Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Tensor Structured Evaluation of Singular Volume Integrals

by

Jonas Ballani and Peter Meszmer

Preprint no.: 70 2012

Tensor Structured Evaluation of Singular Volume Integrals

Jonas Ballani, Peter Meszmer

December 3, 2012

In this article, we introduce a new method for the accurate and fast computation of singular integrals over cuboids in three-dimensional space. Using a straightforward geometric parametrisation of the domain of integration, we interpret the integral as a smooth function on a high-dimensional parameter space. A standard interpolation scheme then leads to a high-dimensional tensor to which an approximation in the data-sparse hierarchical tensor format is applied. Once this approximation is available, the evaluation of an integral value becomes an easy task which does no longer require the treatment of singular terms. Numerical experiments illustrate the potential of the proposed approach for typical examples.

1 Introduction

Singular volume integrals appear in a wide range of applications from e.g. physics and quantum chemistry. In particular, the discretisation of singular integral equations in three-dimensional space leads to the task of evaluating integrals of the form

$$\int_X \int_Y \varphi(x)\kappa(x,y)\psi(y)\,\mathrm{d}y\,\mathrm{d}x, \qquad X, Y \subset \mathbb{R}^3. \tag{1}$$

Here, $\kappa : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ denotes a kernel function that has a singularity at $x = y$ and is analytic elsewhere and the functions $\varphi : X \to \mathbb{R}$, $\psi : Y \to \mathbb{R}$ are assumed to be smooth. In a standard axis-aligned discretisation, the subsets $X, Y \subset \mathbb{R}^3$ are cuboids of the form

$$X = X_1 \times X_2 \times X_3, \quad Y = Y_1 \times Y_2 \times Y_3, \tag{2}$$

with bounded intervals $X_\mu, Y_\mu \subset \mathbb{R}$, $\mu = 1, 2, 3$. Classical examples for this setting are the evaluation of the Newton potential with $\kappa(x,y) = 1/(4\pi\,\|x-y\|)$ and problems from the context of electromagnetic scattering, micromagnetics, or radiative heat transfer.

The computation of integrals of the form (1) is particularly challenging when the two cuboids X and Y either intersect or have small positive distance. Due to the singularity of the kernel function κ, standard quadrature routines can only be applied at significant computational costs in order to obtain high accuracies which motivates the development of alternative techniques.

1.1 Evaluation by Interpolation

In this article, we propose a method which prepares suitable auxiliary data in a setup step which is required only once. A single integral value can then be obtained in a cheap way by a few (simple) operations. The key idea of our approach is the investigation of functions of the type

$$f(a) := \int_{X(a)} \int_{Y(a)} \varphi(x)\kappa(x,y)\psi(y)\,\mathrm{d}y\,\mathrm{d}x \tag{3}$$

where $a \in I := I_1 \times \ldots \times I_p$, $I_\mu \subset \mathbb{R}$, is a parameter tuple that describes the geometric shapes of the cuboids $X(a)$ and $Y(a)$. In practice, we can easily choose this parametrisation in such a way that f is smooth in the whole parameter space $I \subset \mathbb{R}^p$. We therefore expect to be able to approximate f in I by polynomials that can be evaluated very efficiently. Note that we have used this approach already in [1] to accelerate the computation of singular *surface* integrals appearing in the boundary element method.

In order to get rid of the specific choice of the functions φ, ψ, we assume that they can be represented (or approximated) in a fixed basis set. Due to the smoothness assumptions, we restrict ourselves to a polynomial basis of fixed maximal degree. We hence require that we can represent (or approximate) φ, ψ in the form

$$\varphi(x) = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{m}_X} \varphi(\xi_{\boldsymbol{i}}^X) L_{\boldsymbol{i}}^X(x), \qquad \psi(y) = \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{m}_Y} \psi(\xi_{\boldsymbol{j}}^Y) L_{\boldsymbol{j}}^Y(y),$$

with interpolation points $\xi_{\boldsymbol{i}}^X \in X$, $\xi_{\boldsymbol{j}}^Y \in Y$, and products of Lagrange polynomials $L_{\boldsymbol{i}}^X : X \to \mathbb{R}$, $L_{\boldsymbol{j}}^Y : Y \to \mathbb{R}$, of order m_X, m_Y, respectively, where for $X, Y \subset \mathbb{R}^3$ we have $\boldsymbol{m}_X := (m_X, m_X, m_X)$, $\boldsymbol{m}_Y := (m_Y, m_Y, m_Y)$. We are then left with the task of evaluating integrals of the form

$$f_{\boldsymbol{i},\boldsymbol{j}}(a) := \int_{X(a)} \int_{Y(a)} L_{\boldsymbol{i}}^X(x)\kappa(x,y)L_{\boldsymbol{j}}^Y(y)\,\mathrm{d}y\,\mathrm{d}x.$$

Note that for a fixed kernel function κ, the functions $f_{\boldsymbol{i},\boldsymbol{j}} : I \to \mathbb{R}$ do no longer depend on the specific choice of φ and ψ, but only on the geometric shapes of the cuboids X and Y.

We now interpolate the function $f_{\boldsymbol{i},\boldsymbol{j}}$ on the parameter space $I \subset \mathbb{R}^p$ by a multivariate interpolation scheme of order m_I. The values of $f_{\boldsymbol{i},\boldsymbol{j}}$ in the interpolation points $(\xi_{1,\ell_1}, \ldots, \xi_{p,\ell_p}) \in I$, define a $(p+6)$-dimensional tensor

$$A \in \mathbb{R}^{(m_X+1)^3 \times (m_Y+1)^3 \times (m_I+1)^p}$$

given by

$$A_{(\boldsymbol{i},\boldsymbol{j},\boldsymbol{\ell})} := f_{\boldsymbol{i},\boldsymbol{j}}(\xi_{1,\ell_1},\dots,\xi_{p,\ell_p}), \qquad \ell_\mu = 0,\dots,m,\ \mu = 1,\dots,p, \tag{4}$$

with $\boldsymbol{\ell} = (\ell_1,\dots,\ell_p)$.

