Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Low Rank Tucker-Type Tensor Approximation to Classical Potentials

(revised version: October 2006)

by

Boris N. Khoromskij, and Venera Khoromskaia

Preprint no.: 105 2006

Low Rank Tucker-Type Tensor Approximation to Classical Potentials

B.N. Khoromskij and V. Khoromskaia
Max-Planck-Institute for Mathematics in the Sciences,
Inselstr. 22-26, D-04103 Leipzig, Germany.
{bokh, vekh}@mis.mpg.de

Abstract

This paper investigates best rank-$(r_1, ..., r_d)$ Tucker tensor approximation of higher-order tensors arising from the discretization of linear operators and functions in $\mathbb{R}^d$. Super-convergence of the Tucker decomposition with respect to the relative Frobenius norm is proven. Dimensionality reduction by the two-level Tucker-to-canonical approximation is discussed. Tensor-product representation of basic multi-linear algebra operations are considered, including inner, outer and Hadamard products. We also focus on fast convolution of higher-order tensors represented either by the Tucker or via the canonical models. Special versions of the orthogonal alternating least-squares (ALS) algorithm are implemented corresponding to the different formats of input data. We propose and test numerically the novel *mixed CT-model*, which is based on the additive splitting of a tensor as a sum of canonical and Tucker-type representations. This model allows to stabilise the ALS iteration in the case of "ill-conditioned" tensors.

The orthogonal Tucker decomposition is applied to 3D tensors generated by classical potentials, for example $\frac{1}{|x-y|}$, $e^{-\alpha|x-y|}$, $\frac{e^{-|x-y|}}{|x-y|}$ and $\frac{\text{erf}(|x|)}{|x|}$ with $x, y \in \mathbb{R}^d$. Numerical results for tri-linear decompositions illustrate exponential convergence in the Tucker rank, and robustness of the orthogonal ALS iteration.

AMS Subject Classification: 65F30, 65F50, 65N35, 65F10
Key words: Kronecker products, Tucker decomposition, multi-dimensional integral operators, multivariate functions, classical potentials.

1 Introduction

Numerical tensor decomposition methods designed initially for the problems in chemometrics and electronical engineering are becoming more and more attractive for application in large-scale numerical computations in higher dimensions [1, 4, 5, 6]. Indeed, numerical tensor-product decomposition gives the possibility to construct fast and economical algorithms with linear or even sub-linear scaling for computations involving large higher-order tensors. Typical examples are the equations of many-particle modelling for electronic structure calculations which evoke rigorous computations of multi-dimensional interactions via classical potentials. For these purposes, the key point is the efficient approximation of fully populated higher-order tensors representing multivariate functions and operators by using certain data-sparse Kronecker product structures.

Computational techniques of tensor decomposition can be understood as higher-order analogues of standard linear algebra methods for matrix-vector and matrix-matrix computations. In general, the efficient *multi-linear algebra* (MLA) tools cannot be derived via the straightforward extension of classical linear algebra. In fact, instead of "linear operations" such as SVD or EVD factorizations (finite algorithms), we arrive at challenging nonlinear optimisation problems.

In this paper we apply best rank-$(r_1, ..., r_d)$ approximation via the Tucker model which can be viewed as an extension of the best rank-r approximation to a matrix. This model allows dimensionality reduction via transformation of the initial large, higher-order tensor to a smaller *representation coefficients array*, i.e., core tensor, with respect to the problem-dependent orthogonal tensor-product basis, so-called Tucker factors or canonical factors. We also discuss the possible optimisation of the Tucker decomposition by applying the so-called CANDECOMP/PARAFAC (CP) model (cf. [3, 15]) to the core tensor (see Appendix for the definition). This model was first rigorously analysed in [20].

Numerical treatment of multi-dimensional operators/functions including tensor-tensor operations is usually limited by insufficient computational resources for the required computations. As a natural remedy, tensor-product representation provides the performance of standard tensor operations with asymptotically optimal complexity.

In Section 2 we estimate the complexity of tensor-product implementations of the inner, outer and contracted products as well as the Hadamard and convolution products of d-th order tensors. Then we discuss three versions of the orthogonal alternating least-squares algorithm (OALSA) to compute the best rank-$(r_1, ..., r_d)$ Tucker decomposition. The first one addresses decomposing the full-format tensor into the Tucker format (OALSA$(\mathcal{F} \to \mathcal{T}_{\mathbf{r}})$), see [5] for a detailed description of the algorithm. The second one works with the input data presented as canonical components in the CP model (OALSA$(\mathcal{C}_R \to \mathcal{T}_{\mathbf{r}})$) while the third algorithm applies to the input data presented in the Tucker format (OALSA$(\mathcal{T}_{\mathbf{R}} \to \mathcal{T}_{\mathbf{r}})$). The latter can be interpreted as rank reduction methods in the CP and Tucker models, respectively. We introduce the novel mixed CP-Tucker model which allows the stabilisation of the alternating least-squares iteration in the case of "ill-conditioned" target tensors (cf. §2.4.3).

Lemma 2.3 proves the super-convergence property of the Tucker model with respect to the Frobenius norm. To optimise the numerical operator calculus, we discuss different strategies to "compress" the core tensor in the Tucker model:
(A) Best N-term approximation of the core tensor via element-wise truncation;
(B) The CP decomposition of the core tensor (cf. the two-level Tucker model [18] and dimensionality reduction in [6]).

Lemma 2.4 shows that the CP decomposition of a tensor represented in the Tucker format can be reduced to the CP approximation of the corresponding "small size" core tensor. Corollary 2.5 describes an optimisation method for rank-reduction in the CP model.

In Section 3, the Tucker model is applied to classical potentials. We give a number of numerical examples illustrating efficiency of the orthogonal rank-$(r_1, ..., r_d)$ decomposition via OALSA applied to a class of tensors related to the Newton, Yukawa and Helmholtz potentials. Furthermore, we discuss the numerical Tucker decomposition of certain tensors related to the Hartree-Fock equation. Main observations from our numerics are the following:

- Exponential convergence of OALSA in the Tucker rank $r = \max_{\ell} r_\ell$ (cf. (2.13)).
- Quadratic convergence for the relative energy (cf. Lemma 2.4).
- Robust convergence of ALS iteration applied to the classical potentials.
- Efficient tensor operations in the Tucker/CP formats leading to asymptotically optimal multi-linear algebra (MLA), see §2.3.

In the Appendix we present auxiliary results describing the Lagrange equation for the dual maximisation problem, define the canonical decomposition and discuss quadratic convergence for the eigenvalues in the familiar Rayleigh quotient approximation (linear algebra analogue to Lemma 2.4).

2 Orthogonal Rank-$(r_1, ..., r_d)$ Tucker Decomposition

2.1 Preliminaries

We consider the linear space of real-valued d-th order tensors $\mathcal{A} = [a_{i_1 \ldots i_d}] \in \mathbb{R}^{\mathcal{I}}$, defined on the product index set $\mathcal{I} = I_1 \times ... \times I_d$. We make use of the Frobenius (L^2-energy) norm $||\mathcal{A}|| := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$ induced by the *inner product*

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{(i_1,...,i_d) \in \mathcal{I}} a_{i_1 \ldots i_d} b_{i_1 \ldots i_d} \quad \text{with } \mathcal{A}, \mathcal{B} \in \mathbb{R}^{\mathcal{I}}. \tag{2.1}$$

It corresponds to the Euclidean norm of a vector.

In the following the notation "$\times_\ell$" means the *outer product* of vectors which form the canonical (rank-1) tensor

$$\mathcal{U} \equiv \{u_{\mathbf{i}}\}_{\mathbf{i} \in \mathcal{I}} = b \cdot U^{(1)} \times_2 ... \times_d U^{(d)} \in \mathbb{R}^{\mathcal{I}}, \quad b \in \mathbb{R},$$

defined by the entries

$$u_{i_1 \ldots i_d} = b \cdot u^{(1)}_{i_1} \cdots u^{(d)}_{i_d} \quad \text{with} \quad U^{(\ell)} \equiv \{u^{(\ell)}_{i_\ell}\}_{i_\ell \in I_\ell} \in \mathbb{R}^{I_\ell}. \tag{2.2}$$

There is an alternative commonly used notation "$\circ$" for the *outer product* of vectors.

Given $\mathcal{A} \in \mathbb{R}^{\mathcal{I}}$, the rank-$(r_1, ..., r_d)$ Tucker model deals with the approximation over a sum of the rank-1 tensors

$$\mathcal{A}_{(\mathbf{r})} = \sum_{k_1=1}^{r_1} ... \sum_{k_d=1}^{r_d} b_{k_1 \ldots k_d} V^{(1)}_{k_1} \times_2 ... \times_d V^{(d)}_{k_d} \approx \mathcal{A}, \tag{2.3}$$

where the canonical factors $V^{(\ell)}_{k_\ell} \in \mathbb{R}^{I_\ell}$ ($k_\ell = 1, ..., r_\ell$, $\ell = 1, ..., d$) are (real-valued) vectors of the respective size $n_\ell = |I_\ell|$, $\mathbf{r} = (r_1, ..., r_d)$ (the Tucker rank) and $b_{k_1 \ldots k_d} \in \mathbb{R}$ (cf. Fig. 2.1 visualising (2.3) for $d = 3$). We further assume that the vectors $\{V^{(\ell)}_{k_\ell}\}$ are orthonormal, i.e.,

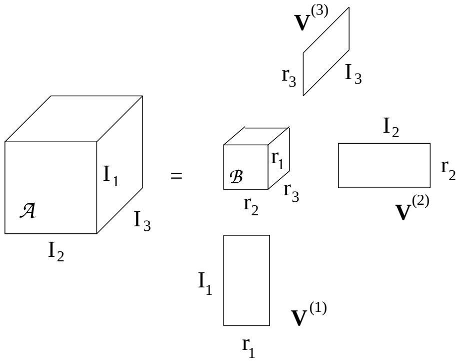

Figure 2.1: Visualisation of the Tucker model for a third-order tensor.

$$\left\langle V_{k_\ell}^{(\ell)}, V_{m_\ell}^{(\ell)} \right\rangle = \delta_{k_\ell,m_\ell}, \quad k_\ell, m_\ell = 1,...,r_\ell;\ \ell = 1,...,d, \tag{2.4}$$

where δ_{k_ℓ,m_ℓ} is Kronecker's delta (i.e., $\mathbf{V}^{(\ell)} = [V_1^{(\ell)} V_2^{(\ell)} ... V_{r_\ell}^{(\ell)}]$ is an orthogonal matrix, ${\mathbf{V}^{(\ell)}}^T \mathbf{V}^{(\ell)} = I$ for $\ell = 1,...,d$). In the following, we denote the set of tensors parametrised by (2.3), (2.4) by $\mathcal{T}_{(\mathbf{n},\mathbf{r})}$ (shortly $\mathcal{T}_{\mathbf{r}}$). For $\mathcal{A}_{(\mathbf{r})} \in \mathcal{T}_{\mathbf{r}}$, we use the conventional concise notation

$$\mathcal{A}_{(\mathbf{r})} = \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} ... \times_d \mathbf{V}^{(d)} \tag{2.5}$$

with the canonical components $\mathbf{V}^{(\ell)} \in \mathbb{R}^{I_\ell \times r_\ell}$ and with the core tensor $\mathcal{B} = \{b_{\mathbf{k}}\} \in \mathbb{R}^{r_1 \times ... \times r_d}$.

