Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Fast and Accurate Tensor Approximation of Multivariate Convolution with Linear Scaling in Dimension

(revised version: May 2009)

by

Boris N. Khoromskij

Preprint no.: 36 2008

Fast and Accurate Tensor Approximation of Multivariate Convolution with Linear Scaling in Dimension

Boris N. Khoromskij
Max-Planck-Institute for Mathematics in the Sciences,
Inselstr. 22-26, D-04103 Leipzig, Germany.
bokh@mis.mpg.de

Abstract

In the present paper we present the tensor-product approximation of multi-dimensional convolution transform discretized via collocation-projection scheme on the uniform or composite refined grids. Examples of convolving kernels are given by the classical Newton, Slater (exponential) and Yukawa potentials, $1/\|x\|$, $e^{-\lambda\|x\|}$ and $e^{-\lambda\|x\|}/\|x\|$ with $x \in \mathbb{R}^d$. For piecewise constant elements on the uniform grid of size n^d, we prove the quadratic convergence $O(h^2)$ in the mesh parameter $h = 1/n$, and then justify the Richardson extrapolation method on a sequence of grids that improves the order of approximation up to $O(h^3)$. The fast algorithm of complexity $O(dR_1R_2n\log n)$ is described for tensor-product convolution on the uniform/composite grids of size n^d, where R_1, R_2 are tensor ranks of convolving functions. We also present the tensor-product convolution scheme in the two-level Tucker-canonical format and discuss the consequent rank reduction strategy. Finally, we give numerical illustrations confirming: (a) the approximation theory for convolution schemes of order $O(h^2)$ and $O(h^3)$; (b) linear-logarithmic scaling of 1D discrete convolution on composite grids; (c) linear-logarithmic scaling in n of our tensor-product convolution method on $n \times n \times n$ grid in the range $n \leq 16384$.

AMS Subject Classification: 65F30, 65F50, 65N35, 65F10
Key words: Kronecker products, Tucker tensor decomposition, canonical tensors, multi-dimensional convolution, FFT, collocation-projection method, Richardson extrapolation, composite grids.

1 Introduction

The multi-dimensional convolution arises in a variety of mathematical models which include multivariate correlation functions, Green's functions of an elliptic operator or some other translation invariant transforms (filtering). As examples in scientific computing, we mention many-particle modelling based on the Hartree-Fock, Kohn-Sham and Boltzmann equations as well as the Lippmann-Schwinger formulation of the Schrödinger equation. Further applications appear in the image/signal processing, population modelling and financial mathematics.

Our particular motivation is concerned with efficient solution methods in electronic and molecular structure calculations. As the basic example, let us consider the Hartree-Fock equation for pairwise orthogonal electronic orbitals $\phi_i : \mathbb{R}^3 \to \mathbb{R}$, which reads as

$$\mathcal{F}_\Phi \phi_i(x) = \lambda_i \, \phi_i(x), \qquad \int_{\mathbb{R}^3} \phi_i \phi_j = \delta_{ij}, \; i,j = 1, ..., N \tag{1.1}$$

with $\mathcal{F}_\Phi$ being the nonlinear Fock operator

$$\mathcal{F}_\Phi := -\frac{1}{2}\Delta - V_c + V_H - \mathcal{K}.$$

Here we use the definitions

$$\tau(x,y) := \sum_{i=1}^{N} \phi_i^*(x)\phi_i(y), \quad \rho(x) := \tau(x,x), \quad V_H := \left(\rho \star \frac{1}{\|\cdot\|}\right) = \int_{\mathbb{R}^3} \frac{\rho(y)}{\|\cdot - y\|} dy,$$

$$(\mathcal{K}\phi)(x) := -\frac{1}{2}\sum_{i=1}^{N} \left(\phi\, \phi_i \star \frac{1}{\|\cdot\|}\right) \phi_i^*(x) = -\frac{1}{2}\int_{\mathbb{R}^3} \frac{\tau(x,y)}{\|x-y\|}\, \phi(y) dy.$$

with the density matrix $\tau(x,y)$, electron density $\rho(x)$, the atomic potential $V_c(x) = \sum_{\nu=1}^{M} \frac{Z_\nu}{|x-a_\nu|}$, the Hartree potential $V_H(x)$ and the nonlocal exchange operator $\mathcal{K}$. The most computationally expensive part in numerical solution of that equation includes the convolution products with the Newton potential in $\mathbb{R}^3$,

$$\rho * \frac{1}{\|\cdot\|} \quad \text{and} \quad \phi\, \phi_i * \frac{1}{\|\cdot\|} \quad (i = 1, ..., N),$$

which should be computed on large spatial grids and many times in the course of iterations on nonlinearity. Recall that the so-called Green iteration for solving the electronic Schrödinger and the Hartree-Fock/Kohn-Sham equations includes the Yukawa convolving kernel $\frac{e^{-\lambda\|\cdot\|}}{\|\cdot\|}$ in $\mathbb{R}^d$, $d \geq 3$ (see [11, 2, 13] for more detailes).

In the present paper, we describe an efficient approximation method for multi-dimensional convolution in $\mathbb{R}^d$ discretised via a collocation scheme over n^d spatial grid points. In the traditional FFT-based methods on equidistant tensor-product grids one arrives at severe computational problems of linear complexity in the volume, $O(n^d \log n)$. Using adaptive grids in higher dimensions seems to be technically troublesome and may run into implementational difficulties in the real-life applications. Notice that the calculation of the 3D FFT on $n \times n \times n$ grids is practically limited by the problem size $n \lesssim 512$, while our current implementation of the *fast tensor convolution transform* (FTCT) allows much larger 3D grids of size $n \leq 1.6 \cdot 10^4$ (MATLAB 7.3). The next table shows the advantage of the proposed FTCT method compared with those based on 3D FFT. We present the CPU time for a high accuracy computation of the Hartree potential for the H_2O molecule [16]. The CPU time for FFT-based scheme with $n \geq 1024$ is obtained by extrapolation.

n^3	64^3	128^3	256^3	512^3	1024^3	2048^3	4096^3	8192^3
$3D$ FFT (sec)	0.41	4.3	55.4	582.8	~ 6000	–	–	~ 70 days
$Conv_{CC}$ (sec)	1.9	1.7	6.1	6.1	35.0	35.0	246.0	769.0

In the recent years the idea of tensor-structured approximation has been recognised as the promising approach to relax the curse of dimensionality for representation of multivariate operators and functions, cf. [1, 2, 10, 8, 12, 9, 15, 18, 13]. Following this concept, we propose to combine the model reduction techniques by low tensor-rank approximation of the convolving d-th order tensors with the fast 1D discrete convolution applied to uniform or composite refined grids (cf. §3.2-3.4 below). This approach reduces the volume integration to a few independent univariate linear operations with linear scaling in d, and leading to the FTCT of complexity

$$O(dR_1R_2n\log n) \ll n^d,$$

where $R_1, R_2 \in \mathbb{N}$ are the so-called separation (tensor) ranks of the convolving functions (see §2.3). Accomplished with the efficient multi-linear algebra (MLA) operations via the "rank truncation" (cf. [3, 4, 12, 15, 16, 6, 17, 18, 20, 22]), this method can be applied for solving high dimensional equations which include multi-dimensional operator calculus. Recall that for a class of applications involving discretized analytic functions we take for granted the theoretical separation-rank estimate $R = O(\log 1/\varepsilon \log n)$, see [9] - [13]. For Green's kernels we make use of the *sinc*-approximation accomplished with the algebraic rank recompression (see §4.2).

In the case of unstructured tensor-product grids, the 1D convolution can be calculated directly in $O(n^2)$ operations. Since unstructured grids normally have moderate grid-size n, the direct $O(n^2)$-calculation leads to the acceptable total cost in multi-dimensional perspective, $O(dR_1R_2n^2)$. In some cases one can apply the fast $O(n\log^q n)$-convolution for a class of 1D hierarchical grids presented in [7] (corresponds to the Galerkin approximation). Algorithms and numerical aspects of the discrete tensor-product convolution on general non-uniform grids in $\mathbb{R}^d$ were briefly discussed in [15]. The computational efficiency of the tensor product convolution in electronic structure calculations is demonstrated in [14, 13, 16, 6].

We only consider piecewise constant approximations to minimise the technicalities, extension to higher order elements is straightforward. For these simple basis functions, our tensor-product collocation scheme defined on n^d grid-points is proven to provide the accuracy of order $O(h^2)$ with the grid parameter $h = O(1/n)$. Simple improvement via the Richardson extrapolation leads to an $O(h^3)$-approximation. It is worth to note that the physically relevant functionals applied to the convolution transform (say, scalar products, the Coulomb integrals, the Rayleigh quotients) are approximated with the same error $O(h^3)$.

The main results of the paper can be summarised as follows:

- In Section 2 we prove the $O(h^2)$ error bound (superconvergence) for the collocation convolution scheme by piecewise constant elements (see Thm. 2.2) and justify the Richardson extrapolation on a sequence of grids, which effectively reduces the error to $O(h^3)$ (see Thm. 2.3).

- In Section 3 the efficient tensor-product convolution method is presented and analysed for different rank-structured tensor formats (Algorithms 1, 2, 1′, Lemmata 3.4 and 3.5). This method applied on the uniform/composite grids of size n^d scales linear-logarithmically in n.

- We present the $O(n\log n)$ convolution method $O(n\log n)$ in the case of 1D composite grids and give related numerical illustrations (Algorithm 3, Lemma 3.6).

- In §4.2 the heuristic rank reduction scheme for the *sinc*-quadrature approximations of the convolving kernels is described in Algorithm 4 and supported by numerical examples.
- Numerical illustrations for the 3D convolution of the Newton potential with quantum chemistry data confirm the approximation theory and complexity bounds (see §4.3).

2 Discretisation of the convolution transform

The multi-dimensional convolution in $L^2(\mathbb{R}^d)$ is defined by the integral transform

$$w(x) := (f \star g)(x) := \int_{\mathbb{R}^d} f(y)g(x-y)dy \quad f, g \in L^2(\mathbb{R}^d), \quad x \in \mathbb{R}^d. \tag{2.1}$$

We are interested in approximate computation of $f \star g$ in some fixed box $\Omega = [-A, A]^d$, assuming that the convolving function f has a support in $\Omega' := [-B, B]^d \subset \Omega$ $(B < A)$, i.e., $supp\, f \subset \Omega'$. In electronic structure calculations the convolving function f may represent electron orbitals or electron densities which normally have an exponential decay.

The common example of the convolving kernel g is given by the restriction of the fundamental solution of an elliptic operator in $\mathbb{R}^d$. For example, in the case of the Laplacian in $\mathbb{R}^d$, $d \geq 3$, we have

$$g(x) = c(d)/\|x\|^{d-2}, \quad x = (x_1, ..., x_d) \in \mathbb{R}^d, \quad \|x\| = \sqrt{x_1^2 + ... + x_d^2},$$

where $c(d) = -2^{\frac{d}{4-d}}/\Gamma(d/2-1)$. This example will be considered in more details.

2.1 Approximating translation invariant integral operators

There are three commonly used discretisation methods for the integral operators: the so-called Nyström, collocation and Galerkin type schemes. For the sake of simplicity, first, we consider the case of uniform grids.

Introduce the equi-distant tensor-product lattice $\omega_{\mathbf{d}} := \omega_1 \times ... \times \omega_d$ of size $h = 2A/n$ by setting $\omega_\ell := \{-A + (k-1)h : k = 1, ..., n+1\}$, where for the sake of convenience $n = 2p$, $p \in \mathbb{N}$, and define the tensor-product index set $\mathcal{I} := \{1, ..., n\}^d$. Hence $\Omega = \cup_{\mathbf{i} \in \mathcal{I}} \Omega_{\mathbf{i}}$ becomes the union of closed boxes $\Omega_{\mathbf{i}} = \bigotimes_{\ell=1}^{d} \Omega_{i_\ell}$ with intervals

$$\Omega_{i_\ell} := \{x_\ell \,:\, x_\ell \in [-A + (i_\ell - 1)h, -A + i_\ell h]\} \subset \mathbb{R}, \quad (\ell = 1, ..., d). \tag{2.2}$$

The *Nyström type scheme* leads to simple discretisation

$$(f \star g)(x_{\mathbf{j}}) \approx h^d \sum_{\mathbf{i} \in \mathcal{I}} f(y_{\mathbf{i}})g(x_{\mathbf{j}} - y_{\mathbf{i}}), \quad \mathbf{j} \in \mathcal{I},$$

where, for the ease of presentation, the evaluation points $x_{\mathbf{j}}$, and the collocation points $y_{\mathbf{i}}$, $\mathbf{i}, \mathbf{j} \in \mathcal{I}$, are assumed to be located on the same cell-centred tensor-product grid corresponding to $\omega_{\mathbf{d}}$. The Nyström type scheme applies to the continuous functions f, g which leads to certain limitations in the case of singular kernels g.

