mathbb{R}^r}$ there is $\|u_{l,\mu}\|_\infty \neq 0$. Therefore, the following tensor v_l with $\operatorname{rank}_{\mathcal{T}}(v) \le r - 1$ is well defined:

$$v_l := \sum_{i=1,\, i \neq j}^{r} v_i^{\mu} \left(v_{i\mu} - \frac{[u_{l,\mu}]_i}{\|u_{l,\mu}\|_\infty} v_{j\mu} \right), \tag{75}$$

where $j := \operatorname{argmax}_{1 \le i \le r} |[u_{l,\mu}]_i|$.

Lemma A.4 *Let $v := \sum_{i=1}^{r} \bigotimes_{\mu=1}^{d} v_{i\mu} \in \mathcal{T}$ and v_l as defined in Eq. (75). We have*

$$\|v - v_l\| \le \sigma_{l,\mu} \frac{\max_{1 \le j \le r} \|v_{j\mu}\|}{\|u_{l,\mu}\|_\infty}. \tag{76}$$

Proof We have

$$\begin{aligned} v - v_l &= \sum_{i=1}^{r} v_i^{\mu}(v_{i\mu}) - \sum_{i=1,\, i \neq j}^{r} v_i^{\mu} \left(v_{i\mu} - \frac{[u_{l,\mu}]_i}{\|u_{l,\mu}\|_\infty} v_{j\mu} \right) \\ &= v_j^{\mu}(v_{j\mu}) + \sum_{i=1,\, i \neq j}^{r} \frac{[u_{l,\mu}]_i}{\|u_{l,\mu}\|_\infty} v_i^{\mu} \left(v_{j\mu} \right) \\ &= \frac{1}{\|u_{l,\mu}\|_\infty} \underbrace{\left(\sum_{i=1}^{r} [u_{l,\mu}]_i v_{i\mu} \right)}_{f_l^{\mu} :=} \bigotimes v_{j\mu}. \end{aligned}$$

Furthermore, we have $\|f_l^{\mu}\|^2 = \langle u_{l,\mu}, G_\mu u_{l,\mu} \rangle = \sigma_{l,\mu}^2$. □

We discuss the computation of the factors $\langle\cdot,\cdot\rangle_{\mu_1\mu_2}$ and $\langle\cdot,\cdot\rangle_{\mu_1}$ introduced in Notation 5.4. A naive computation would have a complexity growing cubically with respect to the order d. But it is possible to compute the terms $\langle\cdot,\cdot\rangle_{\mu_1\mu_2}$ and $\langle\cdot,\cdot\rangle_{\mu_1}$ with a better complexity.

Let $\xi := \sum_{j=1}^{r} \bigotimes_{\mu=1}^{d} \xi_{j\mu}$ and $\alpha := \sum_{i=1}^{d} \bigotimes_{\mu=1}^{d} \alpha_{\mu i}$. First we compute and store all simple inner products, i.e. for $1 \le \mu \le d$, $1 \le j_1, j_2 \le r$ and $1 \le i \le R$ we compute $\left\langle \xi_{j_1\mu}, \xi_{j_2\mu}\right\rangle$ and $\left\langle \xi_{j_1\mu}, \alpha_{i\mu}\right\rangle$. The computational cost of this computation is $\mathcal{O}\left(r\cdot(r+R)\cdot\sum_{\mu=1}^{d} t_\mu\right)$. Obviously, we have

$$\begin{aligned}\left\langle \xi_{j_1}, \xi_{j_2}\right\rangle_{\mu_1\mu_2} &= \prod_{\mu\in\mathbb{N}_{\le d}\setminus\{\mu_1,\mu_2\}} \left\langle \xi_{j_1\mu}, \xi_{j_2\mu}\right\rangle \\ &= \prod_{\mu=1}^{\mu_1-1} \left\langle \xi_{j_1\mu}, \xi_{j_2\mu}\right\rangle \prod_{\mu=\mu_1+1}^{\mu_2-1} \left\langle \xi_{j_1\mu}, \xi_{j_2\mu}\right\rangle \prod_{\mu=(\mu_2+1)}^{d} \left\langle \xi_{j_1\mu}, \xi_{j_2\mu}\right\rangle\end{aligned}$$

for the terms from Notation 5.4 and

$$\begin{aligned}\left\langle \xi_{j_1}, \alpha_i\right\rangle_{\mu_1\mu_2} &= \prod_{\mu\in\mathbb{N}_{\le d}\setminus\{\mu_1,\mu_2\}} \left\langle \xi_{j_1\mu}, \alpha_{i\mu}\right\rangle \\ &= \prod_{\mu=1}^{\mu_1-1} \left\langle \xi_{j_1\mu}, \alpha_{i\mu}\right\rangle \prod_{\mu=\mu_1+1}^{\mu_2-1} \left\langle \xi_{j_1\mu}, \alpha_{i\mu}\right\rangle \prod_{\mu=\mu_2+1}^{d} \left\langle \xi_{j_1\mu}, \alpha_{i\mu}\right\rangle,\end{aligned}$$

where $\mu_1 \le \mu_2$. Note, because of symmetry it is sufficient to consider only this case. In order to calculate $\left\langle \xi_{j_1}, \xi_{j_2}\right\rangle_{\mu_1\mu_2}$ and $\left\langle \xi_{j_1}, \alpha_i\right\rangle_{\mu_1\mu_2}$, it is sufficient to compute the terms

$$X_{\mu_1,j_1,j_2}^{\mu_2} := \prod_{\mu=\mu_1}^{\mu_2} \left\langle \xi_{j_1\mu}, \xi_{j_2\mu}\right\rangle, \quad A_{\mu_1,j_1,i}^{\mu_2} := \prod_{\mu=\mu_1}^{\mu_2} \left\langle \xi_{j_1\mu}, \alpha_{i\mu}\right\rangle$$

since we have

$$\begin{aligned}\left\langle \xi_{j_1}, \xi_{j_2}\right\rangle_{\mu_1\mu_2} &= X_{1,j_1,j_2}^{(\mu_1-1)} \cdot X_{(\mu_1+1),j_1,j_2}^{(\mu_2-1)} \cdot X_{(\mu_2+1),j_1,j_2}^{d}, \\ \left\langle \xi_{j_1}, \alpha_i\right\rangle_{\mu_1\mu_2} &= A_{1,j_1,i}^{(\mu_1-1)} \cdot A_{(\mu_1+1),j_1,i}^{(\mu_2-1)} \cdot A_{(\mu_2+1),j_1,i}^{d}.\end{aligned}$$