The so-called CANDECOMP/PARAFAC (CP) decomposition is a particular case of the Tucker model (2.3) corresponding to the choice $r = r_\ell$ ($\ell = 1,...,d$), where the only superdiagonal of $\mathcal{B}$ contains nonzero elements (see Appendix). We denote by $\mathcal{C}_{(\mathbf{n},r)}$ (shortly $\mathcal{C}_r$) the corresponding set of the dth order tensors. Notice that each tensor in $\mathcal{T}_{\mathbf{r}}$ can be interpreted (highly redundant representation) as an element in $\mathcal{C}_{r'}$ with $r' = r_1 \cdots r_d$, i.e. we have

$$\mathcal{T}_{\mathbf{r}} \subset \mathcal{C}_{r'}.$$

In some applications (say, in numerical calculus of multi-dimensional operators) both the Tucker and CP models can be combined gainfully (cf. the *two-level Tucker model* in [18] and a *dimensionality reduction* approach in calculation of the CP model [6]). Hence, in the following, we also discuss the two-level Tucker model $\mathcal{T}_{(\mathbf{n},\mathbf{r},q)}$, which contains all elements $\mathcal{A} \in \mathcal{T}_{(\mathbf{n},\mathbf{r})}$ such that for the corresponding core tensor we have $\mathcal{B} \in \mathcal{C}_{(\mathbf{r},q)}$. Clearly, we have

$$\mathcal{T}_{(\mathbf{n},\mathbf{r},q)} \subset \mathcal{C}_{(\mathbf{n},q)}.$$

Without loss of generality we can assume that $q \leq |\mathbf{r}| := \max\limits_{\ell} r_\ell$ (see the discussion in §2.3).

Remark 2.1 *Let $r = r_\ell$, $n = n_\ell$ ($\ell = 1,...,d$). Then the Tucker model requires only drn numbers to represent the canonical components plus r^d memory units for the core tensor. In general, the memory consumption is $\sum\limits_{\ell=1}^{d} r_\ell n_\ell + \prod\limits_{\ell=1}^{d} r_\ell$. Compared with the canonical representation in $\mathcal{K}_{\mathbf{n},r}$, the Tucker model requires an extra memory $r^d - r$ (our numerical results indicate that an additional cost to save the core tensor pays off here in full).*

On the other hand, the two-level Tucker model $\boldsymbol{\mathcal{T}}_{(\mathbf{n},\mathbf{r},q)}$ has the reduced memory consumption $dr(n+q)$ since, in general, $drq \ll r^d$, providing a good basis for various tensor-tensor operations (see §2.2).

2.2 Tensorisation of Basic MLA Operations

For the sake of clarity (and without loss of generality) in this section we consider the case $r = r_\ell$, $n = n_\ell$ $(\ell = 1, ..., d)$. If there is no confusion, we can skip the index $\mathbf{n}$. We denote by $\mathcal{N}$ the complexity of various tensor operations (say, $\mathcal{N}_{\langle\cdot,\cdot\rangle}$) or the related memory requirements (say, $\mathcal{N}_{mem(\mathcal{B})}$). We distinguish the following standard tensor-product operations: the inner product, the outer and the contracted products (cf. [1]) as well as the so-called Hadamard (component-wise) product.

To estimate the complexity of numerical decomposition in $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$ and related computations, we take more close look on the standard MLA operations in the Tucker format[1]:
(**I**) Memory demands;
(**II**) Frobenius and generalised L^2-inner product;
(**III**) Various tensor-times-tensor product operations including the Hadamard product;
(**IV**) Convolution of tensors.

Usually, the numerical Tucker decomposition leads to the fully populated core tensor, i.e., it is represented by r^d nonzero elements. However, in some cases a special data structure can be imposed (cf. [18]) which reduces the complexity of the corresponding MLA. In this discussion we assume one of the following situations:
(A) $\mathcal{B}$ is sparsely populated, more precisely, it has only $\mathcal{N}_{mem(\mathcal{B})} \leq r^d$ nonzero elements. Furthermore, we denote by $\mathcal{S}(\mathcal{B})$ a sparsity pattern of $\mathcal{B}$, such that $\#(\mathcal{S}(\mathcal{B})) = \mathcal{N}_{mem(\mathcal{B})}$;
(B) $\mathcal{B}$ has zero off-diagonal elements (i.e., $\mathbf{r} = (r, ..., r)$, $\mathcal{A} \in \boldsymbol{\mathcal{C}}_r$ and $\mathcal{N}_{mem(\mathcal{B})} = r$);
(C) $\mathcal{B}$ is represented in the canonical format, $\mathcal{B} \in \boldsymbol{\mathcal{C}}_q$ (two-level Tucker model $\boldsymbol{\mathcal{T}}_{(\mathbf{r},q)}$).

2.2.1 Memory Requirements

In the case (A), the Tucker model requires $drn + \mathcal{N}_{mem(\mathcal{B})}$ memory to represent a tensor, where $\mathcal{N}_{mem(\mathcal{B})} \leq r^d$. In turn, the CP model includes the reduced number of parameters $drn + r$ (case(B)). The difference $\mathcal{N}_{mem(\mathcal{B})} - r \leq r^d - r$ is negligible if d is not large (cf. Remark 2.1). In the case (C), the memory demands are $dr(n+q)$.

2.2.2 L^2- and Generalised L^2-Inner Product

For given tensors $\mathcal{A}_1 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}_1}, \mathcal{A}_2 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}_2}$ represented in the form (2.5), i.e.,

$$\mathcal{A}_1 = \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} ... \times_d \mathbf{U}^{(d)}, \quad \mathcal{A}_2 = \mathcal{C} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} ... \times_d \mathbf{V}^{(d)}, \tag{2.6}$$

the L^2-*inner product* (2.1) can be computed by

$$\langle \mathcal{A}_1, \mathcal{A}_2 \rangle := \sum_{\mathbf{k}=1}^{\mathbf{r}_1} \sum_{\mathbf{m}=1}^{\mathbf{r}_2} b_{k_1 \ldots k_d} c_{m_1 \ldots m_d} \prod_{\ell=1}^{d} \left\langle U^{(\ell)}_{k_\ell}, V^{(\ell)}_{m_\ell} \right\rangle. \tag{2.7}$$

[1] Complexity analysis of MLA in the CP model can be viewed as the particular case, corresponding to the class of tensors in $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$ with $\mathbf{r} = (r, ..., r)$ and with zero off-diagonal terms in the core tensor.

We further simplify and suppose $\mathbf{r}_1 = \mathbf{r}_2$. Then calculation in (2.7) includes $dr(r+1)/2$ inner products of vectors of size n (due to the symmetry argument) plus $2\#\mathcal{S}(\mathcal{A}_1) \cdot \#\mathcal{S}(\mathcal{A}_2)$ multiplications, leading to the overall complexity

$$\mathcal{N}_{\langle \cdot,\cdot \rangle} = O(dn\frac{r(r+1)}{2} + 2\#\mathcal{S}(\mathcal{A}_1) \cdot \#\mathcal{S}(\mathcal{A}_2)),$$

and the same for the Frobenius norm. In the case (C) the inner product can be computed in $Cq^2 + dr^2n + dq^2r$ operations (cf. [18], Lemma 2.8).

Let $\mathbf{A} : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{I}}$ be a matrix having Kronecker-product form: $\mathbf{A} = \mathbf{A}^{(1)} \otimes ... \otimes \mathbf{A}^{(d)} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ (rank-1 tensor of order d) with $\mathbf{A}^{(\ell)} \in \mathbb{R}^{n\times n}$. Then the generalised inner product

$$\langle \mathbf{A}\mathcal{A}_1, \mathcal{A}_2 \rangle := \sum_{\mathbf{k}=1}^{\mathbf{r}_1} \sum_{\mathbf{m}=1}^{\mathbf{r}_2} b_{k_1...k_d} c_{m_1...m_d} \prod_{\ell=1}^{d} \left\langle \mathbf{A}^{(\ell)} U_{k_\ell}^{(\ell)}, V_{m_\ell}^{(\ell)} \right\rangle \tag{2.8}$$

has the same computational cost as above, where n is substituted by the cost of the low-dimensional matrix-vector product in $\mathbb{R}^n$. For example, if the components $\mathbf{A}^{(\ell)} \in \mathbb{R}^{n\times n}$ have $\mathcal{H}$-matrix (resp. Toeplitz) structure or they inherit the wavelet-based sparsity pattern, then we have

$$\mathcal{N}_{\langle \mathbf{A}\cdot,\cdot \rangle} = O(d\frac{r(r+1)}{2} n \log^q n + 2\#\mathcal{S}(\mathcal{A}_1) \cdot \#\mathcal{S}(\mathcal{A}_2)).$$

2.2.3 Outer, Contracted and Hadamard Products

For given tensors $\mathcal{A} \in \mathbb{R}^{\mathcal{I}_1}$ and $\mathcal{B} \in \mathbb{R}^{\mathcal{I}_2}$, the *outer product* $\mathcal{A} \circ \mathcal{B} \in \mathbb{R}^{\mathcal{I}_1\times\mathcal{I}_2}$ is of size $\mathcal{I}_1 \times \mathcal{I}_2$ with the components

$$(\mathcal{A} \circ \mathcal{B})_{\mathbf{i}_1,\mathbf{i}_2} = \mathcal{A}_{\mathbf{i}_1} \cdot \mathcal{B}_{\mathbf{i}_2}, \quad \mathbf{i}_1 \in \mathcal{I}_1,\ \mathbf{i}_2 \in \mathcal{I}_2.$$

Clearly, for $\mathcal{A}_1, \mathcal{A}_2 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ we are able to tensorize the outer product by

$$\mathcal{A}_1 \circ \mathcal{A}_2 := \sum_{1\le k_\ell, m_\ell \le r,\ \ell=1,...,d} b_{k_1...k_d} c_{m_1...m_d} \left(U_{k_1}^{(1)} \circ V_{m_1}^{(1)} \right) \times_2 ... \times_d \left(U_{k_d}^{(d)} \circ V_{m_d}^{(d)} \right). \tag{2.9}$$

This leads to the memory demands

$$\mathcal{N}_{mem(\mathcal{A}\circ\mathcal{B})} = O(2dr^2n + r^{2d})$$

for the naive component-wise storage. However, the implicit representation of complexity $O(2drn + r^{2d})$ can be implemented if one stores only $2r$ components $U_{k_\ell}^{(\ell)}$, $V_{m_\ell}^{(\ell)}$ for each $\ell = 1, ..., d$.

The *contracted product* of two tensors is an extension of the matrix-vector multiplication combined with the outer product: for some portion of modes we compute the inner product while for the remaining components we calculate the outer product with the corresponding ordering of modes. For given tensors $\mathcal{A} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ and $\mathcal{B} \in \mathbb{R}^{\mathcal{I}\times\mathcal{M}}$, the contracted product along the first index $\mathcal{I}$ results in a tensor $\mathcal{Z} := \langle \mathcal{A}, \mathcal{B} \rangle_{\mathcal{I}\times\mathcal{I}}$ of size $\mathcal{J} \times \mathcal{M}$, given by

$$\mathcal{Z} = \{z_{\mathbf{jm}}\} \in \mathbb{R}^{\mathcal{J}\times\mathcal{M}} \quad \text{with} \quad z_{\mathbf{jm}} = \sum_{\mathbf{i}\in\mathcal{I}} a_{\mathbf{ij}} b_{\mathbf{im}}.$$

For tensors in the Tucker form we obtain

$$\langle \mathcal{A}_1, \mathcal{A}_2 \rangle_{\mathcal{I}\times\mathcal{I}} = \sum_{1\le k_\ell, m_\ell \le r,\ \ell=1,...,d} b_{k_1...k_d} c_{m_1...m_d} \left\langle U_{k_1}^{(1)}, V_{m_1}^{(1)} \right\rangle_{\mathcal{I}_1\times\mathcal{I}_1} \times_2 ... \times_d \left\langle U_{k_d}^{(d)}, V_{m_d}^{(d)} \right\rangle_{\mathcal{I}_d\times\mathcal{I}_d}$$

with the memory requirements

$$\mathcal{N}_{mem(\langle \mathcal{A}_1,\mathcal{A}_2\rangle_{\mathcal{I}\times\mathcal{I}})} = O(2dr^2 n_{\mathcal{J}} n_{\mathcal{M}} + r^{2d}).$$