The interpolation on its own does not lead to an efficient scheme since the tensor A contains $(m_X+1)^3(m_Y+1)^3(m_I+1)^p$ entries which even may exceed the available memory for moderate values of m_X, m_Y, m_I, and p. We therefore look for a tensor $\tilde{A}$ in a data-sparse representation that approximates A and which can be evaluated efficiently.

1.2 Tensor Approximation

Several formats have been proposed to represent a tensor $A \in \mathbb{R}^{n_1\times\cdots\times n_d}$ in a data-sparse way:

1. The Tucker format which represents a tensor in a storage complexity of $\mathcal{O}(k^d+dkn)$.

2. The canonical format (CP, r-term representation) which represents a tensor in a storage complexity of $\mathcal{O}(drn)$.

3. The hierarchical format by Hackbusch and Kühn [9] which represents a tensor in a storage complexity of $\mathcal{O}(dk^3+dkn)$. This format also contains the TT format introduced by Oseledets and Tyrtyshnikov [12] as a special case.

Here, r and k denote ranks in the specific format which depend on the tensor which has to be represented and $n := \max_{1\le\mu\le d} n_\mu$. For a detailed introduction to tensor representations, we refer the reader to [8].

In this article, we focus on the hierarchical tensor format since it allows for a structured and data-sparse representation of tensors even in high dimensions d. Moreover, reliable truncation procedures are available [5] that enable us to determine the necessary representation ranks adaptively to reach a given target accuracy. In addition, in [2] we have developed an efficient black box strategy to find an approximation of a tensor in the hierarchical format from a small subset of its entries. This is particularly important if the quantity n^d is very large such that a full representation of A is not available. Once an approximation of the tensor A from (4) in the hierarchical format has been obtained, the evaluation of a single integral becomes a simple task which can be carried by a few standard arithmetic operations.

1.3 Alternative Evaluation Techniques for Singular Integrals

Various techniques have been developed in the literature to cope with singular volume integrals. For the Newton potential, the integration can be carried out analytically where a proper stabilisation of the resulting antiderivatives is required [6]. A large class of integrals can be computed by the approximation by exponential sums [7] or by optimised sinc-quadruature [3]. Both methods exploit the tensor structure of approximate kernel representations such that integral values even in high spatial dimensions can be obtained at moderate costs.

In [10], we have introduced a quite general hierarchical quadrature scheme for the computation of singular integrals that only relies on standard quadrature routines for regular integrals. This method even allows the computation of hyper-singular integrals and is hence applicable to a very general class of singular integrals. As a drawback, the number of required regular integrals that need to be computed is relatively large. Fortunately, in our new approach which is based on interpolation in a geometric parameter space, the initial quadrature routine is only required once in the setup step and not needed for the actual evaluation of singular integrals. We can therefore afford to use the method developed in [10] in the setup step which enables us to treat a large class of singular integrals by a uniform approach. We will review the concept of hierarchical quadrature in Section 5.

2 Tensor Structured Tools in High Dimensions

In this section, we recall some basic concepts from the field of high-dimensional approximation. We start with the definition of a standard multivariate interpolation scheme which naturally leads to the problem of the representation and approximation of data in high dimensions. We continue with an introduction to the hierarchical tensor format from [9]. Furthermore, we summarise some approximation results for this format from the literature.

2.1 Multivariate Interpolation

Let $d \in \mathbb{N}$ and $f : I \to \mathbb{R}$ be smooth in a subset $I \subset \mathbb{R}^d$ which is given as the cartesian product of intervals,

$$I := I_1 \times \ldots \times I_d, \qquad I_\mu = [\alpha_\mu, \beta_\mu] \subset \mathbb{R}, \quad \mu = 1, \ldots, d. \tag{5}$$

In order to construct an interpolant of f on I, we first define the interpolation on an interval $[\alpha, \beta] \subset \mathbb{R}$. On the reference interval $[-1, 1]$, we introduce the m-th order Chebyshev points by

$$\hat{\xi}_i := \cos\left(\frac{2i+1}{2m+2}\pi\right), \qquad i = 0, \ldots, m.$$

The reference points $\hat{\xi}_i$ may be transformed to the given interval $[\alpha, \beta]$ by the elementary relation

$$\xi_i^{[\alpha,\beta]} := \frac{\alpha+\beta}{2} + \frac{\beta-\alpha}{2}\hat{\xi}_i, \qquad i = 0, \ldots, m.$$

The corresponding Lagrange polynomials have the form

$$L_i^{[\alpha,\beta]}(x) = \prod_{j=0, j\neq i}^{m} \frac{x - \xi_j^{[\alpha,\beta]}}{\xi_i^{[\alpha,\beta]} - \xi_j^{[\alpha,\beta]}}, \qquad i = 0, \ldots, m.$$

Now, we can define a one-dimensional interpolation operator by

$$\mathfrak{I}_m^{[\alpha,\beta]} : C[\alpha,\beta] \to \mathbb{P}_m, \qquad g \mapsto \sum_{i=0}^{m} g\left(\xi_i^{[\alpha,\beta]}\right) L_i^{[\alpha,\beta]}, \tag{6}$$

where $\mathbb{P}_m$ is the space of polynomials of degree at most m.

We can apply the one-dimensional construction on all intervals I_μ with interpolation points $\xi_\mu^{I_\mu} \in I_\mu$ and corresponding Lagrange polynomials $L_i^{I_\mu}$, $i = 0, \ldots, m$, to define interpolation operators $\mathfrak{I}_m^{I_\mu}$ as in (6) for all $\mu = 1, \ldots, d$. Due to the product structure of I, the one-dimensional interpolation operators can be combined to a d-dimensional interpolation operator by forming their tensor product

$$\mathfrak{I}_m^{I} := \mathfrak{I}_m^{I_1} \otimes \ldots \otimes \mathfrak{I}_m^{I_d}. \tag{7}$$

The interpolation function then explicitly reads

$$\mathfrak{I}_m^{I}[f](x) = \sum_{i_1=0}^{m} \cdots \sum_{i_d=0}^{m} f(\xi_{i_1}^{I_1}, \ldots, \xi_{i_d}^{I_d}) L_{i_1}^{I_1}(x_1) \ldots L_{i_d}^{I_d}(x_d) \tag{8}$$

with $x = (x_1, \ldots, x_d)$.