The *collocation-projection discretisation* can be applied to a much more general class of integral operators than the Nyström methods including Green's kernels with the diagonal singularity, say to the Newton potential $g(x) = 1/\|x\|$. We consider the case of tensor-product piecewise constant basis functions $\{\phi_{\mathbf{i}}\}$ associated with $\omega_{\mathbf{d}}$, so that $\phi_{\mathbf{i}} = \chi_{\Omega_{\mathbf{i}}}$ is the characteristic function of $\Omega_{\mathbf{i}}$,

$$\phi_{\mathbf{i}}(x) = \prod_{\ell=1}^{d} \phi_{i_\ell}(x_\ell), \quad \text{where} \quad \phi_{i_\ell} = \chi_{\Omega_{i_\ell}}. \tag{2.3}$$

Let $x_{\mathbf{m}} \in \omega_{\mathbf{d}}$ be the set of collocation points with $\mathbf{m} \in \mathcal{M}_n := \{1, ..., n+1\}^d$ (we use the notation $\mathcal{M}_n = \mathcal{M}$ if there is no confusion), and let $f_{\mathbf{i}}$ be the representation coefficients of f in $\{\phi_{\mathbf{i}}\}$,

$$f(y) \approx \widetilde{f}(y) := \sum_{\mathbf{i}\in\mathcal{I}} f_{\mathbf{i}} \phi_{\mathbf{i}}(y).$$

In the following we specify the coefficients as $f_{\mathbf{i}} = f(y_{\mathbf{i}})$, where $y_{\mathbf{i}}$ is the midpoint of $\Omega_{\mathbf{i}}$, $\mathbf{i} \in \mathcal{I}$. We consider the following *discrete collocation-projection scheme*

$$f \star g \approx \{w_{\mathbf{m}}\}, \quad w_{\mathbf{m}} := \sum_{\mathbf{i}\in\mathcal{I}} f_{\mathbf{i}} \int_{\mathbb{R}^d} \phi_{\mathbf{i}}(y) g(x_{\mathbf{m}} - y) dy, \quad x_{\mathbf{m}} \in \omega_{\mathbf{d}}, \ \ \mathbf{m} \in \mathcal{M}. \tag{2.4}$$

Pointwise evaluation of this scheme requires $O(n^{2d})$ operations. In the case of equidistant grids the computational complexity can be reduced to $O(n^d \log n)$ by applying the multi-dimensional FFT.

To transform the collocation scheme (2.4) to the discrete convolution, we precompute the collocation coefficients

$$g_{\mathbf{i}} = \int_{\mathbb{R}^d} \phi_{\mathbf{i}}(y) g(-y) dy, \quad \mathbf{i} \in \mathcal{I}, \tag{2.5}$$

define the d-th order tensors $\mathcal{F} = \{f_{\mathbf{i}}\}, \mathcal{G} = \{g_{\mathbf{i}}\} \in \mathbb{R}^{\mathcal{I}}$, and introduce the d-dimensional *discrete convolution*

$$\mathcal{F} \star \mathcal{G} := \{z_{\mathbf{j}}\}, \quad z_{\mathbf{j}} := \sum_{\mathbf{i}} f_{\mathbf{i}} g_{\mathbf{j}-\mathbf{i}+\mathbf{1}}, \quad \mathbf{j} \in \mathcal{J} := \{1, ..., 2n-1\}^d, \tag{2.6}$$

where the sum is over all $\mathbf{i} \in \mathcal{I}$ which lead to legal subscripts for $g_{\mathbf{j}-\mathbf{i}+\mathbf{1}}$, $\mathbf{j} - \mathbf{i} + \mathbf{1} \in \mathcal{I}$. Specifically, for $j_\ell = 1, ..., 2n-1$,

$$i_\ell \in [max(1, j_\ell + 1 - n), min(j_\ell, n)], \quad \ell = 1, ..., d.$$

The discrete convolution can be gainfully applied to fast calculation of $\{w_{\mathbf{m}}\}_{\mathbf{m}\in\mathcal{M}}$ in the collocation scheme (2.4) as shown in the following statement.

Proposition 2.1 *The discrete collocation scheme $\{w_{\mathbf{m}}\}$, $\mathbf{m} \in \mathcal{M}$, is obtained by copying the corresponding portion of $\{z_{\mathbf{j}}\}$ from (2.6), centred at $\mathbf{j} = \mathbf{n} = n^{\otimes d}$,*

$$\{w_{\mathbf{m}}\} = \{z_{\mathbf{j}}\}_{|\mathbf{j}=\mathbf{j}_0+\mathbf{m}}, \quad \mathbf{m} \in \mathcal{M}, \quad \mathbf{j}_0 = \mathbf{n}/2.$$

Proof. In the 1D case we have

$$z(1) = f(1) \cdot g(1), \quad z(2) = f(1) \cdot g(2) + f(2) \cdot g(1), \ ...,$$

$$z(n) = f(1) \cdot g(n) + f(2) \cdot g(n-1) + ... + f(n) \cdot g(1), \ ..., z(2n-1) = f(n) \cdot g(n).$$

Then we find that elements $\{w_m\}$ coincide with $\{z_j\}_{|j=j_0+m}$, $m \in \mathcal{M}$, $j_0 = n/2$. The general case $d \geq 1$ can be justified by applying the above argument to each spatial variable. ■

The *Galerkin method* of discretisation reads as follows

$$f \star g \approx \sum_{\mathbf{i}, \mathbf{j}-\mathbf{i}+\mathbf{1} \in \mathcal{I}, \, \mathbf{j} \in \mathbf{j}_0 + \mathcal{M}} f_{\mathbf{i}} g_{\mathbf{j}-\mathbf{i}+1} \quad \text{with} \quad g_{\mathbf{j}-\mathbf{i}+1} := \int_{\mathbb{R}^d} \phi_{\mathbf{j}}(x) \phi_{\mathbf{i}}(y) g(x-y) dx dy$$

and with the choice $f_{\mathbf{i}} = \langle f, \phi_{\mathbf{i}} \rangle_{L^2}$. The Galerkin scheme is known as the most convenient for theoretical error analysis. However, compared with the collocation method, it has higher implementational cost because of the presence of double integration. Hence classical discretisation methods mentioned above may differ from each other by construction of the tensor-product decompositions. To keep a reasonable compromise between the numerical complexity of the scheme and its generality, in the following we focus on the collocation method by simple low order finite elements.

2.2 $O(h^2)$- and $O(h^3)$-error bounds

In the case of piecewise constant basis functions we prove the error bound $O(h^2)$ for the collocation scheme and then present a more refined error analysis which justifies the Richardson extrapolation method on a sequence of grids providing the better approximation error $O(h^3)$. Such an extrapolation, when available, allows a substantial reduction of the approximation error without extra cost. It is worth to note that the Richardson extrapolation can also be applied to some functionals of the convolution product, say to eigenvalues of the operator that includes the discrete convolution.

We use the multivariate Taylor expansion to find a local polynomial approximation of order m for a function with certain smoothness. Let us suppose that $f \in C^m(\mathbb{R}^d)$. The *Taylor polynomial* of order m evaluated at y is given by

$$T_y^m f(x) := \sum_{|\alpha|<m} \frac{1}{\alpha!} D^\alpha f(y)(x-y)^\alpha, \quad x, y \in \mathbb{R}^d,$$

where $\alpha = (\alpha_1, ..., \alpha_d)$ is an d-tuple of nonnegative integers, $x^\alpha = \prod_{\ell=1}^d x_\ell^{\alpha_\ell}$, $\alpha! = \prod_{\ell=1}^d \alpha_\ell!$ and $|\alpha| = \sum_{\ell=1}^d \alpha_\ell$. We restrict to the case of m-times continuously differentiable functions. For a given hypercube $B \in \mathbb{R}^d$ of size H, let $f \in C^m(B)$. We apply the Taylor expansion at the point $y \in B$ in the form

$$f(x) = T_y^m f(x) + R_y^{(m)}(x), \quad x \in B \tag{2.7}$$

with

$$R_y^{(m)}(x) := m \sum_{|\alpha|=m} (x-y)^\alpha \int_0^1 \frac{1}{\alpha!} s^{m-1} D^\alpha f(x + s(y-x)) ds.$$

In the following we need the standard error estimate

$$\|f(x) - T_y^m f(x)\|_{L^\infty(B)} \leq C_{m,d} H^m \|f\|_{C^m(B)}. \tag{2.8}$$

We recall that continuous Fourier transform in $\mathbb{R}^d$ is given by

$$\mathcal{F}(f)(\kappa) := \int_{\mathbb{R}^d} f(x) e^{-i\langle \kappa, x\rangle} dx, \quad \kappa \in \mathbb{R}^d.$$

Theorem 2.2 *Let $f \in C^2(\Omega)$ and let $g \in L^1(\Omega)$. Furthermore, we assume that there exist $\mu \geq 1$ and $\beta > 0$, such that*

$$|\mathcal{F}(g)(\kappa)| \leq C/\|\kappa\|^{\mu} \quad \text{as } \|\kappa\| \to \infty, \ \ \kappa \in \mathbb{R}^d \tag{2.9}$$

and

$$|\nabla_y g(x-y)| \leq C/\|x-y\|^{\beta} \quad \text{for } x, y \in \Omega, \ \ x \neq y. \tag{2.10}$$

Then there is a constant $C > 0$ independent of h such that for w defined in (2.1), and for $w_{\mathbf{m}}$ defined in (2.4), we have

$$|w(x_{\mathbf{m}}) - w_{\mathbf{m}}| \leq Ch^2, \quad \mathbf{m} \in \mathcal{M}. \tag{2.11}$$

Proof. Introduce the "local" interpolation error by

$$\delta_{\mathbf{i}}(y) = (f(y) - f(y_{\mathbf{i}}))\phi_{\mathbf{i}}(y), \quad y \in \Omega \quad \text{with } \ supp(\delta_{\mathbf{i}}) = \Omega_{\mathbf{i}}.$$

Define the error function as

$$E(x) := w(x) - \widetilde{f} \star g(x) = \sum_{\mathbf{i} \in \mathcal{I}} \delta_{\mathbf{i}} \star g(x) \quad \text{with } \ \widetilde{f} = \sum_{\mathbf{i} \in \mathcal{I}} f(y_{\mathbf{i}})\phi_{\mathbf{i}}.$$

For any fixed $\mathbf{i} \in \mathcal{I}$, we will estimate the individual term of the total error, $E_{\mathbf{i}}(x) = \delta_{\mathbf{i}} \star g(x)$. To that end let us apply the Taylor expansion (2.7) on $B = \Omega_{\mathbf{i}}$ with $m = 2$ to obtain

$$\delta_{\mathbf{i}}(y) = \langle \nabla f(y_{\mathbf{i}}), y - y_{\mathbf{i}} \rangle + R^{(2)}_{y_{\mathbf{i}}}(y), \quad y \in B.$$

Step 1. It is easy to see that (2.8) implies

$$\|R^{(2)}_{y_{\mathbf{i}}}(\cdot)\|_{L^\infty(B)} \leq Ch^2,$$

hence the condition $g \in L^1(\Omega)$ leads to

$$\left\| \sum_{\mathbf{i} \in \mathcal{I}} R^{(2)}_{y_{\mathbf{i}}} \star g \right\|_{L^\infty(\Omega)} \leq Ch^2 \|g\|_{L^1(\Omega)} = O(h^2). \tag{2.12}$$

Next we analyse the rest part of $E(x)$ at some fixed collocation point $x_{\mathbf{m}}$, $\mathbf{m} \in \mathcal{M}$.