For given tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{\mathcal{I}}$, the *Hadamard product* $\mathcal{A} \odot \mathcal{B} \in \mathbb{R}^{\mathcal{I}}$ of two tensors of the same size $\mathcal{I}$ is defined component-wise

$$(\mathcal{A} \odot \mathcal{B})_{\mathbf{i}} = \mathcal{A}_{\mathbf{i}} \cdot \mathcal{B}_{\mathbf{i}}, \quad \mathbf{i} \in \mathcal{I}.$$

Hence, for $\mathcal{A}_1, \mathcal{A}_2 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ we tensorize the Hadamard product by

$$\mathcal{A}_1 \odot \mathcal{A}_2 := \sum_{1\le k_\ell, m_\ell \le r,\ \ell=1,...,d} b_{k_1...k_d} c_{m_1...m_d} \left(U_{k_1}^{(1)} \odot V_{m_1}^{(1)} \right) \times_2 ... \times_d \left(U_{k_d}^{(d)} \odot V_{m_d}^{(d)} \right). \tag{2.10}$$

This leads to the memory requirements (due to the symmetry argument)

$$\mathcal{N}_{mem(\mathcal{A}\odot\mathcal{B})} = O(d\frac{r(r+1)}{2}n + r^{2d}).$$

2.2.4 Multi-Dimensional Convolution Product

The multi-dimensional convolution product is one of the basic transforms in the wide range of applications including many-particle models (e.g., see examples in §3). We consider the discrete version of the multi-dimensional convolution transform in $\mathbb{R}^d$ based on the Nyström type scheme (similar for the collocation with piecewise constant basis functions)

$$(f * g)(x) := \int_{\mathbb{R}^d} f(y) g(x-y) dy \approx h^d \sum_{\mathbf{i}\in\mathcal{I}} f(y_{\mathbf{i}}) g(x_{\mathbf{j}} - y_{\mathbf{i}}), \quad \mathcal{I} := \{1, ..., n\}^d,$$

where, for the ease of presentation, the collocation points $x_{\mathbf{j}}, y_{\mathbf{i}}$ are assumed to be located on the same equi-distant spatial tensor-product grid of size h. The functions f, g are supposed to have the finite support $[0, R]^d$ with $R = nh$.

Introducing the corresponding function generated tensors (FGTs) $\mathcal{F} = \{f(x_{\mathbf{i}})\}, \mathcal{G} = \{g(x_{\mathbf{i}})\} \in \mathbb{R}^{\mathcal{I}}$, we define their *discrete* convolution product by

$$\mathcal{F} * \mathcal{G} := \left(\sum_{\mathbf{k}\in\mathcal{I}} \mathcal{F}_{\mathbf{k}} \mathcal{G}_{\mathbf{j}-\mathbf{k}} \right)_{\mathbf{j}\in\mathcal{J}}, \quad \mathcal{J} := \{1, ..., 2n-1\}^d.$$

For given $\mathcal{A}_1, \mathcal{A}_2 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$, we now tensorize the convolution product via

$$\mathcal{A}_1 * \mathcal{A}_2 := h^d \sum_{\mathbf{k},\mathbf{m}=1}^{\mathbf{r}} b_{k_1...k_d} c_{m_1...m_d} \left(U_{k_1}^{(1)} * V_{m_1}^{(1)} \right) \times_2 ... \times_d \left(U_{k_d}^{(d)} * V_{m_d}^{(d)} \right). \tag{2.11}$$

Assuming that one-dimensional convolutions $U^{(\ell)}_{k_\ell} * V^{(\ell)}_{m_\ell} \in \mathbb{R}^{2n-1}$ can be computed in $O(n \log^q n)$ operations, we arrive at the overall complexity estimate

$$\mathcal{N}_{\cdot * \cdot} = O(d\frac{r(r+1)}{2} n \log^q n + d n^d r^{2d}).$$

In our particular case of equidistant grids we obtain (by setting $a = U^{(\ell)}_{k_\ell}, b = V^{(\ell)}_{m_\ell} \in \mathbb{R}^n$)

$$(U^{(\ell)}_{k_\ell} * V^{(\ell)}_{m_\ell})_j = \sum_{k=1}^{n} a_k b_{j-k}, \quad j = 1, ..., 2n-1.$$

Hence, the one-dimensional convolution can be performed by FFT in $O(n \log n)$ operations. One-dimensional $O(n \log n)$-complexity convolution on non-equidistant grids is discussed in [10]. It can be directly applied in (2.11).

We notice that the convolution product appears to be one of the most computationally elaborate operations (cf. Lemma 2.3) since in general one might have $\#\mathcal{S}(\mathcal{B}_1) \cdot \#\mathcal{S}(\mathcal{B}_2) = r^{2d}$. Significant complexity reduction is observed if one of the convolving tensors can be represented by the CP model, where we have $\#\mathcal{S}(\mathcal{B}_1) \cdot \#\mathcal{S}(\mathcal{B}_2) = r^{d+1}$. Hence, the complexity reduction factor is r^{d-1}.

Below, we give numerical examples illustrating the performance of the convolution product in the Tucker format. We use the following (rather simple) algorithm: Given the function-generated tensors (FGTs) $\mathcal{A}_1 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}_1}$, $\mathcal{A}_2 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}_2}$ with the moderate grid-size $n_1 = n_2 = ... = n_d = n$ corresponding to the tensor-product $n \times ... \times n$ grid in $[0, A]^d$ with arbitrary grid-spacing in each spatial direction (say, adaptive grid) and with minimal grid-size h_{min}. We introduce an auxiliary equidistant grid of size $N \times ... \times N$ with $n \ll N$ which satifies $h := A/N \leq h_{min}$ due to the approximation requirements. Let $P_{n \to N}$ and $P_{N \to n}$ be the corresponding 1D linear interpolation operators from the adaptive to fine grid and vice-versa, and in each spatial direction. Then we perform the following steps:

CONV ($\boldsymbol{\mathcal{T}}_{\mathbf{r}_1}$, $\boldsymbol{\mathcal{T}}_{\mathbf{r}_2}$).

Given the input tensors $\mathcal{A}_1 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}_1}$, $\mathcal{A}_2 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}_2}$.
Step I: Using $P_{n \to N}$, interpolate the canonical components to the fine grid;
Step II: Compute one-dimensional convolution products $h\, U^{(\ell)}_{k_\ell} * V^{(\ell)}_{m_\ell} \in \mathbb{R}^{2N-1}$ in $O(dr^2 N \log N)$ operations by the FFT;
Step III: Using $P_{N \to n}$, interpolate the result to the initial grid;
Step IV: Compute the convolution product via (2.11) in $O(dr^{2d}n^d)$ operations.

Remark 2.2 *Algorithm* **CONV** *($\boldsymbol{\mathcal{T}}_{\mathbf{r}_1}$, $\boldsymbol{\mathcal{T}}_{\mathbf{r}_2}$) has linear scaling in dimension d. However, in the case of strong mesh refinement the auxiliary dimension N may be large enough (say, $N \geq 10^4$), hence the complexity of Step II, $O(dr^2 N \log N)$, might be the bottleneck of our numerical scheme. For such situations, based on the ideas in [10], one can use the special modification of the convolution of complexity $O(n \log^q n)$ that applies to composite refined grids. The corresponding numerical results will be presented elsewhere.*

First, we demonstrate the reduction factor r^{d-1} between the computational times t_{TT} and t_{CT}, corresponding to the cases of Tucker-Tucker and CP-Tucker convolving tensors, respectively. In particular, the next table represents the ratios $t_{CT}/t_{CT}(2)$ and $t_{TT}/t_{CT}(2)$ (scaled time) for different values of the Tucker rank $r = 2, 3, ..., 6$, but with fixed parameters $n = 32, 64$ and $N = 320, 640$, respectively. Here $t_{CT}(2)$ referes to the computational time-unit for the CT-type convolution with $r = 2$ and with $n = 32$.

r		2	3	4	5	6	7	8	9	10
CT	$n = 32$	1	4.67	7.41	15.78	30.00	52.22	85.19	131.48	194.81
	$n = 64$	6.93	22.22	57.41	122.96	234.44	410.37	668.17	-	-
TT	$n = 32$	3.52	25.19	115.19	392.22	1085.19	-	-	-	-

Note that the computational time in the full format is about $T_1 = 0.02, T_2 = 1.11, T_3 = 68.4$ for $n = 16, 32, 64$, respectively, all presented in computational time-unit $t_{CT}(10)$ with $n = 32$ (the asymptotical complexity estimate is $O(8n^6)$).

The second table represents the scaled computational time $t_{CT}/t_{CT}(1)$ for fixed $r = 5$ and fixed $n = 32, 64$ but for different values of $N = 2^p n$ $(p = 1, 2, ..., 8)$. Here $t_{CT}(1)$ corresponds to the computational time for $p = 1$, $n = 32$.

p		1	2	3	4	5	6	7	8
CT	$n = 32$	1	1	1	1	1.13	1.85	4.59	15.46
	$n = 64$	7.13	7.13	7.20	7.17	8.07	10.87	21.70	-

These data indicate that FFT on the equi-distant fine grid has negligible cost compared with computing the sum (2.11), at least in the parameter domain $N \leq 4096$. Hence, Algorithm **CONV** $(\boldsymbol{\mathcal{T}}_{\mathbf{r}_1}, \boldsymbol{\mathcal{T}}_{\mathbf{r}_2})$ can be applied successfully in the case of moderate mesh-refinement such that $N \leq 10^4$.

2.2.5 Resumé of the Complexity of MLA in the Tucker Model

The next Lemma collects all the previous results but now presented in the general case of a fixed sparsity pattern of the target core tensors.

Lemma 2.3 *(complexity of MLA in the Tucker model). For given tensors $\mathcal{A}_1, \mathcal{A}_2 \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ with fixed sparsity patterns of the core tensors $\mathcal{S}(\mathcal{B}_1), \mathcal{S}(\mathcal{B}_2)$, respectively, we have*
*(**I**) Memory requirements $\mathcal{N}_{mem(\mathcal{A}_1)} = dnr + \#(\mathcal{S}(\mathcal{B}_1))$.*
*(**II**) Complexity of L^2-inner product*

$$\mathcal{N}_{\langle \mathcal{A}_1, \mathcal{A}_2 \rangle} = \frac{1}{2} dnr(r+1) + 2\#\mathcal{S}(\mathcal{B}_1) \cdot \#\mathcal{S}(\mathcal{B}_2),$$

and the same for the Frobenius norm.
*(**III**) Memory requirements for the outer and Hadamard products are given by*

$$\mathcal{N}_{mem(\mathcal{A}_1 \circ \mathcal{A}_2)} = 2dr^2 n + \#\mathcal{S}(\mathcal{B}_1) \cdot \#\mathcal{S}(\mathcal{B}_2),$$

$$\mathcal{N}_{mem(\mathcal{A}_1 \odot \mathcal{A}_2)} = \frac{1}{2} dr(r+1)n + \#\mathcal{S}(\mathcal{B}_1) \cdot \#\mathcal{S}(\mathcal{B}_2)),$$

respectively. Complexity estimates for these operations have similar bounds.
(**IV**) *Assuming that one-dimensional convolution products $U^{(\ell)}_{k_\ell} * V^{(\ell)}_{m_\ell} \in \mathbb{R}^{2n-1}$ can be computed in $O(n \log^q n)$ operations, we obtain the complexity bound*

$$\mathcal{N}_{\mathcal{A}_1 * \mathcal{A}_2} = \frac{1}{2} dr(r+1) n \log^q n + dn^d \# \mathcal{S}(\mathcal{B}_1) \cdot \# \mathcal{S}(\mathcal{B}_2).$$

Complexity of the convolution product in the full format is estimated by $O(n^{2d})$.

Proof. The proof is elementary and it goes along the line of the previous discussion. ■

2.3 Some Properties of the Orthogonal Tucker Decomposition

The numerical Tucker approximation of d-th order tensors is one of the most practically important MLA operation. This operation is, in fact, the higher order extension of the best rank-r approximation in the linear algebra (in particular, of the truncated SVD).