2.2 Hierarchical Tensor Representation

Given $d \in \mathbb{N}$, let $\mathcal{I}_1, \ldots, \mathcal{I}_d$ be finite index sets with $\mathcal{I} := \mathcal{I}_1 \times \cdots \times \mathcal{I}_d$. Already for moderate d, the data-sparse representation of tensors $A \in \mathbb{R}^{\mathcal{I}}$ is of interest. A quite general concept for the representation of tensors has been introduced by Hackbusch and Kühn [9] and was further analysed by Grasedyck [5]. In this approach, data-sparsity is obtained by a hierarchical representation of tensors which can be applied even in high dimensions d. As a first important ingredient, we define a matrix representation of a given tensor.

Definition 1 (matricisation)**.** Let $D := \{1, \ldots, d\}$. Given a subset $t \subset D$ with complement $s := D \setminus t$, the *matricisation*

$$\mathcal{M}_t : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{I}_t} \otimes \mathbb{R}^{\mathcal{I}_s}, \qquad \mathcal{I}_t := \bigtimes_{\mu \in t} \mathcal{I}_\mu,\ \mathcal{I}_s := \bigtimes_{\mu \in s} \mathcal{I}_\mu,$$

of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ is defined by its entries

$$\mathcal{M}_t(A)_{(i_\mu)_{\mu\in t},(i_\mu)_{\mu\in s}} := A_{(i_1,\ldots,i_d)}, \qquad i_\mu \in \mathcal{I}_\mu,\ \mu \in D.$$

In order to allow for a hierarchical representation, the subsets $t \subset D$ are organised in a binary tree.

Definition 2 (dimension tree)**.** Let $D := \{1, \ldots, d\}$. A tree T_D is called a *dimension tree* if the following three conditions hold:

(a) the index set D is the root of the tree T_D,

(b) all vertices $t \in T_D$ are non-empty subsets $t \subset D$,

(c) every vertex $t \in T_D$ with $\#t \geq 2$ has two sons $t_1, t_2 \in T_D$ with the property

$$t = t_1 \cup t_2, \qquad t_1 \cap t_2 = \emptyset.$$

The set of leaves of T_D is defined by $\mathcal{L}(T_D) := \{t \in T_D : \#t = 1\}$. For all $t \in T_D \setminus \mathcal{L}(T_D)$, we denote the set of sons of t by $\operatorname{sons}(t)$.

Based on the concept of the matricisation of tensors and the definition of a dimension tree, we can now introduce the hierarchical tensor format.

Definition 3 (hierarchical rank, hierarchical format)**.** Let T_D be a dimension tree. The *hierarchical rank* $\boldsymbol{k} := (k_t)_{t \in T_D}$ of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ is defined by

$$k_t := \operatorname{rank}(\mathcal{M}_t(A)), \quad t \in T_D.$$

For a given hierarchical rank $\boldsymbol{k} := (k_t)_{t \in T_D}$, the *hierarchical format* $\mathcal{H}_{\boldsymbol{k}}$ is defined by

$$\mathcal{H}_{\boldsymbol{k}} := \{A \in \mathbb{R}^{\mathcal{I}} : \operatorname{rank}(\mathcal{M}_t(A)) \leq k_t,\, t \in T_D\}.$$

A key feature of a tensor in $\mathcal{H}_{\boldsymbol{k}}$ is that it can be represented in a recursive fashion.

Lemma 4 (cf. [5])**.** *Let $A \in \mathcal{H}_{\boldsymbol{k}}$. Then $A = (U_D)_{\cdot,1}$ can be represented by the recursive relation*

$$(U_t)_{\cdot,j} = \sum_{j_1=1}^{k_{t_1}} \sum_{j_2=1}^{k_{t_2}} (B_t)_{j,j_1,j_2} (U_{t_1})_{\cdot,j_1} \otimes (U_{t_2})_{\cdot,j_2}, \quad j = 1, \ldots, k_t,$$

for all $t \in T_D \setminus \mathcal{L}(T_D)$ with $\operatorname{sons}(t) = \{t_1, t_2\}$ *where $B_t \in \mathbb{R}^{k_t \times k_{t_1} \times k_{t_2}}$ and $U_t \in \mathbb{R}^{\mathcal{I}_t \times k_t}$, $\mathcal{I}_t := \times_{\mu \in t} \mathcal{I}_\mu$, for all $t \in T_D$.*

As a consequence of the last lemma, one only needs to store the matrices $U_t \in \mathbb{R}^{\mathcal{I}_\mu \times k_t}$ in the leaves $t = \{\mu\} \in \mathcal{L}(T_D)$ and the transfer tensors $B_t \in \mathbb{R}^{k_t \times k_{t_1} \times k_{t_2}}$ for all inner nodes $t = \{t_1, t_2\} \in T_D \setminus \mathcal{L}(T_D)$ in order to represent a tensor in $\mathcal{H}_{\boldsymbol{k}}$. The complexity for this representation then sums up to $\mathcal{O}((d-1)k^3 + dkn)$, where $k := \max_{t \in T_D} k_t$, $n := \max_{\mu \in D} \#\mathcal{I}_\mu$.

2.3 Tensor Approximation

Let $A \in \mathbb{R}^{\mathcal{I}}$ be defined by the values of its entries $A_{(i_1,\ldots,i_d)}$. Given $\varepsilon > 0$, we look for a tensor $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$ with an appropriately chosen hierarchical rank $\boldsymbol{k}$ such that $\|A - \tilde{A}\|_2 \leq \varepsilon$. For moderate sizes n^d, $n := \max_{\mu \in D} \#\mathcal{I}_\mu$, it is still possible to compute and store all entries of A. This full representation can then be used to find an approximation in $\mathcal{H}_{\boldsymbol{k}}$. For large sizes n^d, we can no longer compute and store all entries of A. We therefore have to find an approximation of A in $\mathcal{H}_{\boldsymbol{k}}$ from a subset of its entries in a black box fashion.