Step 2. Let us consider the contribution to the error from the individual terms $\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}} \rangle \star g(\cdot)$ for all

$$\mathbf{i} \in \Sigma_{\mathbf{m}} := \{\mathbf{j} \in \mathcal{I} : x_{\mathbf{m}} \in \Omega_{\mathbf{j}}\}.$$

To that end we estimate the Fourier transform of such terms,

$$\mathcal{F}\left(\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}} \rangle \star g(\cdot)\right) = \mathcal{F}\left(\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}} \rangle\right) \cdot \mathcal{F}(g), \tag{2.13}$$

where $\mathcal{F}(g)$ is understood as a temporary distribution. Since $g \in L^1(\Omega)$, we have

$$\|\mathcal{F}(g)\|_{L^\infty(\mathbb{R}^d)} \leq C\|g\|_{L^1}.$$

Furthermore, we will need a "directional" estimate on $|\mathcal{F}(g)|$. At this point we apply the classical inequality of the harmonic and geometric mean: let $a_1, ..., a_d$ be the positive real numbers, then

$$\frac{d}{\frac{1}{a_1} + ... + \frac{1}{a_d}} \le \sqrt[d]{a_1 a_2 ... a_d}.$$

Let us set $a_k = 1/x_k^2$ for $x \in \mathbb{R}^d$, which leads to

$$\frac{1}{\|x\|} = \frac{1}{\sqrt{x_1^2 + ... + x_d^2}} \le \frac{1}{\sqrt{d}} \prod_{\ell=1}^{d} \frac{1}{\sqrt[d]{|x_\ell|}}.$$

Hence, the assumption on the decay property (2.9) implies the desired "directional" bound

$$|\mathcal{F}(g)(\kappa)| \le \frac{C}{\|\kappa\|^\mu} \le \frac{C}{\sqrt{d^\mu}} \prod_{\ell=1}^{d} \frac{1}{|\kappa_\ell|^{\mu/d}}. \tag{2.14}$$

Furthermore, for the first factor in the right-hand side of (2.13) we are able to prove

$$|\mathcal{F}\left(\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}} \rangle\right)| \le C h^{d+2} P_{\mathbf{i}}, \quad P_{\mathbf{i}} > 0 \tag{2.15}$$

with the uniformly bounded sum $\sum_{\mathbf{i} \in \mathcal{I}} P_{\mathbf{i}} \le C$. In fact, due to separability of $\mathcal{F}$ in $\mathbb{R}^d$ with respect to the 1-dimensional Fourier transforms $\mathcal{F}_k$ in variable y_k ($k = 1, ..., d$), one can represent

$$\mathcal{F}\left(\langle \nabla f(y_{\mathbf{i}}), y - y_{\mathbf{i}} \rangle\right) = \langle \nabla f(y_{\mathbf{i}}), U_{\mathbf{i}} \rangle, \quad U_{\mathbf{i}} \in (L^\infty(\mathbb{R}))^d$$

with

$$(U_{\mathbf{i}})_k(\kappa) = \mathcal{F}_k(\chi_{\Omega_{i_k}}(\cdot - y_{\mathbf{i},k}))(\kappa_k) \prod_{\ell=1,\, \ell \ne k}^{d} \mathcal{F}_\ell(\chi_{\Omega_{i_\ell}})(\kappa_\ell).$$

For each fixed $k = 1, ..., d$, consider the individual term

$$\mathcal{F}_k(\chi_{\Omega_{i_k}}(t - y_{\mathbf{i},k}))(\tau) = e^{-i y_{\mathbf{i},k} \tau} \mathcal{F}(\chi_{[-h/2,h/2]} t)(\tau)$$

with $\mathcal{F}$ being the Fourier transform in $\mathbb{R}$, and derive

$$\begin{aligned} e_h(\tau) := \mathcal{F}(\chi_{[-h/2,h/2]} t)(\tau) &= \left[\frac{\sin(\tau y)}{\tau^2} - \frac{y \cos(\tau y)}{\tau} \right]_{-h/2}^{h/2} \\ &= \frac{2 \sin(\tau h/2)}{\tau^2} - \frac{h \cos(\tau h/2)}{\tau}. \end{aligned}$$

Hence we have the asymptotic expansions

$$e_h(\tau) = \frac{\tau h^3}{12} + O(h^5) \quad \text{as } |\tau h| \le O(1)$$

and

$$|e_h(\tau)| \le C(\frac{h}{\tau} + \frac{1}{\tau^2}) \quad \text{as } |\tau| \to \infty.$$

We apply (2.14) with $\mu > 0$, take the directional factor

$$g_1(\tau) = \min\{1, 1/\tau^{\mu/d}\},$$

and then consider the parametric function

$$p_h(\tau) := g_1(\tau)\left(\frac{2\sin(\tau h/2)}{\tau^2} - \frac{h\cos(\tau h/2)}{\tau}\right).$$

We can prove by the scaling argument that

$$|p_h(\tau)| = Ch^{2+\mu/d}P(u), \quad \text{with } C = C(\mu, d),$$

where, with $u = \tau h/2$,

$$P(u) = [\frac{\sin u}{u^2} - \frac{\cos u}{u}]\min\{h^{-\mu/d}, u^{-\mu/d}\} \in L^1(\mathbb{R}).$$

The standard scaling argument leads to the relation

$$\|p_h(\tau)\|_{L^1} \le Ch^{1+\mu/d}\|P(u)\|_{L^1}. \tag{2.16}$$

Likewise, we have

$$\mathcal{F}(\chi_{[-h/2,h/2]})(\tau) = \left[\frac{\sin(\tau y)}{\tau}\right]_{-h/2}^{h/2} = \frac{2\sin(\tau h/2)}{\tau},$$

$$q_h(\tau) := g_1(\tau)\mathcal{F}(\chi_{[-h/2,h/2]})(\tau) = Ch^{1+\mu/d}sinc(u)\min\{h^{-\mu/d}, u^{-\mu/d}\}, \tag{2.17}$$

which implies

$$\|q_h(\tau)\|_{L^1} \le Ch^{\mu/d}\|sinc(u)\min\{h^{-\mu/d}, u^{-\mu/d}\}\|_{L^1}.$$

With fixed index $\mathbf{i} \in \Sigma_{\mathbf{m}}$, we apply the inverse Fourier transform $\mathcal{F}^{-1}$ to (2.13), then make use of the bounds (2.14), (2.16) and (2.17) to obtain

$$\begin{aligned} |\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}}\rangle \star g(\cdot)| &\le \|\mathcal{F}(\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}}\rangle)\cdot \mathcal{F}(g)\|_{L^1} \\ &\le \|\,|\mathcal{F}(\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}}\rangle)|\cdot|\mathcal{F}(g)|\|_{L^1} \\ &\le Cd\|p_h(\tau)\|_{L^1}\prod_{\ell=2,\dots,d}\|q_h(\tau_\ell)\|_{L^1} \\ &\le Cdh^{1+\mu/d}\prod_{\ell=2,\dots,d}h^{\mu/d} = Cdh^{1+\mu}. \end{aligned} \tag{2.18}$$

Summing over $\mathbf{i} \in \Sigma_{\mathbf{m}}$ leads to the desired "local" estimate of order $Cd\,2^d\,h^{1+\mu}$.

Step 3. In the final step, we estimate the contribution from "nondiagonal" terms corresponding to $\mathbf{i} \in \mathcal{I} \setminus \Sigma_{\mathbf{m}}$. For such terms we just apply the Taylor expansion around $y_{\mathbf{i}}$ with $m = 2$ to the convolving kernel $g(x_{\mathbf{m}} - y)$, $y \in B = \Omega_{\mathbf{i}}$, and take into account (2.10), which leads to the bound (with $n = 1/h$, $\mathbf{n} = n^{\otimes d}$ and $\beta \ne d$)

$$\left|\sum_{\mathbf{i}\in\mathcal{I}\setminus\Sigma_{\mathbf{m}}}\left(\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}}\rangle \star g\right)(x_{\mathbf{m}})\right| =$$

$$\left|\sum_{\mathbf{i}\in\mathcal{I}\setminus\Sigma_{\mathbf{m}}}\int_{\Omega_{\mathbf{i}}}\langle \nabla f(y_{\mathbf{i}}), y - y_{\mathbf{i}}\rangle g(x_{\mathbf{m}} - y)dy\right| =$$

$$|\sum_{\mathbf{i}\in\mathcal{I}\setminus\Sigma_{\mathbf{m}}}\int_{\Omega_{\mathbf{i}}}\langle \nabla f(y_{\mathbf{i}}), y - y_{\mathbf{i}}\rangle\left(\langle \nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}}), y - y_{\mathbf{i}}\rangle + R^{(2)}_{y_{\mathbf{i}}}(y)\right)dy| \le$$

$$\sum_{\mathbf{i}\in\mathcal{I}\setminus\Sigma_{\mathbf{m}}}\int_{\Omega_{\mathbf{i}}}|\langle\nabla f(y_{\mathbf{i}}),y-y_{\mathbf{i}}\rangle|\cdot|y-y_{\mathbf{i}}|/\|x_{\mathbf{m}}-y_{\mathbf{i}}\|^{\beta}dy+O(h^3)=$$
$$Cd\sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}}\frac{h^{d+2}}{|\mathbf{k}h|^{\beta}}=C\frac{h^{2+d}}{h^{\beta}}\sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}}\frac{1}{|\mathbf{k}|^{\beta}}\leq$$
$$C\frac{h^{2+d}}{h^{\beta}}\cdot h^{\beta-d}=Ch^2.$$

Combining this result with (2.12) completes the proof. ■

Theorem 2.2 indicates the "superconvergence property" for low order elements in the case of smooth enough convolving functions. To illustrate the applicability of above Theorem we notice that the fundamental solution of the Laplace operator in $\mathbb{R}^d$ is given by $g(x)=c(d)/\|x\|^{d-2}$ with the Fourier transform $\mathcal{F}(g)=C/\|\kappa\|^2$. Hence Theorem 2.2 applies with $\beta=d-1$, $\mu=2$. It also applies to the Yukawa potential $g(x)=e^{-\lambda\|x\|}/\|x\|$ for $x\in\mathbb{R}^3$ with any $\mu\geq 1$ and with $\beta=2$.

The approximation error $O(h^2)$ can be improved up to $O(h^3)$ using the Richardson extrapolation scheme on a sequence of grids. We show that the linear combination of solutions $w_{\mathbf{m}}^{(n)}$, $\mathbf{m}\in\mathcal{M}_n$, and $w_{\mathbf{m}}^{(2n)}$, $\mathbf{m}\in\mathcal{M}_{2n}$, corresponding to the grid-size n and $2n$, respectively, ensures the expected high order approximation.

Theorem 2.3 *Let $f\in C^3(\Omega)$, and assume that the conditions of Theorem 2.2 are satisfied with $\mu\geq 2$ and $\beta\neq d$ (technical condition). Moreover, suppose that*

$$|\nabla_y^2 g(x-y)|\leq C/\|x-y\|^{\gamma}\quad \textit{with}\quad \gamma>0. \tag{2.19}$$

Then for $\mathbf{m}\in\mathcal{M}_n$, there is a constant $C>0$ independent of h such that

$$(4w_{\mathbf{m}}^{(2n)}-w_{\mathbf{m}}^{(n)})/3=w(x_{\mathbf{m}})+\eta_{\mathbf{m},n},\quad \eta_{\mathbf{m},n}\in\mathbb{R}\quad \textit{with}\ |\eta_{\mathbf{m},n}|\leq Ch^3. \tag{2.20}$$

Proof. Using the notations from the previous proof, we rewrite relation (2.20) in terms of the error function

$$(4E^{(2n)}(x_{\mathbf{m}})-E^{(n)}(x_{\mathbf{m}}))/3=\eta_{\mathbf{m},n},\quad \mathbf{m}\in\mathcal{M}_n.$$

We start from arguments which are similar to those in the proof of Theorem 2.2. To represent the local error, we apply the Taylor expansion (2.7) with $m=3$ on $B=\Omega_{\mathbf{i}}$ for $\mathbf{i}\in\mathcal{M}_n$ to obtain

$$\delta_{\mathbf{i}}^{(n)}(y)=\langle\nabla f(y_{\mathbf{i}}),y-y_{\mathbf{i}}\rangle+2\sum_{|\alpha|=2}\frac{1}{\alpha!}(y-y_{\mathbf{i}})^{\alpha}D^{\alpha}f(y_{\mathbf{i}})+R_{y_{\mathbf{i}}}^{(3)}(y),\quad y\in B. \tag{2.21}$$

Step 1. It is easy to see that (2.8) implies

$$\|R_{y_{\mathbf{i}}}^{(3)}(\cdot)\|_{L^{\infty}(B)}\leq Ch^3,$$

hence the condition $g\in L^1(\Omega)$ again leads to

$$\left\|\sum_{\mathbf{i}\in\mathcal{I}}R_{y_{\mathbf{i}}}^{(3)}\star g\right\|_{L^{\infty}(\Omega)}\leq C\|g\|_{L^1(\Omega)}h^3=O(h^3). \tag{2.22}$$

Next we analyse the remaining couple of terms in $E^{(n)}(x)$ (resp. $E^{(2n)}(x)$) at some fixed collocation point $x_{\mathbf{m}}$, $\mathbf{m} \in \mathcal{M}_n$.