Given $\mathcal{A}_0 \in \mathbb{R}^{I_1 \times \ldots \times I_d}$, its Tucker approximation can be derived by straightforward minimisation of the quadratic cost functional

$$f(\mathcal{A}) := ||\mathcal{A} - \mathcal{A}_0||^2 \to min \tag{2.12}$$

over all rank-$\mathbf{r}$ tensors $\mathcal{A} \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$, which will be parametrised as in (2.3) and with the constraints

$$\mathbf{V}^{(\ell)} \in \mathcal{V}_\ell := \mathcal{V}_{n_\ell, r_\ell} \quad (\ell = 1, ..., d),$$

where $\mathcal{V}_{n,r} := \{Y \in \mathbb{R}^{n \times r} : Y^T Y = I \in \mathbb{R}^{r \times r}\}$ is the so-called Stiefel manifold. The Appendix discusses the existence of local minima in (2.12) and describes the Lagrange equations for the corresponding dual problem (cf. Lemma 4.2). In general, the starting value in the minimisation process for solving (2.12) can be computed using the so-called higher-order SVD [4].

For a wide class of FGTs, the quality of approximation via minimisation (2.12) can be effectively controlled by the Tucker rank. In particular, for certain analytic generating functions we are able to prove the exponential convergence (cf. [11, 18]),

$$||\mathcal{A}_{(\mathbf{r})} - \mathcal{A}_0|| \le Ce^{-\alpha r} \quad \text{with} \quad r = \max_\ell r_\ell. \tag{2.13}$$

As a consequence, the approximation error $\varepsilon > 0$ can be achieved with $r = O(|\log \varepsilon|)$.

The following Lemma proves that the relative energy error of the Tucker decomposition $\mathcal{A}_{(\mathbf{r})}$ is estimated by the square of the relative energy norm of $\mathcal{A}_{(\mathbf{r})} - \mathcal{A}_0$. This result is a reminiscence of the error bound for the Rayleigh quotient approximation to the symmetric eigen-value problem in linear algebra (cf. (4.9) in Appendix).

Lemma 2.4 *(super-convergence in energy). Let $\mathcal{A}_{(\mathbf{r})} \in \mathbb{R}^{I_1 \times \ldots \times I_d}$ solve the minimisation problem (2.12) over $\mathcal{A} \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$. Then we have the "quadratic" relative error bound*

$$\frac{||\mathcal{A}_0|| - ||\mathcal{A}_{(\mathbf{r})}||}{||\mathcal{A}_0||} \le \frac{||\mathcal{A}_{(\mathbf{r})} - \mathcal{A}_0||^2}{||\mathcal{A}_0||^2}. \tag{2.14}$$

Proof. First part of the proof is given for the completeness (cf. [4] for a short exposition). Letting $\mathcal{A}_{(\mathbf{r})} = \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \ldots \times_d \mathbf{V}^{(d)}$, we easily obtain the identity

$$||\mathcal{A}_{(\mathbf{r})}|| = ||\mathcal{B}||, \tag{2.15}$$

since orthogonal components $\mathbf{V}^{(\ell)} \in \mathcal{V}_\ell$ do not effect the Frobenius norm. Furthermore, with fixed $\mathbf{V}^{(\ell)}$ $(\ell = 1, ..., d)$, relation (2.12) is merely a linear least-square problem with respect to $\mathcal{B}$,

$$\langle \mathcal{A}_0, \mathcal{A}_0 \rangle - 2\langle \mathcal{A}_0, \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)} \rangle + \langle \mathcal{B}, \mathcal{B} \rangle \to \min,$$

hence the corresponding Lagrange equation takes the form

$$-\langle \mathcal{A}_0, \delta\mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)} \rangle + \langle \mathcal{B}, \delta\mathcal{B} \rangle = 0 \quad \forall \;\; \delta\mathcal{B} \in \mathbb{R}^{r_1 \times \ldots \times r_d},$$

which implies

$$\mathcal{B} - \mathcal{A}_0 \times_1 \mathbf{V}^{(1)^T} \times_2 \ldots \times_d \mathbf{V}^{(d)^T} = 0. \tag{2.16}$$

Next we readily obtain

$$\begin{aligned} f(\mathcal{A}_\mathbf{r}) &= ||\mathcal{A}_\mathbf{r}||^2 - 2\langle \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)}, \mathcal{A}_0 \rangle + ||\mathcal{A}_0||^2 \\ &= ||\mathcal{A}_\mathbf{r}||^2 + ||\mathcal{A}_0||^2 - 2\langle \mathcal{B}, \mathcal{A}_0 \times_1 \mathbf{V}^{(1)^T} \times_2 \ldots \times_d \mathbf{V}^{(d)^T} \rangle \\ &= ||\mathcal{A}_0||^2 - ||\mathcal{B}||^2, \end{aligned}$$

hence it follows that (compare with 4.10)

$$||\mathcal{A}_0||^2 - ||\mathcal{A}_\mathbf{r}||^2 = ||\mathcal{A}_\mathbf{r} - \mathcal{A}_0||^2.$$

The latter leads to the final estimate (clearly $||\mathcal{A}_0|| \geq ||\mathcal{A}_\mathbf{r}||$)

$$\frac{||\mathcal{A}_0|| - ||\mathcal{A}_\mathbf{r}||}{||\mathcal{A}_0||} = \frac{||\mathcal{A}_\mathbf{r} - \mathcal{A}_0||^2}{(||\mathcal{A}_\mathbf{r}|| + ||\mathcal{A}_0||)||\mathcal{A}_0||} \leq \frac{||\mathcal{A}_\mathbf{r} - \mathcal{A}_0||^2}{||\mathcal{A}_0||^2}.$$

■

Numerical efficiency of standard tensor operations described above depends on the data-sparsity of the core tensor. The next lemma presents a simple but useful characterisation of the two-level Tucker model (cf. [18]) which allows to approximate the elements in $\boldsymbol{\mathcal{T}}_\mathbf{r}$ via the CP decomposition applied to the small sized core tensor (cf. dimensionality reduction in [6]).

Lemma 2.5 *(two-level Tucker-to-CP approximation). Let the target tensor* $\mathcal{A}_0 \in \boldsymbol{\mathcal{T}}_{(\mathbf{n},\mathbf{r})}$ *in the minimisation problem (4.6) have the form* $\mathcal{A}_0 = \mathcal{B}_0 \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} ... \times_d \mathbf{V}^{(d)}$ *with components* $\mathbf{V}^{(\ell)} \in \mathbb{R}^{I_\ell \times r_\ell}$ *in the Stiefel manifold, and with the core tensor* $\mathcal{B}_0 \in \mathbb{R}^{r_1 \times \ldots \times r_d}$*. Then, for a given* $q \leq |\mathbf{r}|$,

$$\min_{\mathcal{A} \in \boldsymbol{\mathcal{C}}_{(\mathbf{n},q)}} ||\mathcal{A} - \mathcal{A}_0||^2 = \min_{\mathcal{B} \in \boldsymbol{\mathcal{C}}_{(\mathbf{r},q)}} ||\mathcal{B} - \mathcal{B}_0||^2. \tag{2.17}$$

Moreover, the optimal rank-q CP approximation $\mathcal{A}_{(q)} \in \boldsymbol{\mathcal{C}}_{(\mathbf{n},q)}$ *of* $\mathcal{A}_0$ *(if existing) and the optimal rank-q CP approximation* $\mathcal{B}_{(q)} \in \boldsymbol{\mathcal{C}}_{(\mathbf{r},q)}$ *of* $\mathcal{B}_0$ *are related by*

$$\mathcal{A}_{(q)} = \mathcal{B}_{(q)} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} ... \times_d \mathbf{V}^{(d)}. \tag{2.18}$$

Proof. Notice that the canonical components $\mathbf{Y}^{(\ell)}$ of any test element

$$\mathcal{A} = \sum_{k=1}^{q} \lambda_k \times_1 Y_k^{(1)} \times_2 ... \times_d Y_k^{(d)} \tag{2.19}$$

in the left-hand side of (2.17) can be chosen in $span\{\mathbf{V}^{(\ell)}\}$ ($\ell = 1, ..., d$), i.e.,

$$Y_k^{(\ell)} = \sum_{m=1}^{r_\ell} \mu_{k,m}^{(\ell)} V_m^{(\ell)}, \quad k = 1, ..., r, \ \ell = 1, ..., d. \tag{2.20}$$

Indeed, assuming

$$Y_k^{(\ell)} = \sum_{m=1}^{r_\ell} \mu_{k,m}^{(\ell)} V_m^{(\ell)} + E_k^{(\ell)} \quad \text{with} \quad E_k^{(\ell)} \perp span\{\mathbf{V}^{(\ell)}\},$$

we conclude that $E_k^{(\ell)}$ does not effect the cost function in (2.17) because of the orthogonality of $\mathbf{V}^{(\ell)}$. Hence, setting $E_k^{(\ell)} = 0$, and substituting (2.20) into (2.19), we arrive at the desired Tucker decomposition $\mathcal{A} = \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} ... \times_d \mathbf{V}^{(d)}$ with the respective core tensor

$$\mathcal{B} = \sum_{k=1}^{q} b_k \times_1 U_k^{(1)} \times_2 ... \times_d U_k^{(d)} \in \boldsymbol{\mathcal{C}}_{(\mathbf{r},q)},$$

where $b_k = \lambda_k$, $U_k^{(\ell)} = \{\mu_{k,m_\ell}^{(\ell)}\}_{m_\ell=1}^{r_\ell} \in \mathbb{R}^{r_\ell}$, obtained from

$$\begin{aligned}
\mathcal{A} &= \sum_{k=1}^{q} \lambda_k \times_1 Y_k^{(1)} \times_2 ... \times_d Y_k^{(d)} \\
&= \sum_{k=1}^{q} \lambda_k \times_1 \Big(\sum_{m_1=1}^{r_1} \mu_{k,m_1}^{(1)} V_{m_1}^{(1)} \Big) \times_2 ... \times_d \Big(\sum_{m_d=1}^{r_d} \mu_{k,m_d}^{(d)} V_{m_d}^{(d)} \Big) \\
&= \sum_{m_1=1}^{r_1} ... \sum_{m_d=1}^{r_d} \left\{ \sum_{k=1}^{q} \lambda_k \prod_{\ell=1}^{d} \mu_{k,m_\ell}^{(\ell)} \right\} \times_1 V_{m_1}^{(1)} \times_2 ... \times_d V_{m_d}^{(d)}.
\end{aligned}$$

Now the relation (2.17) follows since the ℓ-mode multiplication with orthogonal components $\mathbf{V}^{(\ell)}$ does not change the cost function. Similar arguments justify (2.18). ■

Lemma 2.5 suggests a two-level dimensionality reduction approach that leads to a better data structure compared with the standard Tucker model. Though $\mathcal{A}_{(q)} \in \boldsymbol{\mathcal{C}}_{(\mathbf{n},q)}$ can be represented in the CP format, its efficient storage depends on the representation of the canonical components $\mathbf{V}^{(\ell)}$. In fact, if $\mathbf{V}^{(\ell)}$ are obtained via the Tucker decomposition, it is better to store $\mathcal{A}_{(q)}$ in the CP format of the complexity rdn (adaptive two-level model [18]). However, if $\mathbf{V}^{(\ell)}$ are represented in a fixed basis (say, *sinc*- or a wavelet basis) then one can store the core tensor only in the CP format, which leads to substantial memory reduction to qdr (no dependency on the data-size n).

The next statement is a direct consequence of the previous lemma.