Approximation from Full Representation to $\mathcal{H}_{\boldsymbol{k}}$

The approximation of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ from its full representation to a tensor $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$ has been investigated in [9, 5]. Let T_D be a dimension tree and assume that for $\varepsilon > 0$ there exists a best approximation $A_{\text{best}} \in \mathcal{H}_{\boldsymbol{k}}$ of hierarchical rank $\boldsymbol{k} = (k_t)_{t \in T_D}$ with the property

$$\|A - A_{\text{best}}\|_2 \leq \varepsilon.$$

Then Remark 3.12 from [5] shows that by means of a hierarchical singular value decomposition it is possible to construct an approximation $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$ of the best approximation with hierarchical rank $\boldsymbol{k}$ and the property

$$\|A - \tilde{A}\|_2 \leq \sqrt{2d-3}\|A - A_{\text{best}}\|_2 \leq \sqrt{2d-3}\varepsilon.$$

The complexity for the hierarchical singular value decomposition is bounded by

$$\mathcal{O}\left(dn^{d+1} + dk^2n^d\right),$$

where $k := \max_{t \in T_D} k_t$, $n := \max_{\mu \in D} \#\mathcal{I}_\mu$. For given $\varepsilon > 0$, the necessary hierarchical rank $\boldsymbol{k}$ can be determined adaptively by standard linear algebra tools like the SVD.

Black Box Approximation to $\mathcal{H}_{\boldsymbol{k}}$

In [2] we have introduced a heuristic algorithm for the approximation of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ in $\mathcal{H}_{\boldsymbol{k}}$ which is given by the evaluation of a function

$$A_{(i_1,\ldots,i_d)} := f(\xi_{1,i_1}, \ldots, \xi_{d,i_d}), \qquad i_\mu \in \mathcal{I}_\mu,\ \mu = 1, \ldots, d.$$

The algorithm is based on the construction of low rank approximations of the matricisations of A by inspecting only very few tensor entries. Given a balanced dimension tree T_D of depth $\log(d)$, the number of required function evaluations is bounded by

$$\mathcal{O}\left(dk^3 + d\log(d)k^2n\right),$$

where $k := \max_{t \in T_D} k_t$, $n := \max_{\mu \in D} \#\mathcal{I}_\mu$. The algorithm is rank-adaptive in the sense that it finds the necessary hierarchical rank $\boldsymbol{k} = (k_t)_{t \in T_D}$ for the representation of $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$ with $\tilde{A} \approx A$ automatically to guarantee a (heuristic) accuracy of

$$\|A - \tilde{A}\|_\infty \leq \varepsilon.$$

The overall complexity for the setup of $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$ is bounded by

$$\mathcal{O}\left(dk^4 + d\log(d)k^2n\right).$$

Despite its heuristic nature, the proposed black box strategy has produced reliable and accurate results in a number of numerical examples, cf. [2, 1].

3 Approximate Evaluation of Integrals

In the following, we introduce an efficient scheme for the evaluation of

$$f(a) := \int_{X(a)} \int_{Y(a)} \varphi(x)\kappa(x,y)\psi(y)\,\mathrm{d}y\,\mathrm{d}x \tag{9}$$

which is based on an appropriately chosen parametrisation of the cuboids $X(a), Y(a)$ in terms of a parameter vector $a \in I \subset \mathbb{R}^p$. In a first step, the functions φ, ψ are interpolated within the cuboids X, Y. Secondly, we apply a multivariate interpolation scheme in the parameter space $I \subset \mathbb{R}^p$. In a third step, we approximate the resulting interpolation function by a tensor-structured expression which allows for an efficient evaluation.

3.1 Interpolation

Let $X, Y \subset \mathbb{R}^3$ be cuboids of the form (2) and let φ, ψ be polynomials of fixed maximal degree on X, Y, respectively. We can express φ, ψ in the form

$$\begin{aligned}\varphi(x) &= \sum_{i_1=0}^{m_X} \sum_{i_2=0}^{m_X} \sum_{i_3=0}^{m_X} \varphi\left(\xi_{i_1}^{X_1}, \xi_{i_2}^{X_2}, \xi_{i_3}^{X_3}\right) \prod_{\mu=1}^{3} L_{i_\mu}^{X_\mu}(x_\mu),\\ \psi(y) &= \sum_{j_1=0}^{m_Y} \sum_{j_2=0}^{m_Y} \sum_{j_3=0}^{m_Y} \psi\left(\xi_{j_1}^{Y_1}, \xi_{j_2}^{Y_2}, \xi_{j_3}^{Y_3}\right) \prod_{\mu=1}^{3} L_{j_\mu}^{Y_\mu}(y_\mu)\end{aligned}$$

with interpolation points $\xi_{i_\mu}^{X_\mu} \in X_\mu$, $\xi_{j_\mu}^{Y_\mu} \in Y_\mu$ and Lagrange polynomials $L_{i_\mu}^{X_\mu} : X_\mu \to \mathbb{R}$, $L_{j_\mu}^{Y_\mu} : Y_\mu \to \mathbb{R}$ of order m_X, m_Y, respectively. We may then write (9) in the form

$$f(a) = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{m}_X} \sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{m}_Y} v_{\boldsymbol{i}} w_{\boldsymbol{j}} f_{\boldsymbol{i},\boldsymbol{j}}(a) \tag{10}$$

with

$$v_{\boldsymbol{i}} := \varphi\left(\xi_{i_1}^{X_1}, \xi_{i_2}^{X_2}, \xi_{i_3}^{X_3}\right), \quad w_{\boldsymbol{j}} := \psi\left(\xi_{j_1}^{Y_1}, \xi_{j_2}^{Y_2}, \xi_{j_3}^{Y_3}\right) \tag{11}$$

and

$$f_{\boldsymbol{i},\boldsymbol{j}}(a) := \int_{X(a)} \int_{Y(a)} L_{\boldsymbol{i}}^{X}(x)\kappa(x,y)L_{\boldsymbol{j}}^{Y}(y)\,\mathrm{d}x\,\mathrm{d}y, \tag{12}$$

where $\boldsymbol{m}_X := (m_X, m_X, m_X)$, $\boldsymbol{m}_Y := (m_Y, m_Y, m_Y)$, and

$$L_{\boldsymbol{i}}^{X}(x) := \prod_{\mu=1}^{3} L_{i_\mu}^{X_\mu}(x_\mu), \quad L_{\boldsymbol{j}}^{Y}(y) := \prod_{\mu=1}^{3} L_{j_\mu}^{Y_\mu}(y_\mu).$$