Step 2(a). The contribution to the error from the individual terms $\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}} \rangle \star g(\cdot)$ for all $\mathbf{i} \in \Sigma_{\mathbf{m}} := \{\mathbf{j} \in \mathcal{I} : x_{\mathbf{m}} \in \Omega_{\mathbf{j}}\}$ can be estimated by similar argument as in the proof of Theorem 2.2 (see Step 2). At the final estimate in (2.18), we take into account that $\mu \geq 2$, which leads to the total bound $O(h^3)$ for the "local" terms.

Step 2(b). Now we analyse the contribution from "nondiagonal" terms (corresponding to $\mathbf{i} \in \mathcal{I} \setminus \Sigma_{\mathbf{m}}$) in the linear part of the Taylor expansion. For such terms we further apply the Taylor expansion to the convolving kernel $g(x_{\mathbf{m}} - y)$, $y \in B = \Omega_{\mathbf{i}}$, around $y_{\mathbf{i}}$ with $m = 2$, and take into account (2.10), which leads to the equation (with technical assumption $\beta \neq d$)

$$\left| \sum_{\mathbf{i} \in \mathcal{I} \setminus \Sigma_{\mathbf{m}}} \left(\langle \nabla f(y_{\mathbf{i}}), \cdot - y_{\mathbf{i}} \rangle \star g \right)(x_{\mathbf{m}}) \right| =$$
$$\left| \sum_{\mathbf{i} \in \mathcal{I} \setminus \Sigma_{\mathbf{m}}} \int_{\Omega_{\mathbf{i}}} \langle \nabla f(y_{\mathbf{i}}), y - y_{\mathbf{i}} \rangle \langle \nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}}), y - y_{\mathbf{i}} \rangle dy \right| + O(h^3).$$

For given $x_{\mathbf{m}}$, let us fix some box $\Omega_{\mathbf{i}}^{(n)}$, $\mathbf{i} \in \mathcal{I} \setminus \Sigma_{\mathbf{m}}$, and consider the contribution from this box corresponding to the diadically refined grid, so that $\Omega_{\mathbf{i}}^{(n)} = \bigcup_{k=1}^{8} \Omega_{\mathbf{i}_k}^{(2n)}$. Simple calculations show that

$$4 \sum_{k=1}^{8} \int_{\Omega_{\mathbf{i}_k}^{(2n)}} \langle \nabla f(y_{\mathbf{i}_k}), y - y_{\mathbf{i}_k} \rangle \langle \nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}_k}), y - y_{\mathbf{i}_k} \rangle dy -$$
$$\int_{\Omega_{\mathbf{i}}^{(n)}} \langle \nabla f(y_{\mathbf{i}}), y - y_{\mathbf{i}} \rangle \langle \nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}}), y - y_{\mathbf{i}} \rangle dy = O(h^3),$$

where we apply the Taylor expansion with $m = 1$ at point $y_{\mathbf{i}}$ to both gradients $\nabla f(y)$ and $\nabla_y g(x_{\mathbf{m}} - y)$. Summing up the above equation over $\mathbf{i} \in \mathcal{I} \setminus \Sigma_{\mathbf{m}}$ we complete the proof for the first order terms in the Taylor expansion.

Step 3. To analyse the contribution from second order terms in (2.21) we apply the Taylor expansion of order $m = 2$ to the gradient $\nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}_k})$ around the cell centre $y_{\mathbf{i}}$,

$$\nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}_k}) = \nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}}) + \langle \nabla_y^2 g(x_{\mathbf{m}} - y_{\mathbf{i}}), y_{\mathbf{i}_k} - y_{\mathbf{i}} \rangle + O(h^2).$$

This leads to the bound

$$4 \sum_{k=1}^{8} \int_{\Omega_{\mathbf{i}_k}^{(2n)}} \sum_{|\alpha|=2} \frac{1}{\alpha!} (y - \mathbf{i}_k)^\alpha D^\alpha f(y_{\mathbf{i}_k}) \langle \nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}_k}), y - y_{\mathbf{i}_k} \rangle dy -$$
$$\int_{\Omega_{\mathbf{i}}^{(n)}} \sum_{|\alpha|=2} \frac{1}{\alpha!} (y - y_{\mathbf{i}})^\alpha D^\alpha f(y_{\mathbf{i}}) \langle \nabla_y g(x_{\mathbf{m}} - y_{\mathbf{i}}), y - y_{\mathbf{i}} \rangle dy = O(h^3).$$

Now summation of the above equation over $\mathbf{i} \in \mathcal{I} \setminus \Sigma_{\mathbf{m}}$ completes our proof. ■

Remark 2.4 *The Newton potential in 3D, $g(x) = 1/\|x\|$, $x \in \mathbb{R}^3$, satisfies the conditions of Theorem 2.3 with $\mu = 2$ and $\beta = 2$.*

Notice that in the case $\beta = d$ some logarithmic terms in the error estimate may arise.

Below we give numerical examples for the Newton potential with $d = 3$.

3 Tensor approximation to discrete convolution

Recall that in the case of uniform grids the discrete convolution $\mathbb{R}^d$ can be implemented by d-dimensional FFT with linear cost in the volume size, $O(n^d \log n)$, which scales exponentially in d. To break down the curse of dimensionality, we represent the d-dimensional convolution product approximately in the low-rank tensor product formats. This reduces dramatically the computational cost to $O(dn \log n)$.

3.1 Orthogonal Tucker and canonical tensor decompositions

Data sparse representation of high order tensors is based on the Tucker, canonical and mixed models. A tensor is a multidimensional array,

$$V = [v_{i_1,\ldots,i_d} : i_\ell \in I_\ell] \in \mathbb{R}^{\mathcal{I}}, \quad \mathcal{I} = I_1 \times \ldots \times I_d,$$

with $\mathcal{I}$ being the tensor-product index set, where $I_\ell = \{1, ..., n_\ell\}$ (denote by $\mathbf{i}$ the d-tuple $(i_1, ..., i_d)$). It is an element of linear space $\mathbb{V}_\mathbf{n} = \otimes_{\ell=1}^d \mathbb{V}_\ell$ of real-valued (complex-valued) d-th order tensors with $\mathbb{V}_\ell = \mathbb{R}^{I_\ell}$, and equipped with the Euclidean *inner product* $\langle \cdot, \cdot \rangle : \mathbb{V}_\mathbf{n} \times \mathbb{V}_\mathbf{n} \to \mathbb{R}$, defined as

$$\langle V, U \rangle := \sum_{(i_1,\ldots,i_d)\in\mathcal{I}} v_{i_1\ldots i_d} u_{i_1\ldots i_d} \quad \text{for } V, U \in \mathbb{V}_\mathbf{n}. \tag{3.1}$$

Assume for simplicity that $dim\, \mathbb{V}_\ell = \#I_\ell = n$ for all $\ell = 1, ..., d$, then the number of entries in V amounts to n^d, hence growing exponentially in d.

To get rid of exponential scaling in the dimension approximate representations in some classes $\mathcal{S} \subset \mathbb{V}_\mathbf{n}$ of data-sparse "rank structured" tensors will be applied. The basic concept is a representation by a short-term sum of rank-1 tensors. Specifically, the *outer product* of vectors $t_\ell = \{t_{\ell,i_\ell}\}_{i_\ell \in I_\ell} \in \mathbb{V}_\ell$ $(\ell = 1, ..., d)$ forms the canonical rank-1 tensor

$$T \equiv [t_\mathbf{i}]_{\mathbf{i}\in\mathcal{I}} = t_1 \otimes ... \otimes t_d \in \mathbb{V}_\mathbf{n} \quad \text{with entries} \quad t_\mathbf{i} = t_{1,i_1} \cdots t_{d,i_d},$$

which requires only dn numbers to store it (now linear scaling in the dimension). In the case $d = 2$, the outer product of two vectors represents a rank-1 matrix.

Commonly used classes $\mathcal{S}$ of "rank structured" tensors are given by the Tucker, canonical and mixed models. *The rank-$(r_1, \ldots, r_d)$ Tucker representation* [21, 4] is based on subspaces $\mathbb{T}_\mathbf{n} := \otimes_{\ell=1}^d \mathbb{T}_\ell$ of $\mathbb{V}_\mathbf{n}$ for certain $\mathbb{T}_\ell \subset \mathbb{V}_\ell$ with $r_\ell := \dim \mathbb{T}_\ell \le n$. It is worth to note that usually subspaces $\mathbb{T}_\mathbf{n}$ are not fixed, and can be optimised via certain nonlinear approximation process.

Definition 3.1 *Given the vector-valued rank parameter $\mathbf{r} = (r_1, ..., r_d)$, we denote by $\boldsymbol{\mathcal{T}}_{\mathbf{r},\mathbf{n}} \equiv \boldsymbol{\mathcal{T}}_\mathbf{r} \subset \mathbb{V}_\mathbf{n}$ the subset of tensors in $\mathbb{V}_\mathbf{n}$ represented in the so-called Tucker format*

$$V_{(\mathbf{r})} = \sum\nolimits_{\nu_1=1}^{r_1} \cdots \sum\nolimits_{\nu_d=1}^{r_d} \beta_{\nu_1\ldots\nu_d}\, t_1^{\nu_1} \otimes \ldots \otimes t_d^{\nu_d} \tag{3.2}$$

with some vectors $t_\ell^{\nu_\ell} \in \mathbb{V}_\ell$ $(1 \le \nu_\ell \le r_\ell)$, which form the orthonormal basis of $\mathbb{T}_\ell := \mathrm{span}\{t_\ell^\nu\}_{\nu=1}^{r_\ell}$ $(\ell = 1, ..., d)$.

Conventionally, $r = \max_\ell\{r_\ell\}$ is called the Tucker rank. In our applications we have $r \ll n$, say $r = O(\log n)$. The coefficients tensor $\boldsymbol{\beta} = [\beta_{\nu_1 \dots \nu_d}] \in \mathbb{R}^{r_1 \times \dots \times r_d}$, that is an element of the dual (reciprocal) tensor space $\mathbb{B}_{\mathbf{r}}$, is called the *core tensor*. As long as the grid-size $\mathbf{n}$ is fixed, we will skip $\mathbf{n}$ in the notation $\boldsymbol{\mathcal{T}}_{\mathbf{r},\mathbf{n}}$. Introducing the (orthogonal) matrices $T^{(\ell)} = [t_\ell^{\nu_1} \dots t_\ell^{\nu_d}]$, we then use a shorter notation for the Tucker decomposition of $V \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$:

$$V = \boldsymbol{\beta} \times_1 T^{(1)} \times_2 T^{(2)} \dots \times_d T^{(d)},$$

where the symbol "$\times_\ell$" denotes a tensor-by-matrix contraction product along the mode ℓ.

Notice that if the subspaces $\mathbb{T}_\ell = \mathrm{span}\{t_\ell^\nu\}_{\nu=1}^{r_\ell}$ are fixed then we have the outer product representation

$$\boldsymbol{\mathcal{T}}_{\mathbf{r},\mathbf{n}} = \mathbb{T}_{\mathbf{n}} := \otimes_{\ell=1}^d \mathbb{T}_\ell.$$

In this case the approximation $V_{(\mathbf{r})} \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ of the target tensor V is given by the orthogonal projection of V onto the linear space $\mathbb{T}_{\mathbf{n}}$, that is

$$V_{(\mathbf{r})} = \sum_{\nu_1,\dots,\nu_d} \langle t_1^{\nu_1} \otimes \dots \otimes t_d^{\nu_d}, V \rangle \, t_1^{\nu_1} \otimes \dots \otimes t_d^{\nu_d}.$$

This property is crucial in the computation of the best orthogonal Tucker approximation, where the "optimal" subspaces $\mathbb{T}_\ell$ are recalculated within a nonlinear approximation process.

Definition 3.2 *Given the rank parameter $R \in \mathbb{N}$, we denote by $\boldsymbol{\mathcal{C}}_{R,\mathbf{n}} = \boldsymbol{\mathcal{C}}_R \subset \mathbb{V}_{\mathbf{n}}$ a set of tensors which can be represented in the canonical format*

$$V_{(R)} = \sum\nolimits_{\nu=1}^{R} \beta_\nu v_1^\nu \otimes \dots \otimes v_d^\nu, \quad \beta_\nu \in \mathbb{R}, \tag{3.3}$$

with normalised vectors $v_\ell^\nu \in \mathbb{V}_\ell$ ($\ell = 1, \dots, d$).

The minimal parameter R in (3.3) is called the rank (or canonical rank) of a tensor $V_{(R)}$.

To simplify the discussion, we further assume $r_\ell = r$ for all $\ell = 1, \dots, d$. The storage requirements for the Tucker (resp. canonical) decomposition is given by $r^d + drn$ (resp. $R + dRn$), where usually $r \ll R$. In turn, the maximal canonical rank of the Tucker representation is r^{d-1}. Since the Tucker core still presupposes the r^d storage space, we further consider a mixed (two-level) representation which gainfully combines the beneficial features of both the Tucker and canonical models. In the following, the grid-size $\mathbf{n}$ will be fixed.