The following Algorithm 5 computes the terms $X_{\mu_1,j_1,j_2}^{\mu_2}$ and $A_{\mu_1,j_1,i}^{\mu_2}$.

Corollary A.5 *The complexity of the computation of the terms* $\left\langle \xi_{j_1}, \xi_{j_2}\right\rangle_{\mu_1\mu_2}$ *and* $\left\langle \xi_{j_1}, \alpha_i\right\rangle_{\mu_1\mu_2}$ *from Notation 5.4 is*

$$\mathcal{O}\left(r\cdot(r+R)\cdot\left(d^2+\sum_{\mu=1}^{d} t_\mu\right)\right). \tag{77}$$

Algorithm 5 Computation of $X^{\mu_2}_{\mu_1,j_1,j_2}$ and $A^{\mu_2}_{\mu_1,j_1,i}$.

```
for $\mu_1 = 1$ to $d$ do
  for $\mu_2 = \mu_1$ to $d$ do
    for $j_1 = 1$ to $r$ do
      for $j_2 = 1$ to $r$ do
        if $\mu_1 = \mu_2$ then
          $X^{\mu_2}_{\mu_1,j_1,j_2} := \langle \xi_{j_1\mu_1}, \xi_{j_2\mu_1} \rangle$
        else
          $X^{\mu_2}_{\mu_1,j_1,j_2} := X^{(\mu_2-1)}_{\mu_1,j_1,j_2} \cdot \langle \xi_{j_1\mu_2}, \xi_{j_2\mu_2} \rangle$
        end if
      end for
      for $i = 1$ to $R$ do
        if $\mu_1 = \mu_2$ then
          $A^{\mu_2}_{\mu_1,j_1,i} := \langle \xi_{j_1\mu_1}, \alpha_{i\mu_1} \rangle$
        else
          $A^{\mu_2}_{\mu_1,j_1,i} := A^{(\mu_2-1)}_{\mu_1,j_1,i} \cdot \langle \xi_{j_1\mu_2}, \alpha_{i\mu_2} \rangle$
        end if
      end for
    end for
  end for
end for
```

References

1. Beylkin, G., Mohlenkamp, M.J.: Numerical operator calculus in higher dimensions. Proc. Natl. Acad. Sci. **99**(16), 10246–10251 (2002)
2. Beylkin, G., Mohlenkamp, M.J.: Algorithms for numerical analysis in high dimensions. SIAM J. Sci. Comput. **26**(6), 2133–2159 (2005)
3. Braess, D., Hackbusch, W.: Approximation of $1/x$ by exponential sums in $[1,\infty)$. IMA Numer. Anal. **25**, 685–697 (2005)
4. Chinnamsetty, S.R., Espig, M., Hackbusch, W., Flad, H.J.: Canonical tensor products as a generalization of gaussian-type orbitals. Int. J. Res. Phys. Chem. Chem. Phys. **224**(3/4), 681–694 (2010)
5. 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), 84–110 (2007)
6. de Silva, V., Lim, L.-H.: Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J. Matrix Anal. Appl. **30**, 1084–1127 (2008)
7. Espig, M.: Effiziente Bestapproximation mittels Summen von Elementartensoren in hohen Dimensionen. PhD thesis, Universität Leipzig (2008)
8. Greub, W.H.: Multilinear Algebra. Springer, Berlin (1967)
9. Hackbusch, W.: http://www.mis.mpg.de/scicomp/EXP_SUM
10. Hackbusch, W., Khoromskij, B.N., Tyrtyshnikov, E.E.: Approximate iterations for structured matrices. Numer. Math. **109**(3), 365–383 (2008)
11. Harshman, R.A.: Foundations of the PARAFAC procedure. UCLA Working Papers in Phonetics **16**, 1–84 (1970)
12. Kosmol, P.: Methoden zur numerischen Behandlung nichtlinearer Gleichungen und Optimierungsaufgaben. B.G. Teubner Stuttgart, 2. auflage edition (1993)
13. Ortega, J.M., Rheinboldt, W.C.: Iterative solution of nonlinear equations in several variables. Computer Science and Applied Mathematics. Academic Press (1970)
14. Oseledets, I.V., Savost'yanov, D.V.: Minimization methods for approximating tensors and their comparison. Comput. Math. Math. Phys. **46**(10), 1641–1650 (2006)
15. Paatero, P.: A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. Chemometr. Intell. Lab. Syst. **38**, 223–242 (1997)
16. Paatero, P.: The multilinear engine—a table-driven least squares program for solving multilinear problems, including the n-way parallel factor analysis model. J. Comput. Graph. Stat. **8**(4), 1–35 (1999)

17. Schwetlick, H.: Numerische Lösung nichtlinearer Gleichungen. VEB Deutscher Verlag der Wissenschaften (1979)
18. Zhang, T., Golub, G.H.: Rank-one approximation to high order tensors. SIAM J. Matrix Anal. Appl. **23**(2), 534–550 (2001)