Note that for a fixed kernel function κ, the functions $f_{\boldsymbol{i},\boldsymbol{j}}$ from (12) do no longer depend on the specific choice of φ, ψ but only on the geometric shapes of the cuboids X, Y. An interpolation of $f_{\boldsymbol{i},\boldsymbol{j}}$ in the geometric parameter space I now reads

$$f_{\boldsymbol{i},\boldsymbol{j}}(a) \approx \mathfrak{I}^I_{m_I}[f_{\boldsymbol{i},\boldsymbol{j}}](a) = \sum_{\ell_1=0}^{m_I} \cdots \sum_{\ell_p=0}^{m_I} f_{\boldsymbol{i},\boldsymbol{j}}(\xi^{I_1}_{\ell_1}, \ldots, \xi^{I_p}_{\ell_p}) \prod_{\mu=1}^{p} L^{I_\mu}_{\ell_\mu}(a_\mu). \tag{13}$$

The values of $f_{\boldsymbol{i},\boldsymbol{j}}$ at the interpolation points define a d-dimensional tensor $A \in \mathbb{R}^{\mathcal{I}}$, $d := p + 6$, with index set $\mathcal{I} := \mathcal{I}_1 \times \ldots \times \mathcal{I}_d$ where

$$\begin{aligned} \mathcal{I}_1 &= \mathcal{I}_2 = \mathcal{I}_3 = \{0, \ldots, m_X\}, \\ \mathcal{I}_4 &= \mathcal{I}_5 = \mathcal{I}_6 = \{0, \ldots, m_Y\}, \\ \mathcal{I}_7 &= \ldots = \mathcal{I}_d = \{0, \ldots, m_I\}. \end{aligned} \tag{14}$$

The entries of A are given by

$$A_{(\boldsymbol{i},\boldsymbol{j},\boldsymbol{\ell})} := f_{\boldsymbol{i},\boldsymbol{j}}(\xi^{I_1}_{\ell_1}, \ldots, \xi^{I_p}_{\ell_p}), \qquad \boldsymbol{\ell} = (\ell_1, \ldots, \ell_p). \tag{15}$$

It is clear that an efficient scheme can only be obtained if a structured and data-sparse representation of the tensor A is available.

3.2 Tensor Structured Evaluation

The representation of the function f from (10) involves three-dimensional coupling terms of the form (11). In order to fully profit from a tensor-structured approach, we need to make an additional assumption on the functions φ, ψ.

Assumption 5. The functions $\varphi : X \to \mathbb{R}$, $\psi : Y \to \mathbb{R}$, have separable representations of the form

$$\varphi(x) = \sum_{s=1}^{r_\varphi} \prod_{\mu=1}^{3} \varphi_{\mu,s}(x_\mu), \quad \psi(y) = \sum_{s=1}^{r_\psi} \prod_{\mu=1}^{3} \psi_{\mu,s}(y_\mu).$$

Such a representation becomes particularly attractive whenever $r_\varphi \ll (m_X + 1)^3$, or $r_\psi \ll (m_Y + 1)^3$. Under Assumption 5, the coefficients from (11) can be written in the form

$$\begin{aligned} v_{(i_1,i_2,i_3)} &= \sum_{s=1}^{r_\varphi} \prod_{\mu=1}^{3} (v_\mu)_{i_\mu,s}, \quad (v_\mu)_{i_\mu,s} := \varphi_{\mu,s}(\xi^{X_\mu}_{i_\mu}), \\ w_{(j_1,j_2,j_3)} &= \sum_{s=1}^{r_\psi} \prod_{\mu=1}^{3} (w_\mu)_{j_\mu,s}, \quad (w_\mu)_{j_\mu,s} := \psi_{\mu,s}(\xi^{Y_\mu}_{j_\mu}). \end{aligned}$$

We can then express (10) as

$$f(a) = \sum_{s_1=1}^{r_\varphi} \sum_{s_2=1}^{r_\psi} \hat{f}_{(s_1,s_2)}(a) \tag{16}$$

with

$$\hat{f}_{(s_1,s_2)}(a) := \sum_{i_1=0}^{m_X}\sum_{i_2=0}^{m_X}\sum_{i_3=0}^{m_X}\sum_{j_1=0}^{m_Y}\sum_{j_2=0}^{m_Y}\sum_{j_3=0}^{m_Y} f_{\boldsymbol{i},\boldsymbol{j}}(a)\prod_{\mu=1}^{3}((v_\mu)_{i_\mu,s_1}(w_\mu)_{j_\mu,s_2}). \qquad (17)$$

The main task is now to evaluate a single term $\hat{f} = \hat{f}_{(s_1,s_2)}$ from (16) at a fixed point $a \in I$ in a structured way.

Let $D_X := \{1,2,3\}$, $D_Y := \{4,5,6\}$, $D_I := \{7,\ldots,p+6\}$, and $D := D_X \cup D_Y \cup D_I$. Moreover, let T_D be a dimension tree according to Definition 2. We assume that the tensor $A \in \mathbb{R}^{\mathcal{I}}$ from (15) can be approximated by a tensor $\tilde{A} \in \mathbb{R}^{\mathcal{I}}$ represented in hierarchical format with hierarchical rank $\boldsymbol{k} := (k_t)_{t\in T_D}$, i.e. $\tilde{A} \in \mathcal{H}_{\boldsymbol{k}}$. Then $\tilde{A}$ can be represented recursively by

$$(U_t)_{\cdot,j} = \sum_{j_1=1}^{k_{t_1}}\sum_{j_2=1}^{k_{t_2}}(B_t)_{j,j_1,j_2}(U_{t_1})_{\cdot,j_1}\otimes(U_{t_2})_{\cdot,j_2}, \quad j=1,\ldots,k_t,$$

for all $t \in T_D \setminus \mathcal{L}(T_D)$ with $\mathrm{sons}(t) = \{t_1,t_2\}$ and final representation $\tilde{A} = (U_D)_{\cdot,1}$.