Definition 3.3 *(The two-level Tucker-canonical format, cf. [12]). Given the rank parameters $\mathbf{r}, R$, we denote by $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}}$ the subclass of tensors in $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$ with the core $\boldsymbol{\beta}$ represented in the canonical format, $\boldsymbol{\beta} \in \boldsymbol{\mathcal{C}}_{R,\mathbf{r}} \subset \mathbb{B}_{\mathbf{r}}$. Clearly, we have the imbedding $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}} \subset \boldsymbol{\mathcal{C}}_R$.*

The target tensor $V \in \mathbb{V}_{\mathbf{n}}$ can be approximated by a sum of rank-1 tensors as in (3.2), $V_{(\mathbf{r})} \approx V$ (cf. tensor-product schemes (3.4), (3.5) below), or in (3.3), $V_{(R)} \approx V$ (cf. Algorithm 2), or using the format $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}}$ (cf. Algorithm 1).

To unify the complexity analysis of the MLA operations including tensors in $\mathcal{S} = \{\boldsymbol{\mathcal{T}}_{\mathbf{r}}, \boldsymbol{\mathcal{C}}_R, \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}}\}$, we will represent a tensor $V_{(R)} \in \boldsymbol{\mathcal{C}}_R$ in the standard form (3.2) with the diagonal core tensor $\boldsymbol{\beta} = diag\{b_1, \dots, b_R\}$, where the orthogonality of vectors $\{t_\ell^\nu\}_{\nu=1}^{r_\ell}$ ($\ell = 1, \dots, d$) is no longer required. We denote by $\#A$ the number of nonzero elements corresponding to the sparsity pattern of tensor A. In particular, for $V_{(R)} \in \boldsymbol{\mathcal{C}}_R$, we have

for the corresponding core tensor $\#\boldsymbol{\beta} = R$, while in general for $V_{(\mathbf{r})} \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$, one obtains $\#\boldsymbol{\beta} = \prod_{\ell=1}^{d} r_\ell$. In the special case $V_{(R)} \in \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}}$ we have $\#\boldsymbol{\beta} = dRr$.

Multilinear algebraic operations (including visualization) with tensors of order d ($d \geq 3$), can be reduced to the standard linear algebra by *unfolding of a tensor* into a matrix. The unfolding of a tensor along mode ℓ is a (unfolding) matrix of dimension $n_\ell \times (n_{\ell+1}...n_d n_1...n_{\ell-1})$, further denoted by $V_{(\ell)}$, whose columns are the respective fibers of V along the ℓ-th mode.

3.2 Tensor-product convolution on uniform grids

We notice that the multidimensional convolution product appears to be one of the most computationally elaborate MLA operations. In the present paper, the key idea is to calculate the d-dimensional convolution approximately using rank-structured tensor approximations. Recall that for given d-th order tensors $F, G \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$, represented by

$$F = \boldsymbol{\beta} \times_1 F^{(1)} \times_2 F^{(2)} ... \times_d F^{(d)}, \quad \text{and} \quad G = \boldsymbol{\gamma} \times_1 G^{(1)} \times_2 G^{(2)} ... \times_d G^{(d)},$$

the convolution product can be "separated" via (cf. [15])

$$F \star G := \sum_{\mathbf{k}=1}^{\mathbf{r}} \sum_{\mathbf{m}=1}^{\mathbf{r}} \beta_{k_1...k_d} \gamma_{m_1...m_d} \left(f_1^{k_1} \star g_1^{m_1} \right) \otimes ... \otimes \left(f_d^{k_d} \star g_d^{m_d} \right). \tag{3.4}$$

Computing 1D convolution $f_\ell^{k_\ell} \star g_\ell^{m_\ell} \in \mathbb{R}^{2n-1}$ in $O(n \log n)$ operations leads to the overall linear-logarithmic complexity in n,

$$\mathcal{N}_{T \star T} = O(dr^2 n \log n + \#\boldsymbol{\beta} \cdot \#\boldsymbol{\gamma}).$$

In general one might have $\#\boldsymbol{\beta} \cdot \#\boldsymbol{\gamma} = O(r^{2d})$, which may be restrictive even for moderate d.

Significant complexity reduction is observed if at least one of the convolving tensors can be represented by the canonical model. Letting $F \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$, $G \in \boldsymbol{\mathcal{C}}_R$, i.e., $\boldsymbol{\gamma} = diag\{\gamma_1, ..., \gamma_R\}$, we tensorize the convolution product as follows

$$F \star G = \sum_{\mathbf{k=1}}^{\mathbf{r}} \sum_{m=1}^{R} \beta_{k_1...k_d} \gamma_m \left(f_1^{k_1} \star g_1^{m} \right) \otimes ... \otimes \left(f_d^{k_d} \star g_d^{m} \right). \tag{3.5}$$

However, the calculation by (3.5) still scales exponentially in d, which leads to certain limitations in the case of higher dimensions.

To get rid of this exponential scaling, we propose to perform the convolution transform using the two-level tensor format, i.e., $F \in \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1,\mathbf{r}}}$ (see Definition 3.3) in such a way that the result $U = F \star G$ with $G \in \boldsymbol{\mathcal{C}}_{R_G}$ is represented in the two-level Tucker format $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1 R_G, \mathbf{r} R_G}}$. Recall that an explicit representation for $F \in \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1,\mathbf{r}}}$ is given by

$$F = \left(\sum\nolimits_{\nu=1}^{R_1} \beta_\nu z_1^\nu \otimes \ldots \otimes z_d^\nu \right) \times_1 F^{(1)} \times_2 F^{(2)} ... \times_d F^{(d)}, \tag{3.6}$$

so that we have the imbedding $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1,\mathbf{r}}} \subset \boldsymbol{\mathcal{C}}_{R_1,\mathbf{n}}$ with the corresponding (non-orthogonal) side-matrices $S^{(\ell)} = [F^{(\ell)} z_\ell^1 ... F^{(\ell)} z_\ell^{R_1}] \in \mathbb{R}^{n \times R_1}$, and scaling factors β_ν ($\nu = 1, ..., R_1$). Now we represent the tensor-product convolution in the two-level format

$$F \star G = \sum_{m=1}^{R_G} \gamma_m \left(\sum\nolimits_{\nu=1}^{R_1} \beta_\nu z_1^\nu \otimes \ldots \otimes z_d^\nu \right) \times_1 \left(F^{(1)} \star g_1^m \right) \times_2 ... \times_d \left(F^{(d)} \star g_d^m \right), \tag{3.7}$$

such that the above expansion can be evaluated by the following algorithm.

Algorithm 1. (d-dimensional tensor convolution of type $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1,\mathbf{r}}} \star \boldsymbol{\mathcal{C}}_{R_G,\mathbf{n}} \to \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1 R_G, \mathbf{r} R_G}}$).

1. Given $F \in \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1,\mathbf{r}}}$ with the core $\boldsymbol{\beta} = \sum_{\nu=1}^{R_1} \beta_\nu z_1^\nu \otimes \ldots \otimes z_d^\nu \in \boldsymbol{\mathcal{C}}_{R_1,\mathbf{r}}$, and $G \in \boldsymbol{\mathcal{C}}_{R_G,\mathbf{n}}$.

2. For $\ell = 1, ..., d$, compute the set of 1D convolutions $u_\ell^{k,m} = f_\ell^k \star g_\ell^m$ ($k = 1, ..., r$, $m = 1, ..., R_G$) of size $2n-1$, restrict the results onto the index set I_ℓ, and form the $n \times rR_G$ side-matrices $U^{(\ell)} = [U_1^{(\ell)} ... U_{R_G}^{(\ell)}]$, composed of the blocks $U_m^{(\ell)}$ with columns $u_\ell^{k,m}$ as $U_m^{(\ell)} = [f_\ell^1 \star g_\ell^m ... f_\ell^r \star g_\ell^m]$, all at the cost $O(drR_G n \log n)$.

3. Build the core tensor $\boldsymbol{\omega} = \text{blockdiag}\{\gamma_1 \boldsymbol{\beta}, ..., \gamma_R \boldsymbol{\beta}\}$ and represent the resultant two-level Tucker tensor in the form (storage demand is $R_G + R_1 + drR_1 + drR_G n$),
$$U = \boldsymbol{\omega} \times_1 U^{(1)} \times_2 ... \times_d U^{(d)} \in \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1 R_G, \mathbf{r} R_G}}.$$

In some cases one may require the consequent rank reduction of the target tensor U to the two-level format $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_0,\mathbf{r}_0}}$ with moderate rank parameters R_0 and $\mathbf{r}_0 = (r_0, ..., r_0)$. This can be accomplished by the following heuristic Algorithm 1′. In this way, we use several standard constructions. Let $\sigma_{\ell,1} \geq \sigma_{\ell,2} ... \geq \sigma_{\ell,\min(n,R)}$ be the singular values of the ℓ-mode side-matrices $U^{(\ell)} = Z^{(\ell)} D^{(\ell)} V^{(\ell)^T} \in \mathbb{R}^{n \times R}$ with $R = rR_G$, and let the rank-r_0 truncation of SVD for $U^{(\ell)}$ be given by $Z_0^{(\ell)} D_0^{(\ell)} V_0^{(\ell)^T}$, such that $D_0^{(\ell)} = \text{diag}\{\sigma_{\ell,1}, \sigma_{\ell,2}, ..., \sigma_{\ell,r_0}\}$ and $Z_0^{(\ell)}$, $V_0^{(\ell)}$ are given by the respective submatrices of $Z^{(\ell)}$ and $V^{(\ell)}$ containig their first r_0 columns ($\ell = 1, ..., d$). Furthermore, we rewrite the matrices $V^{(\ell)^T}$ and $Z^{(\ell)}$ in the block form,
$$V^{(\ell)^T} = [M_1^{(\ell)^T} M_2^{(\ell)^T} ... M_{R_G}^{(\ell)\,T}], \quad Z^{(\ell)} = [Z_0^{(\ell)} Z_1^{(\ell)}],$$
with $M_m^{(\ell)} \in \mathbb{R}^{r \times n}$, and introduce the additional matrix splittings
$$D^{(\ell)} = \text{blockdiag}\{D_0^{(\ell)}, D_1^{(\ell)}\}, \quad M_m^{(\ell)} = [M_{m,0}^{(\ell)} M_{m,1}^{(\ell)}],$$
where $M_{m,0}^{(\ell)} \in \mathbb{R}^{r \times r_0}$, $M_{m,1}^{(\ell)} \in \mathbb{R}^{r \times (n-r_0)}$ ($m = 1, ..., R_G$).

Recall that the higher-order SVD (HOSVD, cf. [5]) tensor approximation is defined by truncated SVD of the mode-ℓ unfolding matrices.

Algorithm 1′. (Rank reduction for Algorithm 1.)

1. Given tensor U defined by Algorithm 1, and the rank parameters $r_0, R_0 \in \mathbb{N}$ (suppose that $R_0 \ll R_1 R_G$, $r_0 < r$).

2. For $\ell = 1, ..., d$, compute the ℓ-mode r_0-dimensional dominating subspace for $U^{(\ell)}$, specified by the rank-r_0 truncated SVD, given by $Z_0^{(\ell)} D_0^{(\ell)} V_0^{(\ell)^T}$ (cost $O(dnrR_G min\{n, rR_G\})$).

3. Project the target tensor U onto orthogonal basis defined by columns of $Z_0^{(\ell)}$ by calculating the core tensor of size $\mathbf{r}_0 = (r_0, ..., r_0)$ in the product-canonical format (the so-called reduced HOSVD, or shortly RHOSVD, cf. [16]),
$$\widetilde{\boldsymbol{\beta}}_0 = \sum_{m=1}^{R_G} \gamma_m \left(\sum_{\nu=1}^{R_1} \beta_\nu \bigotimes_{\ell=1}^{d} D_0^{(\ell)} M_{m,0}^{(\ell)\,T} z_\ell^\nu \right) \in \boldsymbol{\mathcal{C}}_{R_1 R_G, \mathbf{r}_0},$$
and represent the RHOSVD approximation in the form
$$U_{(\mathbf{r}_0)} = \widetilde{\boldsymbol{\beta}}_0 \times_1 Z_0^{(1)} \times_2 ... \times_d Z_0^{(d)} \in \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_1 R_G, \mathbf{r}_0}}.$$
The related cost is $O(dR_1 R_G r r_0)$.