Corollary 2.6 *(two-level CP-to-CP approximation). Let the target tensor $\mathcal{A}_0 \in \boldsymbol{\mathcal{C}}_{(\mathbf{n},\mathbf{r})}$ with $\mathbf{r} = (r, ..., r)$ in the minimisation problem (4.6) have the canonical form $\mathcal{A}_0 = \mathcal{D}_0 \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} ... \times_d \mathbf{U}^{(d)}$ with normalised components $\mathbf{U}^{(\ell)} \in \mathbb{R}^{n \times r}$ (we ignore the orthogonality requirements), and with the superdiagonal core tensor $\mathcal{D}_0 \in \mathbb{R}^{r \times \cdots \times r}$. Introduce the equivalent representation $\mathcal{A}_0 = \mathcal{B}_0 \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} ... \times_d \mathbf{V}^{(d)}$ with components $\mathbf{V}^{(\ell)} \in \mathbb{R}^{n \times r}$ in the Stiefel manifold (say, computed by the QR-decomposition of each component $\mathbf{U}^{(\ell)}$).*

Then, for a given $q \leq r$,

$$\min_{\mathcal{A} \in \boldsymbol{\mathcal{C}}_{(\mathbf{n},q)}} \|\mathcal{A} - \mathcal{A}_0\|^2 = \min_{\mathcal{B} \in \boldsymbol{\mathcal{C}}_{(\mathbf{r},q)}} \|\mathcal{B} - \mathcal{B}_0\|^2. \tag{2.21}$$

Moreover, the optimal rank-q CP approximation $\mathcal{A}_{(q)} \in \boldsymbol{\mathcal{C}}_{(\mathbf{n},q)}$ of $\mathcal{A}_0$ (if existing) and the optimal rank-q CP approximation $\mathcal{B}_{(q)} \in \boldsymbol{\mathcal{C}}_{(\mathbf{r},q)}$ of the core tensor $\mathcal{B}_0$ are related by (2.18).

Corollary 2.6 indicates that the rank reduction in the CP model can be performed via the CP approximation of a "small size" core tensor arising from the component-wise orthogonalisation in the target data-array (the complexity bound is $O(dr^2n)$).

2.4 Numerical Algorithms

There is a number of algorithms in the literature to compute the CP and the Tucker models (see, e.g., [5, 6, 23, 1]). Based on Lemmata 4.1, 4.2 a rank-$(r_1, ..., r_d)$ approximation can be calculated by the Newton-type methods.

2.4.1 General Input Data

Our current MATLAB implementation of the orthogonal ALS algorithm (OALSA) in the general case of full-format input tensors is based on the method described in [5] (see also implementation in [1]). It contains the following steps.

OALSA ($\boldsymbol{\mathcal{F}} \to \boldsymbol{\mathcal{T}}_{\mathbf{r}}$).

Given the input tensor $\mathcal{A}_0 \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and a rank-parameter $\mathbf{r} = (r_1, ..., r_d) \in \mathbb{N}^d$.
Step I: Compute the initial guess for canonical components.
Step II: For each $(m = 1, ..., d)$ the ALS iteration optimises the canonical component $\mathbf{V}^{(m)}$, while the other matrix-components are kept constant (equivalent to solving the equation number m in the system (4.3) (resp. (4.4)).
Termination criterion: fixed number of iterations or control the current increment.
Step III: Compute the core tensor via convolution (2.16).

Steps II and III are standard, while the method of choice in step I depends on the particular application. We distinguish three particular versions of **OALSA** ($\boldsymbol{\mathcal{F}} \to \boldsymbol{\mathcal{T}}_{\mathbf{r}}$) adapted to different classes of input tensors:
(F) full-format $\boldsymbol{\mathcal{F}}$ (initial guess: truncated higher-order SVD (cf. [5]), approximation with smaller Tucker rank or multi-way ACA algorithm as proposed in [22]);
(C) type $\boldsymbol{\mathcal{C}}_R$ with some $R > |\mathbf{r}|$ (may correspond to an analytic approximation via *sinc*-quadratures or exponential fitting). Initial guess: QR-decomposition with truncated higher-order SVD;

(T) type $\boldsymbol{\mathcal{T}}_{\mathbf{R}}$ (may correspond to an analytic approximation via tensor-product interpolation).
Initial guess: QR-decomposition with truncated higher-order SVD, or approximation with smaller Tucker rank accomplished with best rank-1 approximation to the initial increment;

2.4.2 Canonical-to-Tucker and Tucker-to-Tucker Decompositions

The efficient implementation in the cases (C) and (T) is based on MLA performed in special tensor formats described above. In fact, if the initial guess has data-type $\boldsymbol{\mathcal{T}}_{\mathbf{R}}$ or $\boldsymbol{\mathcal{C}}_R$ with moderate $|\mathbf{R}|$, then truncated higher-order SVD can be performed with lower cost compared with the case of full-format input data as in the case (F). Hence, we specify the corresponding versions of the general algorithm: **OALSA**$(\boldsymbol{\mathcal{C}}_R \to \boldsymbol{\mathcal{T}}_{\mathbf{r}})$ and **OALSA**$(\boldsymbol{\mathcal{T}}_{\mathbf{R}} \to \boldsymbol{\mathcal{T}}_{\mathbf{r}})$.

To perform the truncated higher-order SVD in Step I of **OALSA** $(\boldsymbol{\mathcal{C}}_R \to \boldsymbol{\mathcal{T}}_{\mathbf{r}})$, we notice that for each $\ell = 1, ..., d$ the "matrix unfolding" $A_{(\ell)}$ to the input-tensor $\mathcal{A}_0$, of dimension $I_\ell \times I_{\ell+1} \times ... \times I_d \times ... \times I_1 \times ... \times I_{\ell-1}$, can be represented as a rank-R matrix. For example, for $\ell = 1$, we have

$$A_{(1)} = [a^{(1)}_{i_1(i_2 \dots i_d)}] = [a_{i_1 i_2 \dots i_d}] \in \mathbb{R}^{n_1 \times (n_2 \cdots n_d)}.$$

We introduce the "single hole" product and related dimension parameter $\overline{n}_\ell$

$$\mathcal{V}_k^{(-\ell)} = V_k^{(1)} \times_2 ... \times_{\ell-1} V_k^{(\ell-1)} \times_{\ell+1} V_k^{(\ell+1)} ... \times_d V_k^{(d)}, \quad \overline{n}_\ell = n_1 \cdots n_{\ell-1} n_{\ell+1} \cdots n_d, \tag{2.22}$$

and represent the rank-R matrix unfolding in the form $A_{(\ell)} = A_\ell B_\ell^T$ with $A_\ell \in \mathbb{R}^{n_\ell \times R}$, $B_\ell \in \mathbb{R}^{\overline{n}_\ell \times R}$ given by

$$A_\ell = \mathbf{V}^{(\ell)} D \quad \text{with} \quad D = diag\{b_1, ..., b_R\} \tag{2.23}$$

and with

$$B_\ell = [V_1^{(-\ell)}, ..., V_r^{(-\ell)}], \tag{2.24}$$

where $V_k^{(-\ell)}$ is the vector unfolding to $\mathcal{V}_k^{(-\ell)}$. Then the algorithm reads as follows.

OALSA $(\boldsymbol{\mathcal{C}}_R \to \boldsymbol{\mathcal{T}}_{\mathbf{r}})$
Given the input tensor $\mathcal{A}_0 \in \boldsymbol{\mathcal{C}}_R$ in the form (4.5).
Step I: For each $\ell = 1, ..., d$ perform
1. QR-decomposition of A_ℓ and B_ℓ (cf. (2.23) and (2.24), respectively) in the form, $A_\ell = Q_A^{(\ell)} R_A^{(\ell)}$, $B_\ell = Q_B^{(\ell)} R_B^{(\ell)}$;
2. SVD of a matrix $S_\ell = R_A^{(\ell)} R_B^{(\ell)^T} \in \mathbb{R}^{R \times R}$ in the form $S_\ell = W_\ell D_\ell V_\ell$;
3. Truncation of S_ℓ to rank r_ℓ matrix $\tilde{W}_\ell \tilde{D}_\ell \tilde{V}_\ell$ with $\tilde{W}_\ell \in \mathbb{R}^{R \times r_\ell}$;
4. Compute the canonical components $\mathbf{U}^{(\ell)} = Q_A^{(\ell)} \tilde{W}_\ell$.
Starting with initial values $\mathbf{U}^{(\ell)}$, $\ell = 1, ..., d$, proceed with Steps II and III as in the general version of OALSA.

Efficient implementation of Step I in **OALSA** $(\boldsymbol{\mathcal{T}}_{\mathbf{R}} \to \boldsymbol{\mathcal{T}}_{\mathbf{r}})$ is based on the observation that for each $\ell = 1, ..., d$ the "matrix unfolding" $A_{(\ell)}$ to $\mathcal{A}_0 \in \boldsymbol{\mathcal{T}}_{\mathbf{R}}$, can be represented as a rank-R_ℓ matrix. In fact, it is a direct consequence of representation (2.3). Step I of the corresponding algorithm then can be designed similar to those in **OALSA** $(\boldsymbol{\mathcal{C}}_R \to \boldsymbol{\mathcal{T}}_{\mathbf{r}})$. The resulting algorithm reads as follows.

OALSA ($\mathcal{T}_{\mathbf{R}} \to \mathcal{T}_{\mathbf{r}}$)

```
Given the input tensor $\mathcal{A}_0 \in \mathcal{T}_{\mathbf{R}}$ in the form (2.3).
Step I: For each $\ell = 1, ..., d$ compute
1. Components $A_\ell$ and $B_\ell$ in representation $A_{(\ell)} = A_\ell B_\ell^T$, and perform
2. $QR$-decomposition of $A_\ell$ and $B_\ell$ in the form, $A_\ell = Q_A^{(\ell)} R_A^{(\ell)}$, $B_\ell = Q_B^{(\ell)} R_B^{(\ell)}$;
3. SVD of the matrix $S_\ell = R_A^{(\ell)} R_B^{(\ell)^T} \in \mathbb{R}^{R\times R}$ in the form $S_\ell = W_\ell D_\ell V_\ell$;
4. Truncation of $S_\ell$ to a rank $r_\ell$ matrix $\tilde{W}_\ell \tilde{D}_\ell \tilde{V}_\ell$ with $\tilde{W}_\ell \in \mathbb{R}^{R\times r_\ell}$;
5. Compute the canonical components $\mathbf{U}^{(\ell)} = Q_A^{(\ell)} \tilde{W}_\ell$.
Starting with initial values $\mathbf{U}^{(\ell)}$, $\ell = 1, ..., d$, proceed with Steps II and III as in the general version of OALSA.
```

2.4.3 Combination of the Tucker and CP Formats

In some quantum chemistry applications the target tensor may contain canonical components having different scales of amplitudes and decay rates. In this case we can observe nonstable convergence of the ALS iteration, likely, due to large variation in elements of the core tensor. Similar to the matrix case, we call this phenomenon as *ill-conditioning* of a tensor.

In such cases, to stabilize the convergence of the ALS iteration (without destroying the approximation power), we introduce the *mixed model* denoted by $\mathcal{M}_{(Cr_1, T\mathbf{r}_2)}$. We say that $\mathcal{A} \in \mathcal{M}_{(Cr_1, T\mathbf{r}_2)}$ if

$$\mathcal{A} := \mathcal{A}_1 + \mathcal{A}_2 \quad \text{with } \mathcal{A}_1 \in \mathcal{C}_{r_1},\ \mathcal{A}_2 \in \mathcal{T}_{\mathbf{r}_2}.$$

We assume that the dominating component in $\mathcal{A}$ can be well approximated via the CP model, $\mathcal{A}_1 \in \mathcal{C}_{r_1}$, while the residual (which is better conditioned) can be further approximated in the Tucker format, $\mathcal{A}_2 \in \mathcal{T}_{\mathbf{r}_2}$. To approximate the given tensor $\mathcal{A}_0$ by the $\mathcal{M}_{(Cr_1, T\mathbf{r}_2)}$ model, we apply the following iterative (heuristic) algorithm: Given the input tensor $\mathcal{A}_0 \in \mathbb{R}^{n_1\times \dots \times n_d}$,
1. Compute its CP approximation $\mathcal{C}_0$ in $\mathcal{C}_{r_1}$;
2. Compute the Tucker decomposition $\mathcal{T}_0$ of $\mathcal{A}_0 - \mathcal{C}_0$ with $\mathcal{T}_0 \in \mathcal{T}_{\mathbf{r}_2}$;
3. If $\|\mathcal{A}_0 - \mathcal{C}_0 - \mathcal{T}_0\| \le \varepsilon_{tol}$ then stop, otherwise go to Step 1 with $\mathcal{A}_0$ substituted by $\mathcal{A}_0 - \mathcal{T}_0$.