First note that for each $\mu \in D_X \cup D_Y$, there appears a single sum in expression (17). Equivalently, for each $\mu \in D_I$, there appears a single sum in expression (13). Due to the multilinearity of the tensor product, each of these sums can be moved to a leaf $t \in \mathcal{L}(T_D)$ for which $t = \{\mu\}$. This observation immediately leads to an efficient recursive evaluation scheme.

Evaluation in the Leaves

Let $t = \{\mu\} \in \mathcal{L}(T_D)$. We then define $u_t \in \mathbb{R}^{k_t}$ by

$$(u_t)_j := \sum_{i\in\mathcal{I}_\mu} c_{\mu,i}(U_t)_{i,j}, \quad j=1,\ldots,k_t, \qquad (18)$$

where

$$c_{\mu,i} := \begin{cases}(v_\mu)_{i,s_1}, & \mu \in D_X,\\ (w_{\mu-3})_{i,s_2}, & \mu\in D_Y,\\ L_i^{I_{\mu-6}}(a_{\mu-6}), & \mu\in D_I.\end{cases} \qquad (19)$$

For $\mu \in D_X \cup D_Y$, the evaluation of the vector u_t requires $\mathcal{O}(k_t \cdot \#\mathcal{I}_\mu)$ operations. For $\mu \in D_I$, the same complexity can be obtained by using Horner's scheme applied to Newton's divided differences (which can be precomputed).

Evaluation in the Inner Nodes

Let $t \in T_D \setminus \mathcal{L}(T_D)$ with $\mathrm{sons}(t) = \{t_1,t_2\}$. Assume that vectors $u_{t_1} \in \mathbb{R}^{k_{t_1}}$, $u_{t_2} \in \mathbb{R}^{k_{t_2}}$ have been computed for the sons of t. Then a new vector $u_t \in \mathbb{R}^{k_t}$ is defined by

$$(u_t)_j := \sum_{j_1=1}^{k_{t_1}}\sum_{j_2=1}^{k_{t_2}}(B_t)_{j,j_1,j_2}(u_{t_1})_{j_1}(u_{t_2})_{j_2}, \quad j=1,\ldots,k_t.$$

The evaluation of the vector u_t requires $\mathcal{O}(k_t k_{t_1} k_{t_2})$ operations.

Final Representation

The final value is obtained by the vector $u_D \in \mathbb{R}^1$ in the root of T_D, i.e.

$$\hat{f}(a) \approx (u_D)_1.$$

The complexity for the evaluation at *all* inner nodes sums up to $\mathcal{O}((d-1)k^3)$, $k := \max_{t \in T_D} k_t$. Adding the computational costs for the evaluation at the leaf nodes, we end up with an overall complexity of $\mathcal{O}((d-1)k^3 + dkn)$, $n := \max\{\#\mathcal{I}_\mu : \mu \in D\}$.

A Refined Strategy

So far, the evaluation of a single term from (16) has been discussed. A straightforward summation over all involved terms therefore leads to a computational cost of $\mathcal{O}(r_\varphi r_\psi((d-1)k^3 + dkn))$ operations for the evaluation of a single integral value. However this is not optimal since we can easily identify a repeated computation of the same data that refer to the parameter vector $a \in I$ describing the geometric configuration of the cuboids.

First note that whenever $\mu \in D_I$, the vectors u_t, $t = \{\mu\} \in \mathcal{L}(T_D)$, from (18) do not change for all $r_\varphi r_\psi$ terms in (16). Once they are computed, we can reuse them in all computations involving different values of the coefficients $c_{\mu,i}$ from (19) for $\mu \in D_X \cup D_Y$. The evaluation in the leaves $t = \{\mu\}$ for $\mu \in D_I$ requires $\mathcal{O}(pkn)$ operations. Then the summations at the inner nodes $t \in T_D \setminus \mathcal{L}(T_D)$ for which $t \cap D_3 \neq \emptyset$ can be carried out in $\mathcal{O}(pk^3)$. All further evaluations then require only $\mathcal{O}(5k^3 + 6kn)$ operations. The overall cost is therefore $\mathcal{O}(pk^3 + pkn + r_\varphi r_\psi(5k^3 + 6kn))$.

4 Parametrisation of Cuboids

The aim of this section is to find parametrisations of the cuboids $X(a), Y(a)$ in such a way that the function $f_{i,j}$ from (12) is smooth for all $a \in I \subset \mathbb{R}^p$. Given a fixed coordinate system, each cuboid can be described in terms of its corners by a 6-dimensional parameter tuple. Therefore, in general, we have to deal with a 12-dimensional parameter space I. However, we are not always forced to use a fixed coordinate system to describe the geometrical setting. If the integral value does not change under certain coordinate transformations, a local coordinate system which requires less parameters is sufficient to describe the relative position of the cuboids. Moreover, if the two cuboids intersect, a further reduction of the dimension of the parameter space is possible.

Local Coordinate Systems

A local coordinate system can be defined by taking advantage of the properties of the kernel function κ.

Definition 6 (kernel properties)**.** We call $\kappa : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$

(a) *translationally invariant* if for all $c \in \mathbb{R}^3$

$$\kappa(x + c, y + c) = \kappa(x, y), \qquad x, y \in \mathbb{R}^3, \tag{20}$$

(b) *homogeneous of degree* α if for all $\lambda > 0$

$$\kappa(\lambda x, \lambda y) = \lambda^{\alpha} \kappa(x, y), \qquad x, y \in \mathbb{R}^3. \tag{21}$$

In the following, we assume that κ is at least translationally invariant. W.l.o.g., we may therefore assume that the cuboids X, Y from (2) are defined by

$$X_\mu = [0, X_{\mu,2}], \quad Y_\mu = [Y_{\mu,1}, Y_{\mu,2}], \quad \mu = 1, 2, 3.$$

Note that the Lagrange polynomials in expression (12) do not change under translations of the coordinate system since they were defined with respect to the cuboids X, Y.