4. Recompress the core $\widetilde{\boldsymbol{\beta}}_0$ to the rank-R_0 canonical tensor $\boldsymbol{\beta}_0$ and constitute the result in the contracted product form

$$W_0 = \boldsymbol{\beta}_0 \times_1 Z_0^{(1)} \times_2 ... \times_d Z_0^{(d)} \in \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_0,\mathbf{r}_0}}.$$

5. (Optional.) Use tensor W_0 as the initial guess for few nonlinear (say ALS) iterations to approximate the target tensor U in the $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R_0,\mathbf{r}_0}}$ format.

Notice that the iterative Step 5 in Algorithm 1′ is not mandatory. In our applications the approximation W_0 usually provides sufficiently good accuracy (see Lemma 3.4). The justification of Algorithm 1′ is based on the effective error control of the RHOSVD for U (cf. [16, Theorem 2.5] for the case of canonical input tensor).

Lemma 3.4 *(Error estimate for RHOSVD). The RHOSVD approximation of the Tucker rank* $\mathbf{r}_0$, $U_{(\mathbf{r}_0)}$, *given by the projection of* U *onto the matrices of singular vectors* $Z_0^{(\ell)}$, *exhibits the error estimate*

$$\|U - U_{(\mathbf{r}_0)}\| \le \|\boldsymbol{\gamma}\|\|\boldsymbol{\beta}\| \sum_{\ell=1}^{d} (\sum_{k=r_0+1}^{\min(n,R)} \sigma_{\ell,k}^2)^{1/2}, \quad \textit{where } \|\boldsymbol{\gamma}\|^2 = \sum_{m=1}^{R_G} \gamma_m^2. \tag{3.8}$$

Proof. Let us assume that $R \ge n$. Using the contracted product representation of the two-level Tucker tensor U,

$$U = \boldsymbol{\omega} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_d U^{(d)}, \quad \text{with } \boldsymbol{\omega} = \text{blockdiag}\{\gamma_1\boldsymbol{\beta}, ..., \gamma_{R_G}\boldsymbol{\beta}\},$$

we obtain, by the construction, the following expansion for the RHOSVD approximation

$$U_{(\mathbf{r}_0)} = \boldsymbol{\omega} \times_1 \left[Z_0^{(1)} D_0^{(1)} V_0^{(1)^T}\right] \times_2 \left[Z_0^{(2)} D_0^{(2)} V_0^{(2)^T}\right] \cdots \times_d \left[Z_0^{(d)} D_0^{(d)} V_0^{(d)^T}\right].$$

Introducing the auxiliary quantities

$$\Delta^{(\ell)} = U^{(\ell)} - Z_0^{(\ell)} D_0^{(\ell)} V_0^{(\ell)^T}, \quad W^{(\ell)} = Z_0^{(\ell)} D_0^{(\ell)} V_0^{(\ell)^T},$$

and

$$B_\ell = \boldsymbol{\omega} \times_1 U^{(1)} \cdots \times_{\ell-1} U^{(\ell-1)} \times_\ell \Delta^{(\ell)} \times_{\ell+1} W^{(\ell+1)} \cdots \times_d W^{(d)},$$

and using the triangle inequality, we obtain

$$\|U - U_{(\mathbf{r}_0)}\| = \|\sum_{\ell=1}^{d} B_\ell\| \le \sum_{\ell=1}^{d} \|B_\ell\|.$$

In turn, the ℓ-th term B_ℓ can be represented by a sum

$$\sum_{m=1}^{R_G} \gamma_m \boldsymbol{\beta} \times_1 U_m^{(1)} \cdots \times_{\ell-1} U_m^{(\ell-1)} \times_\ell Z_1^{(\ell)} D_1^{(\ell)} M_{m,1}^{(\ell)\,T} \times_{\ell+1} Z_0^{(\ell+1)} D_0^{(\ell+1)} M_{m,0}^{(\ell+1)^T} \cdots \times_d Z_0^{(d)} D_0^{(d)} M_{m,0}^{(d)\,T},$$

then, taking into account that $\|U_m^{(\ell)}\| = 1$, and $\|Z_0^{(\ell)} D_0^{(\ell)} M_{m,0}^{(\ell)\,T}\| \le 1$ ($\ell = 1, ..., d$, $m = 1, ..., R_G$), we arrive at the estimate

$$\|B_\ell\| \le \|\boldsymbol{\beta}\| \sum_{m=1}^{R_G} |\gamma_m| \|Z_1^{(\ell)} D_1^{(\ell)} M_{m,1}^{(\ell)\,T}\|.$$

Hence, we finalise the error bound using the Cauchy-Schwarz inequality, and taking into account that $V^{(\ell)}$ has orthonormal columns,

$$\begin{aligned}
\|U - U_{(\mathbf{r}_0)}\| &\leq \|\sum_{\ell=1}^{d}\sum_{m=1}^{R_G}|\gamma_m|\|\boldsymbol{\beta}\|\|Z_1^{(\ell)}D_1^{(\ell)}{M_{m,1}^{(\ell)}}^T\| \\
&\leq \|\boldsymbol{\beta}\|\sum_{\ell=1}^{d}\left(\sum_{m=1}^{R_G}\gamma_m^2\right)^{1/2}\left(\sum_{m=1}^{R_G}\|Z_1^{(\ell)}D_1^{(\ell)}{M_{m,1}^{(\ell)}}^T\|^2\right)^{1/2} \\
&= \|\boldsymbol{\gamma}\|\|\boldsymbol{\beta}\|\sum_{\ell=1}^{d}\left(\sum_{k=r_0+1}^{n}\sigma_{\ell,k}^2\sum_{m=1}^{R_G}\|\left(M_{m,1}^{(\ell)}\right)_k\|^2\right)^{1/2} \\
&= \|\boldsymbol{\gamma}\|\|\boldsymbol{\beta}\|\sum_{\ell=1}^{d}\left(\sum_{k=r_0+1}^{n}\sigma_{\ell,k}^2\right)^{1/2}.
\end{aligned}$$

The case $R < n$ can be analysed along the same line. ■

If $F \in \boldsymbol{\mathcal{C}}_{R_F}$ with $\boldsymbol{\beta} = diag\{\beta_1, ..., \beta_{R_F}\}$, and $G \in \boldsymbol{\mathcal{C}}_{R_G}$ as above, then

$$F \star G = \sum_{k=1}^{R_F}\sum_{m=1}^{R_G}\beta_k\gamma_m\left(f_1^k \star g_1^m\right) \otimes ... \otimes \left(f_d^k \star g_d^m\right), \tag{3.9}$$

leading to the reduced cost that scales linearly in dimensionality parameter d and linear-logarithmically in n,

$$\mathcal{N}_{C\star C\to C} = O(dR_FR_Gn\log n).$$

Algorithm 2. (Multidimensional tensor product convolution of type $C \star C \to C$)

1. Given $F \in \boldsymbol{\mathcal{C}}_{R_F,\mathbf{n}}$, $G \in \boldsymbol{\mathcal{C}}_{R_G,\mathbf{n}}$.
2. For $\ell = 1, ..., d$, compute the set of 1D convolutions $f_\ell^k \star g_\ell^m$ ($k = 1, ..., R_F$, $m = 1, ..., R_G$) of size $2n - 1$, restrict the results onto the index set I_ℓ, and form the $n \times R_FR_G$ side-matrix $U^{(\ell)}$ (cost $dR_FR_Gn\log n$).
3. Compute the set of scaling factors $\beta_k\gamma_m$ as in (3.9).

We have proven the following complexity bounds.

Lemma 3.5 *Algorithm 1 scales log-linearly in n and linearly in d,*

$$\mathcal{N}_{T_{\mathcal{C}}\star C\to T_{\mathcal{C}}} = O(drR_Gn\log n + drR_F + drR_Gn).$$

Algorithm 2 provides the complexity bound $O(dR_FR_Gn\log n)$.

The resultant convolution product $F \star G$ in (3.9) may be approximated in either Tucker or canonical formats, depending on further MLA operations applied to this tensor. In the framework of approximate iterations with structured matrices and vectors, we can fix the $\boldsymbol{\mathcal{C}}_{R_0}$-format for the output tensors, hence, the rank-R_0 canonical approximation (with $R_0 < R_FR_G$) would be the proper choice to represent $F \star G$. The tensor truncation of the rank-R_FR_G intermediate result to rank-R_0 tensor can be acomplished by fast multigrid accelerated tensor approximation at the cost $O(dR_FR_GR_0n\log n)$ (cf. [16]), and then the result can be stored by $O(dR_0n)$ reals.

Based on our experience with Algorithms 1 and 2, applied in electronic structure calculations in 3D, we notice that Algorithms 2 is preferable in the case of moderate grid-size (say, $n \leq 10^4$), while Algorithms 1 is faster for large grids. For example, Algorithms 2 works perfectly in electronic structure calculations by the Hartree-Fock model for $d = 3$ [14, 16]. For example, the Hartree potential of simple molecules can be calculated on the $n \times n \times n$ grid up to $n \leq 1.6 \cdot 10^4$ in a few minutes providing the relative accuracy about 10^{-7} already with $n = 8192$. Further numerical illustrations will be given in §4.

3.3 Tensor-product convolution on generic non-uniform grids

In this section we give few remarks concerning the design of multi-dimensional FTCT on non-uniform grids. Again, our key principle is the low rank tensor approximation of the multidimensional convolution transform described above. We stress the following issues:

- As soon as the multi-dimensional convolution is represented in the tensor-product form as in (3.4) - (3.9), the computation is reduced to the fast 1D convolution transforms of ℓ-mode univariate components for $\ell = 1, ..., d$ on equidistant grids, leading to practically negligible cost $O(n \log n)$ in the large range of a grid-size n.

- The 1D convolution on the hierarchically structured refined grids can be effectively computed in almost linear cost as discussed in [7].

- In the case of general tensor-product grids with adaptive grid refinement one can apply the *imbedding strategy* (cf. [15]) to reduce the computation to 1D-FFT on a uniform grid. Specifically, assume that a 1D refined grid of size n is obtained by agglomeration of subintervals of the auxiliary fine grid of size N (usually $n \ll N$, say $n = O(\log N)$), so that there is a natural extension operator $P_{n \to N}$ from adaptive to fine uniform grid. Further, assuming that the convolving tensors F and G in the collocation scheme (2.4) are represented in some structured tensor formats as above, the summation over $\mathbf{i}$ and for all $\mathbf{m} \in \mathcal{M}$ can be reduced to the tensor-product convolution on the auxiliary uniform grid with the cost $O(dr^2 N \log N + \#\boldsymbol{\beta} \cdot \#\boldsymbol{\gamma})$. Taking the proper subvectors of size N from the corresponding ℓ-mode components given on the grid of size $2N - 1$ ($\ell = 1, ..., d$) and interpolating the results to the initial "small" grid, we obtain the approximate convolution on the adaptive grid and in the tensor-product form. Detailed discussion of this issue is beyond the scope of our paper.

Preliminary numerical examples illustrating the efficiency of the convolution product in the Tucker/canonical formats are given in [15]. These results indicate that 1D FFT on the auxiliary equi-distant fine grid has negligible cost compared with the summation in (3.4), (3.5), at least in the parameter domain $N \leq 10^4$. Hence, the imbedding strategy can be successfully applied in the case of moderate mesh-refinement. To reduce the FFT-cost $O(N \log N)$ on the auxiliary uniform grid to the linear-logarithmic complexity in n, we will describe a multi-dimensional FTCT on the two-level composite grids (for ease of presentation we discuss the case of piecewise constant basis functions).

3.4 $O(n \log n)$ convolution on 1D composite grid

Let us describe the fast convolution transform (FCT) on a two-level composite grid defined by the coarse level lattice with mesh-size $H = 2A/n_0$. We introduce the coarse space

$V_{n_0} = span\{\phi_{\mathbf{i}_0}\}$, $\mathbf{i}_0 \in \mathbb{R}^{n_0^{\otimes d}}$, of piecewise constant basic functions (it is only for the ease of exposition) supported by the domain Ω. Assume that p intervals Ω_i, $i = 1, ..., p$ with $p \ll n_0$ are further decomposed by fine uniform grid of size $h = H/n$ (see (2.2)). The union of subdomains will be called $\Omega^{(p)} = \cup_{i=1}^{p} \Omega_i \subset \Omega$. We define by V_n the corresponding fine space of piecewise constant basic functions supported by $\Omega^{(p)}$ and having zero mean value at each subinterval $\Omega_1, ..., \Omega_p$, then introduce the composite space $V = V_{n_0} + V_n$. Our goal is the fast evaluation of the convolution product

$$w = (x_0 + x_h) \star (y_0 + y_h) \quad \text{with } x_0, y_0 \in V_{n_0},\ x_h, y_h \in V_n$$

at the cost $O(n_0 \log n_0 + n \log n)$ assuming that the result is projected to the initial composite space V. The corresponding numerical scheme can be implemented in four steps as follows (for the ease of presentation we further simplify and set $p = 1$).