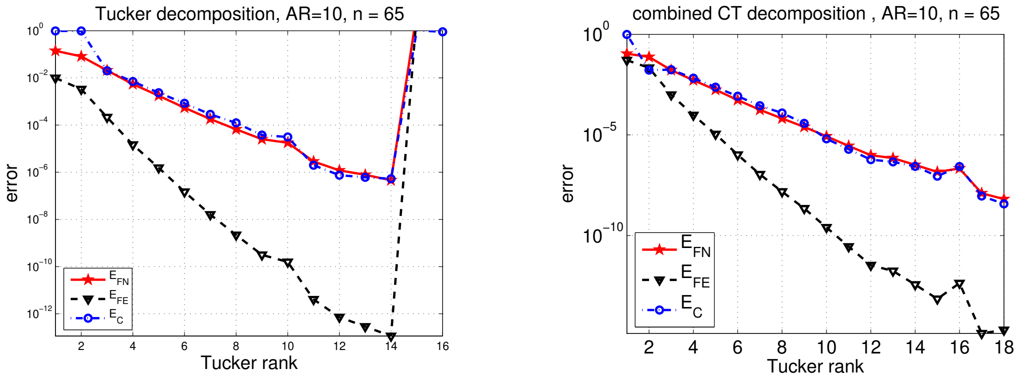

Figure 2.2: Convergence history for the Tucker (left) and mixed CT (right) approximations for the tri-Slater potential (2.25).

We give numerical example for the mixed tri-linear approximation of the FGT corresponding to

$$g(x) = C_1 e^{-\alpha_1|x|} + C_2 e^{-\alpha_2|x-x_2|} + C_3 e^{-\alpha_3|x-x_3|} \tag{2.25}$$

in $\mathbb{R}^3$ with $0 < C_2, C_3 \ll C_1$, $0 < \alpha_2, \alpha_3 \ll \alpha_1$. This function has similar features with the case of H_2O-molecule. For this example, the first component can be well approximated via the rank-1 tensor, hence we use $\mathcal{A}_1 \in \boldsymbol{\mathcal{C}}_1$.

Figure 2.2 corresponds to the choice $C_1 = 150, C_2 = 30, C_3 = 20$, $\alpha_1 = 5000, \alpha_2 = \alpha_3 = 1$ and $x_2 = (AR/6, 0, AR/5); x_3 = (AR/6, 0, -AR/5)$. Pictures illustrate a stable exponential convergence in the Tucker rank for the mixed CT model, which allows to achieve a given accuracy up to the machine precision.

3 Application to Classical Potentials

Here we discuss the Tucker decomposition of 3D tensors arising as discretization of classical potentials. Let $\prod^d \in \mathbb{R}^d$ be a uniform or adaptively refined tensor-product grid indexed by $I_1 \times \ldots \times I_d$. For a given function $g : \Omega \to \mathbb{R}$, with $\Omega \subset \mathbb{R}^d$ and with $\prod^d \in \Omega$, we introduce the collocation-type function-generated tensor (FGT) of order d by

$$\mathcal{A}_0 \equiv \mathcal{A}(g) := [a_{i_1 \ldots i_d}] \in \mathbb{R}^{I_1 \times \ldots \times I_d} \text{ with } \; a_{i_1 \ldots i_d} := g(x_{i_1}^{(1)}, \ldots, x_{i_d}^{(d)}),$$

where $(x_{i_1}^{(1)}, \ldots, x_{i_d}^{(d)}) \in \prod^d \in \mathbb{R}^d$ are grid collocation points. We are interested in the validity and the rank-dependence of the above tensor decomposition algorithms for approximating the 3D FGT generated by the Newton potential, Slater-type functions, the Yukawa and the Helmholtz potentials.

The initial tensor $\mathcal{A}_0$ is decomposed by the Tucker model of the rank $\mathbf{r} = (r, ..., r)$, where the rank-parameter r increases from $r = 1, 2, ...$ to some predefined value. Canonical components and the core tensor of the size $r \times r \times r$ are then used for the reconstruction of the approximating tensor $\mathcal{A}_{(r)} \approx \mathcal{A}_0$, which is used for estimating approximation properties of the tensor decomposition with the given rank. For every rank $\mathbf{r}$ Tucker decomposition, we compute the relative energy-norm (Frobenius norm) as in (2.1)

$$E_{FN} = \frac{||A_0 - A_{(r)}||}{||A_0||},$$

the relative L_2-energy

$$E_{FE} = \frac{||A_0|| - ||A_{(r)}||}{||A_0||},$$

as well as the maximum (Chebyshev) norm

$$E_C := \frac{\max_{i \in \mathcal{I}} |a_{0,i} - a_{r,i}|}{\max_{i \in \mathcal{I}} |a_{0,i}|}.$$

3.1 Newton Potential

We apply the Tucker decomposition algorithm with $\mathbf{r} = (r, ..., r)$ for approximating the Newton potential

$$g(x) = \frac{1}{|x|}, \quad x \in \mathbb{R}^3,$$

in the cube $[0, AR]^3$ $(AR = 10)$ on the cell-centred uniform grid with $n = 64$. Here and in the following $|x|$ denotes the Euclidean norm of $x \in \mathbb{R}^d$. Figure 3.1 shows the convergence of the relative energy- and Chebyshev norms as well as of the relative energy with respect to the Tucker rank up to $r = 12$. The canonical components $U_k^{(1)}$ are given for $k = 1, \ldots, 6$. It is clearly seen the exponential convergence in the Tucker rank r.

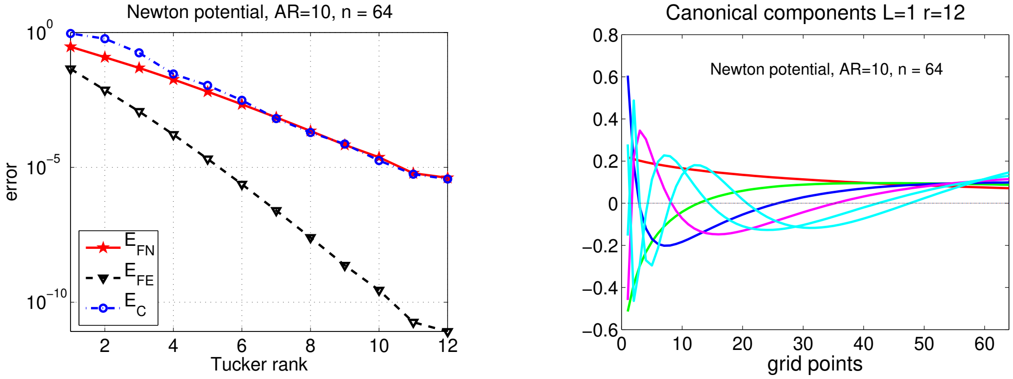

Figure 3.1: Convergence history (left) and the canonical components $U_k^{(1)}$, $k = 1, \ldots, 6$, (right) for the Tucker approximation to the Newton potential.

The particular E_C-error distribution is represented in Figure (3.2).

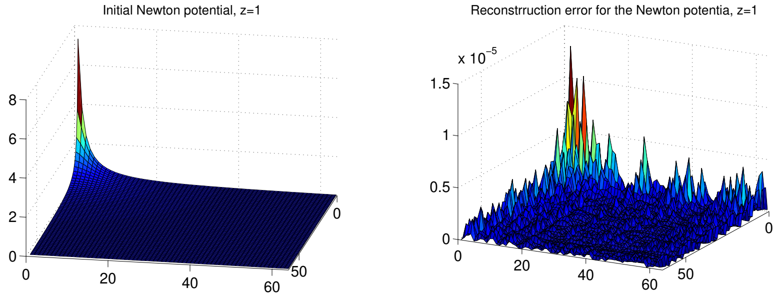

Figure 3.2: A plane section of the 3D Newton potential on the level $z = 33$: The initial function (left), the E_C-error for the same z-coordinate level with $r = 12$ (right).

3.2 Functions Related to the Hartree-Fock Equation

The Hartree-Fock equation for the N-electrons density function reads as the system of nonlinear eigenvalue problems

$$\mathcal{F}\phi_i(x) = \lambda_i\, \phi_i(x), \quad \text{for} \quad i = 1, ..., N/2 \tag{3.1}$$

with $x \in \mathbb{R}^3$. Here the solution-dependent Fock operator is defined by

$$\mathcal{F}\phi(x) := -\frac{1}{2}\Delta\phi(x) - V_c(x)\,\phi(x) + (\mathcal{J}\phi)(x) + (\mathcal{K}\phi)(x), \quad x \in \mathbb{R}^3,$$

where with the given density matrix $\rho(x, y)$, the Hartree and exchange potentials are defined by

$$(\mathcal{J}\phi)(x) := \int d^3y\, \frac{\rho(y,y)}{|x-y|}\,\phi(x), \quad (\mathcal{K}\phi)(x) := -\frac{1}{2}\int d^3y\, \frac{\rho(x,y)}{|x-y|}\,\phi(y),$$

correspondingly. Furthermore, the nuclei potential is given by $V_c(x) = \sum\limits_{a=1}^{K} \frac{Z_a}{|x-R_a|}$, where R_a and Z_a describe the positions and charges of nuclei, respectively. With given eigen-functions ϕ_i, the *density matrix* is defined by the corresponding spectral projection

$$\rho(x, y) = 2\sum_{i=1}^{N/2} \phi_i^*(x)\phi_i(y)$$

with exponential decay $\rho(x, y) \sim \exp(-\lambda|x - y|)$ for nonmetallic systems. The problem consists of a tensor-product representation of the density functions $\rho(x, y)$ and $\rho(x, x)$ as well as of the Hartree and exchange potentials involved. Along with the Newton potential discussed above, in the following we consider some simple examples of the density function. Notice that computation of the Hartree potential can be performed via direct evaluation of the convolution product between the Newton potential and density $\rho(x, x)$ represented in the Tucker format (cf. §2.2.4). In this case the Newton potential is supposed to be approximated by the CP model (see [12],[18] for more details).

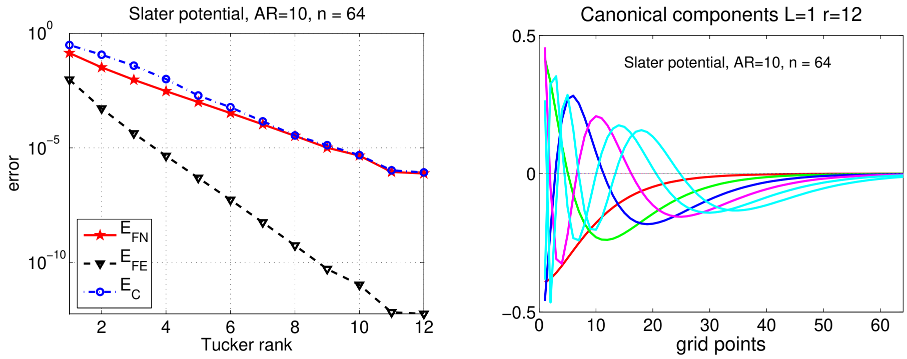

Figure 3.3: Convergence history and canonical components for the Slater potential.

The Slater function given by

$$g(x) = \exp(-\alpha|x|) \quad \text{with} \quad x = (x_1, x_2, x_3)^T \in \mathbb{R}^3$$

presents the electron “orbital” ($\alpha = 1$) and the electron density function ($\alpha = 2$) corresponding to the Hydrogen atom. In this case $g(x)$ satisfies the one-particle Hartree-Fock equation which takes the form

$$-\frac{1}{2}\Delta\phi(x) - \frac{1}{|x|}\phi(x) = \lambda\phi(x), \quad x \in \mathbb{R}^3, \quad \phi \in H^1(\mathbb{R}^3).$$

We apply the orthogonal Tucker decomposition to the FGT defined on the grid $\prod^3$ with $AR = 10$. Figure 3.4 presents the slices of the $9 \times 9 \times 9$ 3D core tensor, where the numbers indicate the maximum values at the given slice of $\mathcal{B}$. Figure shows that the energy of the decomposed function is concentrated in several upper slices of the core tensor. It exposes the potential data compression abilities of the Tucker approximation due to the sparsity of thresholded $\mathcal{B}$. However, our numerical experiments show that the dominating entries in $\mathcal{B}$ are compactly concentrated in its ”upper left corner” which indicates that the core tensor truncation may have, in fact, the similar effect as just a decomposition with a smaller Tucker rank.