Given a suitable local numbering of the coordinate directions $\mu = 1, 2, 3$, the following five situations may occur in a standard conforming mesh:

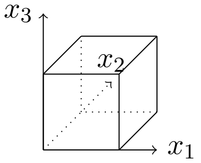
1. the case of identical cuboids $X = Y$ which leads to a parameter space of dimension $p = 3$,
2. the case of cuboids with a common face with $Y_{2,1} = Y_{3,1} = 0$ and $Y_{1,1} = X_{1,2}$, $Y_{2,2} = X_{2,2}$, $Y_{3,2} = X_{3,2}$ which leads to a parameter space of dimension $p = 4$,
3. the case of cuboids with a common edge with $Y_{3,1} = 0$ and $Y_{1,1} = X_{1,2}$, $Y_{2,1} = X_{2,2}$, $Y_{3,2} = X_{3,2}$ such that $p = 5$,
4. the case of cuboids with a common vertex wih $Y_{1,1} = X_{1,2}$, $Y_{2,1} = X_{2,2}$, $Y_{3,1} = X_{3,2}$ such that $p = 6$, and finally
5. the case of disjoint cuboids such that $p = 9$.

In all five cases, a parametrisation of the cuboids X, Y by a parameter tuple $a \in I \subset \mathbb{R}^p$ can be obtained in a straighforward way.

The Case of Identical Cuboids

The parameter space is of dimension $p = 3$. We set

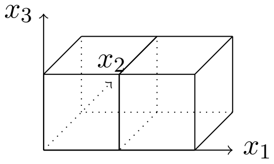
$$\begin{aligned} X_{1,2} &:= a_1, \\ X_{2,2} &:= a_2, \\ X_{3,2} &:= a_3. \end{aligned}$$



The Case of Cuboids with a Common Face

The parameter space is of dimension $p = 4$. We set

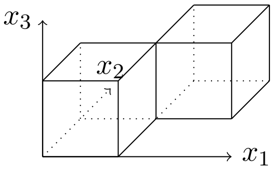
$$\begin{aligned} X_{1,2} &:= a_1, & Y_{1,2} &:= a_1 + a_4, \\ X_{2,2} &:= a_2, \\ X_{3,2} &:= a_3. \end{aligned}$$



The Case of Cuboids with a Common Edge

The parameter space is of dimension $p = 5$. We set

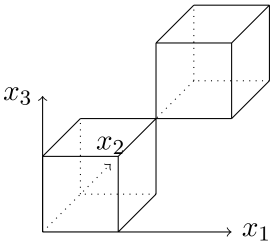
$$\begin{aligned} X_{1,2} &:= a_1, & Y_{1,2} &:= a_1 + a_4, \\ X_{2,2} &:= a_2, & Y_{2,2} &:= a_2 + a_5, \\ X_{3,2} &:= a_3. \end{aligned}$$



The Case of Cuboids with a Common Vertex

The parameter space is of dimension $p = 6$. We set

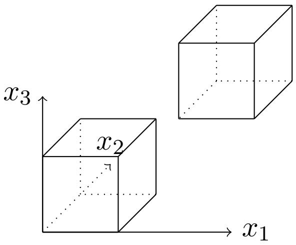
$$\begin{aligned} X_{1,2} &:= a_1, & Y_{1,2} &:= a_1 + a_4, \\ X_{2,2} &:= a_2, & Y_{2,2} &:= a_2 + a_5, \\ X_{3,2} &:= a_3, & Y_{3,2} &:= a_3 + a_6. \end{aligned}$$



The Case of Disjoint Cuboids

The parameter space is of dimension $p = 9$. We set

$$\begin{aligned} X_{1,2} &:= a_1, & Y_{1,1} &:= a_1 + a_7, & Y_{1,2} &:= a_1 + a_4 + a_7, \\ X_{2,2} &:= a_2, & Y_{2,1} &:= a_2 + a_8, & Y_{2,2} &:= a_2 + a_5 + a_8, \\ X_{3,2} &:= a_3, & Y_{3,1} &:= a_3 + a_9, & Y_{3,2} &:= a_3 + a_6 + a_9. \end{aligned}$$



Note that all parametrisations have been chosen in such a way that one can easily define suitable parameter spaces $I \subset \mathbb{R}^p$ such that reasonable geometric configurations can be guaranteed for all $a = (a_1, \ldots, a_p) \in I$. Moreover, the smoothness of the parametrisation of the corners of the cuboids X, Y guarantees the smoothness of the function $f_{i,j}$ from (12) for all $a \in I$. This result directly follows from basic tools from shape calculus mimicking the proof in [1] which is not repeated here.

5 Hierarchical Quadrature

The construction of a data-sparse approximation of the tensor A from (4) at least requires the ability of evaluating entries of A at given indices. Since each entry of A corresponds to the evaluation of a singular integral, this is a challenging task itself. Here, we focus on the technique of hierarchical quadrature which was introduced for the one-dimensional case in [4]. A key feature of this approach is to express singular integrals in terms of regular ones by the solution of an appropriately chosen (simple) linear system. In [10, 11], we could generalise this method to the computation of multidimensional singular integrals over hypercubes.

In the following, we sketch the main idea of hierarchical quadrature by a simple one-dimensional example which can also be found in [4]. We assume that the kernel function κ is translationally invariant (20) and homogeneous of degree α (21). To simplify the presentation, let the orders of the Lagrange polynomials in (12) by given by $m := m_X = m_Y$.