Algorithm 3. (FCT on two-level composite grid)

1. Given $x_0, y_0 \in V_{n_0}$, $x_h, y_h \in V_n$.

2. Compute $w_h = x_h \star y_h$ and project the result to the coarse and fine spaces. This includes one convolution product of size n whose result will be defined on the union of intervals $\Omega_1 \cup \Omega_2$. The coarse components supported by Ω_1 and Ω_2 will be calculated using the mean values of $w_{h|\Omega_1}$ and $w_{h|\Omega_2}$, respectively. Consequently, the fine projection onto V_n has zero mean value. All together, this amounts to $O(n \log n)$ operations.

3. Compute $w_{0h} = x_0 \star y_h + x_h \star y_0$ and project the result to the coarse and fine spaces. The computational scheme is clear from the representation

$$x_h \star y_0 = x_h \star \sum_{i=1}^{n_0} a_i \chi_i = \sum_{i=1}^{n_0} a_i \left(x_h \star \chi_i\right)_{|\Omega_{i-1} \cup \Omega_i \cup \Omega_{i+1}},$$

   where χ_i is the indicator function of the interval Ω_i. This includes one convolution product and three scalar products of size n, plus calculation of the coarse grid projection. The numerical cost is estimated by $O(n \log n + n_0)$.

4. Compute $w_0 = x_0 \star y_0$ and project the result to the coarse and fine spaces. The corresponding computational ansatz

$$x_0 \star y_0 = (\sum_{i=1}^{n_0} a_i \chi_i) \star (\sum_{j=1}^{n_0} b_j \chi_j)$$

   is evaluated at the coarse level by FFT of size n_0. To obtain the projection onto V_n we compute the weighted convolution $a_1 b_1 (\chi_1 \star \chi_1)$ for the vectors of size n supported by Ω_1. Hence, the total cost of Step 4 is estimated by $O(n_0 \log n_0 + n \log n)$.

5. Collect the contributions from Steps 1 - 4 in the coarse and fine spaces which amounts to $O(n_0 + n)$ operations.

This proves the following result.

Lemma 3.6 *The numerical complexity of Algorithm 3 is estimated by* $O(n_0 \log n_0 + n \log n)$.

Algorithm 3 applies to two-level composite grids. However, it can be easily extended in a recursive manner to the case of multilevel composite grids.

Below we present numerical results for FCT on the two-level composite grid with $n_0 = 2^{\ell_0}$, $n = 2^{\ell_n}$, where n_0 and n are the dimensions of the coarse and fine spaces, respectively. The full grid size is given by $n_f = n_0 \cdot n$, which might be very large in our numerical examples (say, $n_f = 2^{17}$ with $n_0 = 2^9$, $n = 2^8$). Algorithm 3 is implemented in MATLAB 7.3. The next table presents CPU times (in sec.) for FFT on the corresponding full grid and for FCT on the composite two-level grid. In this example the finest auxiliary 1D grid attaines the size 2^{17}, which is more than enough to resolve arising singularities. The corresponding FCT appears to be at about $4 \cdot 10^4$ times faster than 1D FFT. Numerics clearly demonstrate the advantage of FCT on large composite grids.

ℓ_0	4	5	6	7	8	9	9	9	9	9	9
ℓ_n	8	8	8	8	8	8	7	6	5	4	3
FFT	0.28	0.77	2.7	10.8	45.9	401.	45.7	10.4	2.7	0.75	0.27
FCT	0.05	0.05	0.04	0.04	0.05	0.08	0.06	0.06	0.05	0.05	0.06

Fig. 3.1 represents the error of the FCT (left) and coarse components

$$x_0 = 0.2\,\{(-1)^k\,k\}_{k=1}^{n_0}, \quad y_0 = \{k\}_{k=1}^{n_0},$$

of the input vectors $x = x_0 + x_h$ and $y = y_0 + y_h$ defined on the coarse grid with $n_0 = 8$, and with step size $H = 1$. Fine components of the input vectors are given by $x_h = \{sin(2\pi\, i \cdot h\cdot)\}_{i=1}^{n}$ and $y_h = \{(-1)^i\}_{i=1}^{n}$ with fine grid size $n = 2^8$.

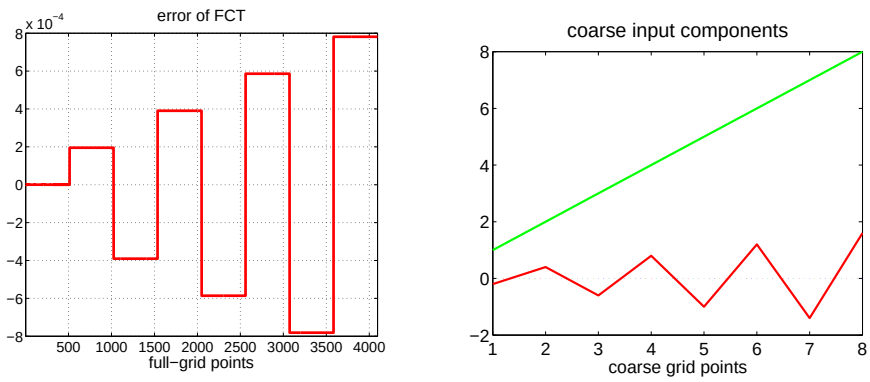

Figure 3.1: The error of the FCT (left) and coarse components x_0, y_0.

Fig. 3.2 represents the fine (left) and coarse (middle) components of the output vector, as well as the graph of the resultant convolution product.

In applications related to electronic structure calculations the number of refined zones may correspond to the number of atoms in the molecule requiring high resolution.

4 Computational aspects and numerical examples

4.1 Low-rank approximation of convolving tensors F and G

In applications related to electronic structure calculations, the function related collocation coefficient tensor $F = [f_{\mathbf{i}}]_{\mathbf{i}\in\mathcal{I}}$ can be generated by the electron density $\rho(x)$, by the product of

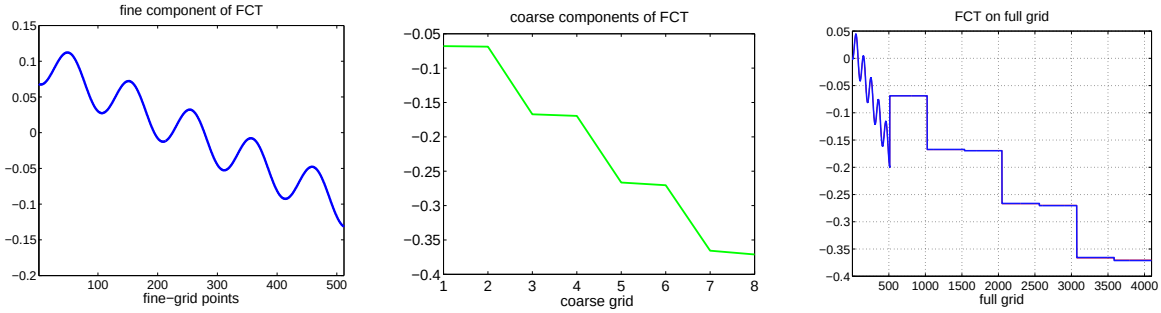

Figure 3.2: Fine and coarse components of the output vector (left, centre), resultant convolution vector (right).

the interaction potential $V(x)$ with the electron orbitals, $V(x)\psi(x)$, or by some related terms. In this way we make an *a priori* assumption on the existence of low rank approximation to the corresponding tensors. This assumption is not easy to analyse, however, it works well in practice.

Example 4.1 *In the case of hydrogen atom we have*

$$\rho(x) = e^{-2\|x\|}, \quad \textit{and} \quad V(x)\psi(x) = \frac{e^{-\|x\|}}{\|x\|} \quad \textit{with} \quad V(x) = \frac{1}{\|x\|}, \quad x \in \mathbb{R}^3,$$

hence, the existence of corresponding low-rank tensor approximations can be proven along the lines of [12, Lemma 4.3] and [13, Theorem 3].

To construct a low-rank approximation of the tensor G, we consider a class of multivariate spherically-symmetric convolving kernels $g : \mathbb{R}^d \to \mathbb{R}$ parametrised by

$$g = g(\rho(y)) \quad \text{with} \quad \rho \equiv \rho(y) = y_1^2 + ... + y_d^2,$$

where the univariate function $g : \mathbb{R}_+ \to \mathbb{R}$ can be represented via a generalised Laplace transform

$$g(\rho) = \int_{\mathbb{R}_+} \widehat{g}(\tau^2) e^{-\rho\tau^2} d\tau. \tag{4.1}$$

Without loss of generality, we introduce one and the same scaling function

$$\phi_i(\cdot) = \phi(\cdot + (i-1)h), \quad i \in I_n,$$

for all spatial dimensions $\ell = 1, ..., d$, where $h > 0$ is the mesh parameter, so that the corresponding tensor-product basis function $\phi_{\mathbf{i}}$ is defined in (2.3).

Using *sinc*-quadrature methods, we approximate the collocation coefficient tensor $G = [g_{\mathbf{i}}]_{\mathbf{i}\in\mathcal{I}}$ in (2.5) via rank-$(2M+1)$ canonical decomposition

$$g \approx \sum_{k=-M}^{M} w_k \, \mathcal{E}(\tau_k) \quad \text{with} \quad \mathcal{E} = [e_{\mathbf{i}}(\tau_k)], \quad \mathbf{i} \in \mathcal{I}, \tag{4.2}$$

with suitably chosen coefficients $w_k \in \mathbb{R}$ and quadrature points $\tau_k \in \mathbb{R}_+$, and with the rank-1 components $\mathcal{E}(\tau_k) \in \mathbb{R}^{\mathcal{I}}$ given by

$$e_{\mathbf{i}}(\tau_k) = \widehat{g}(\tau_k^2) \prod_{\ell=1}^{d} \int_{\mathbb{R}} e^{-y_\ell^2 \tau_k^2} \phi_{i_\ell}(y_\ell) dy_\ell. \tag{4.3}$$

For a class of analytic functions the exponentially fast convergence of above quadrature in M can be proven (see [9, 13]). Notice that the quadrature points τ_k can be chosen symmetrically, i.e., $\tau_k = \tau_{-k}$, hence reducing the number of terms in (4.2) to $r = M + 1$.

In our particular applications in electronic structure calculations we are interested in fast convolution with the Newton or Yukawa kernels. In the case of the Newton kernel, $g(x) = 1/\|x\|$, the approximation theory can be found in [9]. In the case of the Yukawa potential $e^{-\kappa\|x\|}/\|x\|$ for $\kappa \in [0, \infty)$, we apply the generalised Laplace transform (cf. (4.1))

$$g(\rho) = \frac{e^{-\kappa\sqrt{\rho}}}{\sqrt{\rho}} = \frac{2}{\sqrt{\pi}} \int_{\mathbb{R}_+} \exp(-\rho\tau^2 - \kappa^2/4\tau^2) d\tau, \tag{4.4}$$

corresponding to the choice

$$\widehat{g}(\tau^2) = \frac{2}{\sqrt{\pi}} e^{-\kappa^2/4\tau^2}.$$

Approximation theory in the case of Yukawa potential is presented in [13].

In our numerical experiments below the collocation coefficient tensor $G \in \mathbb{R}^{\mathcal{I}}$ for the Newton kernel is approximated in the rank-R canonical format with $R \in [20, 30]$ providing an accuracy about $10^{-7} \div 10^{-5}$ for the grid-size up to $n = 10^4$.

4.2 Algebraic recompression of the sinc approximation

In the case of large computational grids the tensor rank of the (problem independent) convolving kernel g can be reduced by an algebraic recompression procedure. For ease of presentation let us consider the case $d = 3$. The idea of our recompression algorithm is based on the observation that a typical feature of the analytic tensor approximation by the sinc quadratures as in (4.2)-(4.3) (for symmetric quadrature points it is agglomerated to the sequence with $k = 0, 1, ..., M$) is the presence of many terms all supported only by a few gridpoints from $p \times p \times p$ grid (domain $\Omega^{(p)}$) in the vicinity of the point-type singularity (say, at $x = 0$). Assume that this group of rank-1 tensors is numbered by $k = 0, ..., K < M$. The sum of these tensors, further called as A_p, effectively belongs to the low-dimensional space of trilinear $p \times p \times p$-tensors, hence the maximal tensor rank of A_p does not exceed $r = p^2 \leq K$. Furthermore, we can perform the rank-R_0 canonical approximation of this small tensor with $R_0 \ll K$ using the ALS or gradient type optimisation.