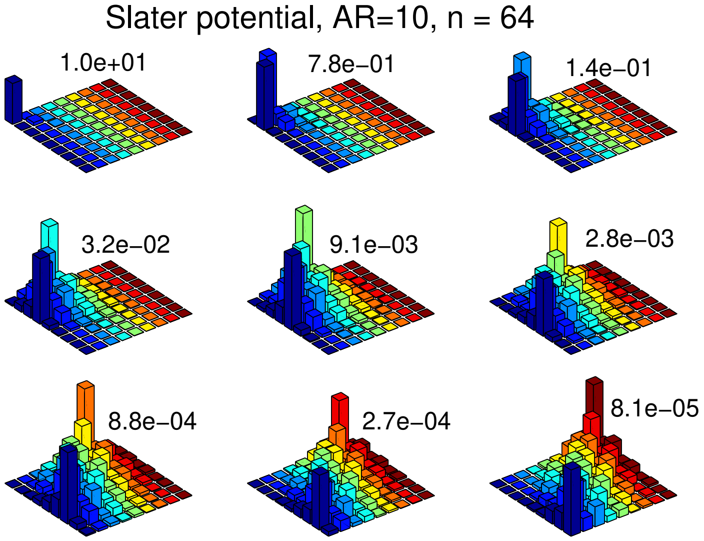

Figure 3.4: Slices of the $9 \times 9 \times 9$ core tensor for the Tucker approximation of the Slater potential. Numbers indicate maximum values in $\mathcal{B}$ for the given slice.

Since the Hartree-Fock equation contains the product $\frac{1}{|x|}\phi(x)$ with $\phi(x) = e^{-|x|}$ (the so-called Yukawa potential) it is interesting to compute the tensor decomposition of the Hadamard product of the discrete Newton and Slater potentials. The convergence results for the Tucker approximation are presented in Fig. 3.8. Next, we consider a radially-symmetric potential generated by the modified erf-function given by

$$g(x) = \frac{\mathrm{erf}(|x|)}{|x|} \quad \text{with} \quad x = (x_1, x_2, x_3)^T \in \mathbb{R}^3,$$

which frequently arises in quantum chemistry computations. Here we define

$$\mathrm{erf}(z) := \int_0^z e^{-z^2t^2}dt, \quad z \geq 0.$$

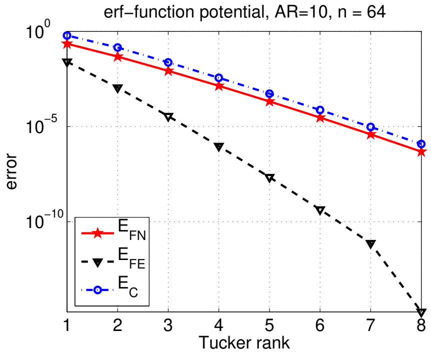

Figure 3.5: Approximation error for the function $\frac{\operatorname{erf}(x)}{|x|}$.

Behaviour of the approximation errors for the 3D Tucker decomposition of erf-function is shown in Figure 3.5. Computations were performed on the $n \times n \times n$ grid with $n = 64$, for the interval $AR = 10$. Rank-r approximation exhibits a good convergence rate already for the Tucker rank $r = 8$, the corresponding Frobenius norm error is of the order 10^{-6}.

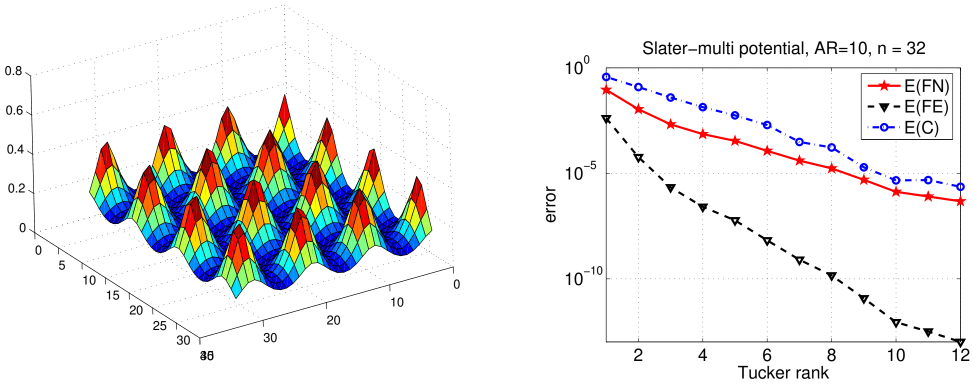

Figure 3.6: A slice $n \times n \times 2$ of a 3D multi-centred Slater potential (left) and the corresponding approximation error vs. the Tucker rank (right).

Finally, we analyse the "multi-centred Slater potential“ obtained by displacing a single Slater potential with respect to the $m \times m \times m$ spatial grid of size $H > 0$ with randomly perturbed centres,

$$g(x) = \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} e^{-\alpha \sqrt{(x_1 - iH)^2 + (x_2 - jH)^2 + (x_3 - kH)^2}}.$$

Figure 3.6 shows the multi-centered Slater potential for $m = 4$, $H = 3$, $\alpha = 2$ approximated in the cube $[0, AR]^3$ with $AR = 10$ on the $n \times n \times n$ grid with $n = 64$, the surface level corresponds to $z = 2$.

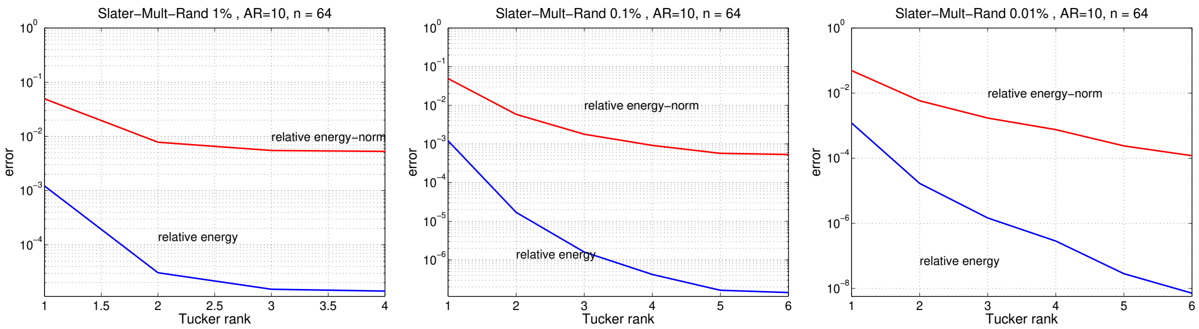

Figure 3.7: Convergence history for the multi-centred randomly perturbed Slater potential.

3.3 Yukawa and Helmholtz Potentials

In the next example, we consider a trilinear Tucker decomposition of the third-order function-related tensor generated by the Yukawa potential

$$g(x) = \frac{e^{-|x|}}{|x|} \quad \text{with} \quad x = (x_1, x_2, x_3)^T \in \mathbb{R}^3.$$

We consider the FGT with cell-centred collocation points with respect to the $n \times n \times n$-grid over $[0, AR]^3$ with $AR = 10$. Figure 3.8 shows the convergence history and the canonical components for the Tucker decomposition of the Yukawa potential given on uniform grids with $n = 32, 64, 128$, and with the core tensor of size $6 \times 6 \times 6$. These components represent the "optimal" adaptive basis which has a tendency to reproduce shapes similar to the *Sinc*-functions. In almost all cases the ALS method requires not more than 5 iterations indicating robust convergence in the considered applications.

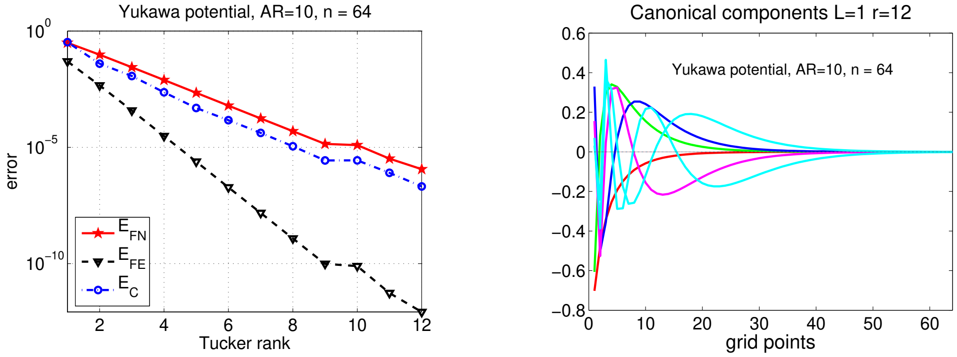

Figure 3.8: Convergence history for the Tucker approximation of the Yukawa potential and example of the canonical components.

Computations for the Helmholtz function given by

$$g(x) = \frac{\cos |x|}{|x|} \quad \text{with} \quad x = (x_1, x_2, x_3)^T \in \mathbb{R}^3$$

provide the results from Fig. 3.9.

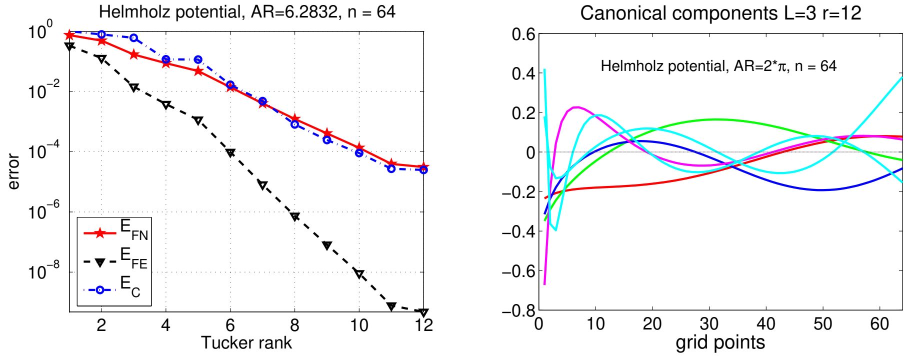

Figure 3.9: Convergence history (left) and canonical components U_k, $k = 1, \ldots, 6$, (right) for the Tucker approximation of the Helmholtz potential.

4 Appendix

4.1 Lagrange Equation for the Dual Maximisation Problem

For given components $\mathbf{V}^{(\ell)} = [V_1^{(\ell)} V_2^{(\ell)} ... V_{r_\ell}^{(\ell)}]$, we define the "single hole" tensor

$$\mathcal{B}^{(\neg m)} = \mathcal{A}_0 \times_1 \mathbf{V}^{(1)^T} ... \times_{m-1} \mathbf{V}^{(m-1)^T} \times_{m+1} \mathbf{V}^{(m+1)^T} ... \times_d \mathbf{V}^{(d)^T}$$

and let $\mathbf{B}^{(\neg m)} \in \mathbb{R}^{n_m \times \overline{r}_m}$ be the corresponding matrix representation, where $\overline{r}_m$ is defined in (2.22). The following lemma reduces the minimisation of the original quadratic functional to the dual maximisation problem thus eliminating the core tensor $\mathcal{B}$ from the solution process.

Lemma 4.1 *([5]) For given $\mathcal{A}_0 \in \mathbb{R}^{I_1 \times ... \times I_d}$, the minimisation problem (2.12) on $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$ is equivalent to the dual maximisation problem*

$$g(\mathbf{V}^{(1)}, ..., \mathbf{V}^{(d)}) := \left\| \mathcal{A}_0 \times_1 \mathbf{V}^{(1)^T} \times_2 ... \times_d \mathbf{V}^{(d)^T} \right\|^2 \rightarrow max \qquad (4.1)$$

over a set $\mathbf{V}^{(\ell)} \in \mathbb{R}^{|I_\ell| \times r_\ell}$ from the Stiefel manifold, i.e., $\mathbf{V}^{(\ell)} \in \mathcal{V}_\ell$ $(\ell = 1, ..., d)$. For given matrices $\mathbf{V}^{(m)}$ $(m = 1, ..., d)$, the tensor $\mathcal{B}$ minimising (2.12) is represented by

$$\mathcal{B} = \mathcal{A}_0 \times_1 \mathbf{V}^{(1)^T} \times_2 ... \times_d \mathbf{V}^{(d)^T} \in \mathbb{R}^{r_1 \times ... \times r_d}. \qquad (4.2)$$

The following lemma provides the explicit Lagrange equations for the dual maximisation problem.

Lemma 4.2 *([18]) The problem (4.1) has at least one global maximum. At each extremal point the corresponding Lagrange equations read as*

$$2(\mathbf{I} - \mathbf{V}^{(m)} \mathbf{V}^{(m)^T}) \cdot \mathbf{B}^{(\neg m)} \cdot \mathbf{B}^{(\neg m)^T} \cdot \mathbf{V}^{(m)} = 0 \quad (1 \leq m \leq d). \qquad (4.3)$$

Under the compatibility condition $r_m \leq \overline{r}_m$ $(1 \leq m \leq d)$ equation (4.3) is solvable for any $m = 1, ..., d$.

It is readily seen that in the case of a rank-1 approximation (i.e., $\mathbf{r} = (1, ..., 1)$) the system of a Lagrange equations (4.3) combined with (4.2) can be written in the form

$$\begin{aligned}
&\mathcal{A}_0 \times_1 V^{(1)^T} ... \times_{m-1} V^{(m-1)^T} \times_{m+1} V^{(m+1)^T} ... \times_d V^{(d)^T} = b_1 V^{(m)}, \qquad (4.4)\\
&\mathcal{A}_0 \times_1 V^{(1)^T} \times_2 ... \times_d V^{(d)^T} = b_1,\\
&\|V^{(m)}\| = 1 \quad (1 \leq m \leq d)
\end{aligned}$$

with $b_1 \in \mathbb{R}$ (cf. [5]). These equations explicitly represent the numerical scheme of the ALS iteration for computing the best rank-1 approximation.

4.2 Canonical Tensor Decomposition

The CP model is a simplified version of a *general* Tucker decomposition (2.3) defined by

$$\mathcal{A}_{(r)} = \sum_{k=1}^{r} b_k \times_1 V_k^{(1)} \times_2 ... \times_d V_k^{(d)} \approx \mathcal{A}, \quad b_k \in \mathbb{C}, \qquad (4.5)$$

where the Kronecker factors $V_k^{(\ell)} \in \mathbb{C}^{I_\ell}$ are unit-norm vectors. Indeed, the decomposition (4.5) can be viewed as a special case of the Tucker model (2.3), where $r = r_1 = ... = r_d$ and $b_{k_1 ... k_d} = 0$ unless $k_1 = k_2 = ... = k_d$, i.e., only the super-diagonal of $\mathcal{B} = \{b_{\mathbf{k}}\}$ is non-zero.

The trilinear CP-decomposition is visualised in Fig. 4.1. The minimal number r in the representation (4.5) is called the Kronecker rank of a given tensor $\mathcal{A}_{(r)}$. Under moderate assumptions, the CP decomposition with rank r is unique. We denote by $\boldsymbol{\mathcal{C}}_r$ the set of component-wise normalised tensors parametrised by (4.5). Given $\mathcal{A}_0 \in \mathbb{R}^{I_1 \times ... \times I_d}$, its CP approximation can be derived by minimisation of the quadratic cost functional

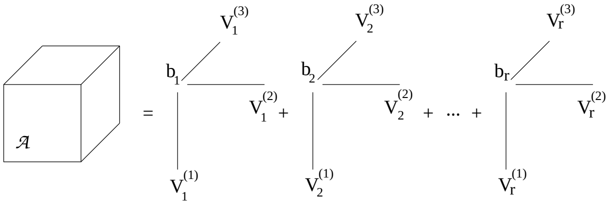

Figure 4.1: Visualisation of the CP-decomposition for a third-order tensor.

$$f(\mathcal{A}) := \|\mathcal{A} - \mathcal{A}_0\|^2 \to min \qquad (4.6)$$

over all rank-r tensors $\mathcal{A} \in \boldsymbol{\mathcal{C}}_r$. Simple methods to construct the CP approximation which avoid solving the minimisation problem (4.6) can be based on greedy algorithms (cf. [25]).

4.3 Rayleigh Quotient Approximation

For a given symmetric matrix $A \in V = \mathbb{R}^{n \times n}$, the so-called Rayleigh quotient

$$R(u) := \frac{\langle Au, u \rangle}{\langle u, u \rangle}, \quad u \in \mathbb{R}^n$$

is known to have the fundamental property, that

$$\lambda = \min_{u \in V,\, u \neq 0} R(u) \quad \text{and} \quad v = \operatorname*{argmin}_{u \in V,\, u \neq 0} R(u) \tag{4.7}$$

appear to be the minimal eigen-value and the corresponding eigen-vector of A,

$$Av = \lambda v.$$

Proposition 4.3 *Assume that we have an approximation to λ (resp to v) via minimisation on a certain subspace $V_r \subset V$ with $\dim(V_r) = r < n$,*

$$\lambda_r = \min_{u \in V_r,\, u \neq 0} R(u) \quad \textit{and} \quad v_r = \operatorname*{argmin}_{u \in V_r,\, u \neq 0} R(u). \tag{4.8}$$

Then one obtains the quadratic error estimate for the eigen-value λ_r

$$\lambda_r - \lambda \leq \|A - \lambda I\|_2 \|v - v_r\|^2. \tag{4.9}$$

Proof. The proof is instructive. Supposing that $\langle v, v \rangle = \langle v_r, v_r \rangle = 1$, we obtain (cf. [24])

$$\begin{aligned} \langle A(v - v_r), v - v_r \rangle &= \langle Av, v \rangle - 2\langle Av, v_r \rangle + \langle Av_r, v_r \rangle \\ &= \lambda - 2\langle v, v_r \rangle + \lambda_r \\ &= \lambda(2 - 2\langle v, v_r \rangle) + \lambda_r - \lambda \\ &= \lambda\langle v - v_r, v - v_r \rangle + \lambda_r - \lambda, \end{aligned}$$

which implies

$$\lambda_r - \lambda = \langle (A - \lambda I)(v - v_r), v - v_r \rangle. \tag{4.10}$$

Then (4.9) follows.

■

Acknowledgements. Numerous helpful suggestions by Prof. W. Hackbusch are gratefully acknowledged.

References

[1] B.W. Bader and T.G. Kolda: MATLAB Tensor Classes for Fast Algorithm Prototyping. SANDIA Report, SAND2004-5187, Sandia National Laboratories, 2004.

[2] G. Beylkin, M. M. Mohlenkamp: Numerical operator calculus in higher dimensions, *PNAS*, Vol. 99, no. 16, 2002, 10246–10251.

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

[4] L. De Lathauwer, B. De Moor, J. Vandewalle: *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., Vol. 21, no. 4, 2000, 1253–1278.

[5] L. De Lathauwer, B. De Moor, J. Vandewalle: *On the best rank-*1 *and rank-*$(R_1, ..., R_N)$ *approximation of higher-order tensors.* SIAM J. Matrix Anal. Appl., 21 (2000) 1324-1342.

[6] L. De Lathauwer, B. De Moor, J. Vandewalle: *Computation of the canonical decomposition by means of a simultaneous generalised Schur decomposition.* SIAM J. Matrix Anal. Appl., 26 (2004) 295-327.

[7] H.-J. Flad, W. Hackbusch, B.N. Khoromskij, and R. Schneider: *Concept of data-sparse tensor-product approximation in many-particle models*, Leipzig-Kiel, 2006 (in preparation).

[8] I.P. Gavrilyuk, W. Hackbusch, and B.N. Khoromskij: *Tensor-product approximation to elliptic and parabolic solution operators in higher dimensions.* Computing **74** (2005), 131-157.

[9] G.H. Golub, C.F. Van Loan: Matrix Computations, Johns Hopkins University Press, Baltimore, MD, 1996.

[10] W. Hackbusch: *Fast and Exact Projected Convolution for Non-equidistant Gridss.* Preprint **102**, MPI MIS, Leipzig 2006.

[11] W. Hackbusch and B.N. Khoromskij: *Low-rank Kronecker product approximation to multi-dimensional nonlocal operators. Part I. Separable approximation of multi-variate functions;* Computing **76** (2006), 177-202.

[12] W. Hackbusch and B.N. Khoromskij: *Low-rank Kronecker product approximation to multi-dimensional nonlocal operators. Part II. HKT representations of certain operators.* Computing **76** (2006), 203-225.

[13] W. Hackbusch, B.N. Khoromskij and E.E. Tyrtyshnikov: *Hierarchical Kronecker tensor-product approximations.* J. Numer. Math. **13** (2005), 119–156.

[14] W. Hackbusch, B.N. Khoromskij and E.E. Tyrtyshnikov: *Approximate Iterations for Structured Matrices.* Preprint 112, MPI MIS, Leipzig 2005 (Numer. Math., submitted).

[15] R. Harshman: *Foundation of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis.* UCLA Working Papers in Phonetics, 16 (1970), 1-84.

[16] B.N. Khoromskij: *Structured data-sparse approximation to high order tensors arising from the deterministic Boltzmann equation.* Preprint 4, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2005 (to appear in Math. Comp.).

[17] B.N. Khoromskij: *An Introduction to Structured Tensor-Product Representation of Discrete Nonlocal Operators.* Lecture Notes 27, MPI MIS, Leipzig 2005.

[18] B.N. Khoromskij: *Structured Rank-*$(r_1, ..., r_d)$ *Decomposition of Function-related Tensors in* $\mathbb{R}^d$. Comp. Meth. in Applied Math., **6** (2006), 2, 194-220.

[19] T. Kolda: *Orthogonal tensor decompositions.* SIAM J. Matrix Anal. Appl. 23 (2001) 243-255.

[20] J.B. Kruskal: *Three-way arrays: rank and uniqueness of trilineardecompositions, with applications to arithmetic complexity and statistics.* Linear Algebra Appl., 18 (1977) 95-138.

[21] Ch. Lubich: *On Variational Approximations in Quantum Molecular Dynamics.* Math. Comp. **74** (2005), 765-779.

[22] I.V. Oseledets, D.V. Savostianov, and E.E. Tyrtyshnikov: Tucker Dimensionality Reduction of Three-Dimensional Arrays in Linear Time. SIAM J. Matrix Anal. Appl., 2006 (to appear).

[23] A. Smilde, R. Bro, P. Geladi: Muti-way Analysis. Wiley, 2004.

[24] J. Strang and G.J. Fix. An Analysis of the Finite Element Method. Prentice-Hall, inc. N. J., 1973.

[25] V.N. Temlyakov: Greedy Algorithms and M-Term Approximation with Regard to Redundant Dictionaries. J. of Approx. Theory **98** (1999), 117-145.

[26] L.R. Tucker: *Some mathematical notes on three-mode factor analysis.* Psychometrika 31 (1966) 279-311.

[27] E.E. Tyrtyshnikov: *Tensor approximations of matrices generated by asymptotically smooth functions.* Sbornik: Mathematics **194**, No. 5-6 (2003), 941–954 (translated from *Mat. Sb.* **194**, No. 6 (2003), 146–160).

[28] T. Zang and G. Golub: Rank-0ne approximation to high order tensors. SIAM J. Matrix Anal. Appl. v. 23 (2001) 534-550.