Algorithm 4. (Rank recompression for the canonical sinc-based approximation)

1. Given the canonical tensor A of rank $R = M + 1$.
2. Agglomerate all rank-1 terms supported by only one point, say by $\Omega^{(1)}$, into one rank-1 tensor, further called as A_1.
3. Agglomerate by a summation all terms supported by $\Omega^{(2)} \setminus \Omega^{(1)}$ in one tensor A_2 (with maximal rank 3), and approximate with the tensor rank $r_2 \leq 3$, and so on until we end up with tensor A_p supported by $\Omega^{(p)} \setminus \Omega^{(p-1)} \setminus ... \setminus \Omega^{(1)}$.
4. Approximate the canonical sum $A_1 + ... + A_p$ by low rank tensor.

Notice that in our sinc-quadrature approximations most of these "local" terms are supported by only one point, say by Ω_1, hence they are all agglomerated in rank-1 tensor. In approximation of the classical potentails like $1/\|\cdot\|$ or $e^{-\|\cdot\|}/\|\cdot\|$ the usual choice is $p = 1, 2$.

The simple rank recompression procedure described above allows to reduce noticeably the initial rank $R = M + 1$ appearing in the (symmetric) sinc quadratures. Numerical examples on the corresponding rank reduction by Algorithm 4 are depicted in [13],Figure 2. Figure

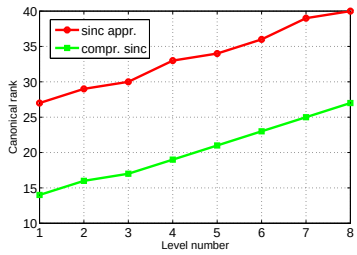

Figure 4.1: Tensor rank of the *sinc*- and recompressed *sinc*-approximation for $1/\|x\|$.

4.1 presents the rank parameters obtained from the sinc approximations of $g(x) = 1/\|x\|$ up to threshold $\varepsilon = 0.5 \cdot 10^{-6}$ in max-norm, computed on $n \times n \times n$ grids with $n = 2^{L+3}$ for the level number $L = 1, ..., 8$ (upper curve), and the corresponding values obtained by Algorithm 4 with $p = 1$ (lower curve). One observes the significant reduction of the tensor rank.

4.3 Numerical verification on quantum chemistry data, $g(x) = \frac{1}{\|x\|}$

We test the approximation error of the tensor-product collocation convolution scheme on practically interesting data arising in electronic structure calculations using the Hartee-Fock equation (see [14] for more details). We consider the (pseudo) electron density of simple CH_4-molecule represented by the exponential sum

$$f(x) := \sum_{\nu=1}^{M} \left(\sum_{k=1}^{R_0} c_{\nu,k} (x - x_k)^{\beta_k} e^{-\lambda_k (x - x_k)^2} \right)^2, \quad x \in \mathbb{R}^3, \quad R_0 = 50, \ M = 4 \tag{4.5}$$

with x_k corresponding to the locations of the C and H atoms. We extract the "principal exponential" approximation of the electron density, f_0, obtained by setting $\beta_k = 0$ ($k = 1, ..., R_0$) in (4.5). Using the fast tensor-product convolution method, the Hartree potential of f_0,

$$V_H(x) = \int_\Omega \frac{f_0(y)}{\|x - y\|} dy, \quad x \in \Omega = [-A, A]^3,$$

is computed with high accuracy on a sequence of uniform $(n+1) \times (n+1) \times (n+1)$ grids with $n = 2^p$, $p = 5, 5, ..., 12$, and with $A = 9.6$. The initial rank of the input tensor $F = [f_0(y_{\mathbf{i}})]_{\mathbf{i} \in \mathcal{I}}$, presented in the canonical format, is bounded by $R \leq \frac{R_0(R_0+1)}{2}$ (even for simple molecules it normally takes about several thousands). The collocation coefficients tensor G in (2.5) for the Newton kernel is approximated by the *sinc*-method with the algebraic rank-recompression described in Algorithm 4. Figure 4.2 represents the shape of the functions f_0 and V_H.

Note that the Hartree potential has slow polynomial decay, i.e.,

$$V_H(x) = O(\frac{1}{\|x\|}) \quad \text{as} \quad \|x\| \to \infty,$$

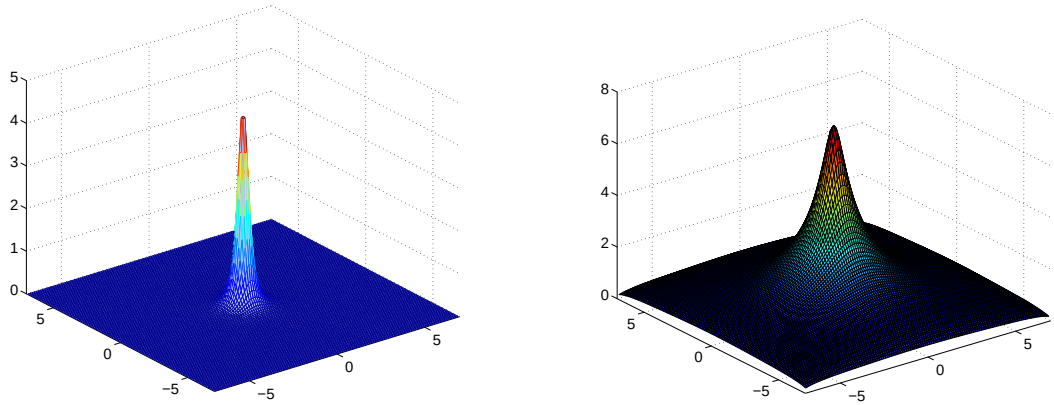

Figure 4.2: The density $f_0(x_1, x_2, 0)$ (left) and its Hartree potential $V_H(x_1, x_2, 0)$ (right).

however, the density f_0 decays exponentially. Hence the accurate tensor approximation is computed in some smaller box $\Omega' = [-B, B]^3 \subset \Omega$, $B < A$.

In this numerical example the resultant convolution product with the Newton convolving kernel can be calculated exactly by using the analytic representation for each individual Gaussian,

$$\left(e^{-\alpha\|\cdot\|^2} \star \frac{1}{\|\cdot\|}\right)(x) = \left(\frac{\alpha}{\pi}\right)^{-3/2} \frac{1}{\|x\|} \operatorname{erf}(\sqrt{\alpha}\|x\|),$$

where the erf-function is defined by

$$\operatorname{erf}(t) := \frac{2}{\sqrt{\pi}} \int_0^t exp(-\tau) d\tau, \quad t \geq 0.$$

The Hartree potential $V_H = f_0 \star 1/\|\cdot\|$ attains its maximum value at the origin $x = 0$ that is $V_H(0) = 7.19$. Figure 4.3(left) demonstrates the accuracy $O(h^2)$ of our tensor approximation and of the corresponding improved values, $O(h^3)$, due to the Richardson extrapolation. Here the grid-size is given by $n = n_\ell = 2^{\ell+4}$ for the level number $\ell = 1, ..., 7$, with the finest grid-size $n_7 = 2048$. It is seen that beginning from the level number $\ell = 5$ ($n_5 = 512$) the extrapolated scheme already achieves the saturation error 10^{-6} of the tensor approximation related to the chosen Tucker rank $r = 22$. This demonstrates high accuracy of the Richardson extrapolation.

Absolute error for the Hartree potential of CH_4 molecule are given in Figure 4.3(middle) compared with the commonly used MOLPRO calculations (here we have $\max |V_H| = 8.6$).

Figure 4.3(right) presents the CPU times (min) to compute the 3D FTCT on $n \times n \times n$ grid for a sequence of grid-sizes $n \in [64, 128, ..., 8192]$, and with the input rank $R_2 = 256, 484$. It confirms the theoretical result on linear-logarithmic scaling in n, and linear scaling in R_2.

Acknowledgements. The author is appreciative to Prof. Dr. W. Hackbusch for valuable discussions on the fast convolution algorithms on composite grids. The author is thankful to Dipl.-Phys. V. Khoromskaia (MPI MIS, Leipzig) for substantial help with numerical computations.

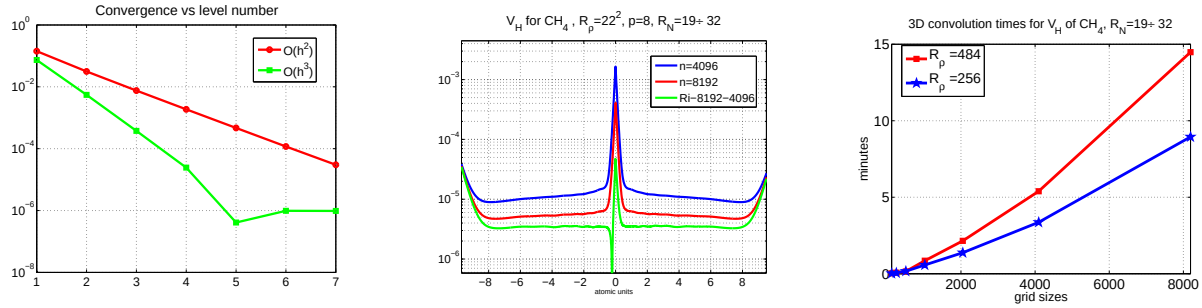

Figure 4.3: Convergence history for the $O(h^2)$- and $O(h^3)$-extrapolated convolution schemes (left); absolute error for the Hartree potential of CH_4 molecule (middle); CPU time vs. the grid size and the input rank R_2 (right).

References

[1] G. Beylkin and M.J. Mohlenkamp: *Numerical operator calculus in higher dimensions.* Proc. Natl. Acad. Sci. USA, **99** (2002), 10246-10251.

[2] G. Beylkin, M.J. Mohlenkamp and F. Pérez: *Approximating a wavefunction as an unconstrained sum of Slater determinants.* Journal of Math. Phys., **49** 032107 (2008).

[3] P. Comon: *Tensor Decomposition: State of the Art and Applications.* IMA Conf. Math. in Signal Proc., Warwick, UK, 2000 (December).

[4] 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.

[5] L. De Lathauwer, B. De Moor, J. Vandewalle, *A multilinear singular value decomposition.* SIAM J. Matrix Anal. Appl., 21 (2000) 1253-1278.

[6] H.-J. Flad, B.N. Khoromskij, D.V. Savostyanov, and E.E. Tyrtyshnikov: *Verification of the cross 3d algorithm on quantum chemistry data.* J. Numer. Math. 2008 (submitted).

[7] W. Hackbusch, *Fast and Exact Projected Convolution for Non-equidistant Grids.* Computing **80** (2007), 137-168.

[8] I.P. Gavrilyuk, W. Hackbusch and B.N. Khoromskij: *Hierarchical Tensor-Product Approximation of the Inverse and Related Operators in High-dimensional Elliptic Problems.* Computing **74** (2005), 131-157.

[9] 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.

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

[11] R.J. Harrison, G.I. Fann, T. Yanai, Z. Gan, and G. Beylkin: *Multiresolution quantum chemistry: Basic theory and initial applications.* J. of Chemical Physics, 121 (23): 11587-11598, 2004.

[12] 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.

[13] B.N. Khoromskij: *On Tensor Approximation of Green Iterations for Kohn-Sham Equations.* Computing and Visualization in Science, 11:259-271 (2008); DOI:10.1007/s00791-008-0097-x, 2008.

[14] B.N. Khoromskij, S.R. Chinnamsetty, H.-J- Flad, V. Khoromskaia: *Tensor decomposition in electronic structure calculations on Cartesian grids.* Preprint MPI MIS 65, Leipzig 2007. J. Comp. Phys. (accepted).

[15] B.N. Khoromskij and V. Khoromskaia: *Low Rank Tucker Tensor Approximation to the Classical Potentials.* Central European J. of Math., **5**(3) 2007, 1-28.

[16] B.N. Khoromskij and V. Khoromskaia: *Multigrid Tensor Approximation of Function Related Arrays.* Preprint MPI MIS 40, Leipzig 2008 (SIAM J. Sci. Comp., 2009, to appear).

[17] T. G. Kolda and B. W. Bader: *Tensor Decompositions and Applications.* Technical Report Number SAND2007-6702, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, November 2007 (SIAM Review, 2008, to appear).

[18] I.V. Oseledets, D.V. Savostyanov: and E.E. Tyrtyshnikov, *Tucker dimensionality reduction of three-dimensional arrays in linear time.* SIMAX, 30(3), 939-956 (2008).

[19] I.V. Oseledets, D.V. Savostyanov: and E.E. Tyrtyshnikov, *Linear Algebra for Tensor Problems.* Computing 2009, to appear.

[20] A. Smilde, R. Bro, and P. Geladi: *Multiway Analysis.* Wiley, 2004.

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

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