One-Dimensional Case

In the one-dimensional case, we have to compute integrals

$$f_{i,j} := \int_X \int_Y L_i^X(x)\kappa(x,y)L_j^Y(y)\,\mathrm{d}y\,\mathrm{d}x$$

over intervals $X := [\alpha_x, \beta_x]$, $Y := [\alpha_y, \beta_y]$. In order to transform the integral to the reference domain $[0,1]^2$, we introduce mappings

$$\begin{aligned}\Phi(x) &:= x(\beta_x - \alpha_x),\\ \Psi(y) &:= y(\beta_y - \alpha_y).\end{aligned}$$

We first consider the case of identical elements, i.e. when $X = Y$. Due to (20), we may then write

$$\begin{aligned}f_{i,j} &= c_{\mathrm{det}} \int_0^1 \int_0^1 L_i^X(\alpha_x + \Phi(x))\kappa(\alpha_x + \Phi(x), \alpha_y + \Psi(y))L_j^Y(\alpha_y + \Psi(y))\,\mathrm{d}y\,\mathrm{d}x\\ &= c_{\mathrm{det}} \int_0^1 \int_0^1 L_i(x)\kappa(\Phi(x), \Psi(y))L_j(y)\,\mathrm{d}y\,\mathrm{d}x,\end{aligned}$$

where $c_{\mathrm{det}} := |\det \Phi'|\,|\det \Psi'|$ and $L_i := L_i^{[0,1]}$. For $\Omega \subset \Omega_0 := [0,1]^2$, we introduce integrals

$$f_{i,j}^{\Omega,(\lambda,\nu)} := \iint_\Omega L_i(x)\kappa(\Phi(\lambda x), \Psi(\nu y))L_j(y)\,\mathrm{d}y\,\mathrm{d}x, \tag{22}$$

where $\lambda, \nu \in \{-1, 1\}$. With this notation, we have

$$f_{i,j} = c_{\mathrm{det}} f_{i,j}^{\Omega_0,(1,1)}.$$

The domain Ω_0 is split into four subdomains

$$\Omega_1 := [0, \tfrac{1}{2}] \times [0, \tfrac{1}{2}], \quad \Omega_2 := [0, \tfrac{1}{2}] \times [\tfrac{1}{2}, 1], \quad \Omega_3 := [\tfrac{1}{2}, 1] \times [0, \tfrac{1}{2}], \quad \Omega_4 := [\tfrac{1}{2}, 1] \times [\tfrac{1}{2}, 1],$$

such that $\Omega_0 = \Omega_1 \cup \Omega_2 \cup \Omega_3 \cup \Omega_4$. Omitting the constant c_{det}, the integral can now be represented by the sum

$$f_{i,j}^{\Omega_0,(\lambda,\nu)} = f_{i,j}^{\Omega_1,(\lambda,\nu)} + f_{i,j}^{\Omega_2,(\lambda,\nu)} + f_{i,j}^{\Omega_3,(\lambda,\nu)} + f_{i,j}^{\Omega_4,(\lambda,\nu)}. \tag{23}$$

For $\lambda = \nu = 1$, all four terms are again singular. Our aim is now to express the integrals on the right-hand side of (23) in terms of integrals of the form (22) with $\Omega = \Omega_0$. Due to (21), we can write the first term as

$$f_{i,j}^{\Omega_1,(1,1)} = c_{\text{hom}} \iint_{\Omega_0} L_i\left(\frac{x}{2}\right) \kappa(\Phi(x), \Psi(y)) L_j\left(\frac{y}{2}\right) \,\mathrm{d}y \,\mathrm{d}x$$

with $c_{\text{hom}} := \frac{1}{4}\left(\frac{1}{2}\right)^{\alpha}$. In order to get rid of the factor $\frac{1}{2}$ within the arguments of the Lagrange polynomials, we introduce transfer matrices $T^{\delta} \in \mathbb{R}^{(m+1)\times(m+1)}$ by

$$T_{i,j}^{\delta} := L_i\left(\frac{\xi_j^{[0,1]} + \delta}{2}\right), \quad \delta = 0, 1.$$

Clearly,

$$L_i\left(\frac{x+\delta}{2}\right) = \sum_{s=0}^{m} T_{i,s}^{\delta} L_s(x).$$

We can therefore write

$$f_{i,j}^{\Omega_1,(1,1)} = c_{\text{hom}} \sum_{s=0}^{m}\sum_{t=0}^{m} T_{i,s}^{0} T_{j,t}^{0} f_{s,t}^{\Omega_0,(1,1)}.$$

Analogously, we get for the fourth term from (23)

$$f_{i,j}^{\Omega_4,(1,1)} = c_{\text{hom}} \sum_{s=0}^{m}\sum_{t=0}^{m} T_{i,s}^{1} T_{j,t}^{1} f_{s,t}^{\Omega_0,(1,1)}.$$

The second term from (23) reads

$$\begin{aligned}
f_{i,j}^{\Omega_2,(1,1)} &= \int_0^{1/2}\int_{1/2}^{1} L_i(x)\,\kappa(\Phi(x), \Psi(y)) L_j(y) \,\mathrm{d}y \,\mathrm{d}x \\
&= \int_0^{1/2}\int_0^{1/2} L_i\left(\frac{1}{2} - x\right) \kappa\left(\Phi\left(\frac{1}{2} - x\right), \Psi\left(y + \frac{1}{2}\right)\right) L_j\left(y + \frac{1}{2}\right) \,\mathrm{d}y \,\mathrm{d}x \\
&= c_{\text{hom}} \int_0^1\int_0^1 L_i\left(\frac{1-x}{2}\right) \kappa(\Phi(-x), \Psi(y)) L_j\left(\frac{y+1}{2}\right) \,\mathrm{d}y \,\mathrm{d}x \\
&= c_{\text{hom}} \int_0^1\int_0^1 L_{m-i}\left(\frac{x+1}{2}\right) \kappa(\Phi(-x), \Psi(y)) L_j\left(\frac{y+1}{2}\right) \,\mathrm{d}y \,\mathrm{d}x \\
&= c_{\text{hom}} \sum_{s=0}^{m}\sum_{t=0}^{m} T_{m-i,s}^{1} T_{j,t}^{1} f_{s,t}^{\Omega_0,(-1,1)}.
\end{aligned}$$

Equivalently, the third term from (23) can be expressed by

$$f_{i,j}^{\Omega_3,(1,1)} = c_{\text{hom}} \sum_{s=0}^{m}\sum_{t=0}^{m} T_{i,s}^{1} T_{m-j,t}^{1} f_{s,t}^{\Omega_0,(1,-1)}.$$

The terms $f_{i,j}^{\Omega_0,(-1,1)}$ and $f_{i,j}^{\Omega_0,(1,-1)}$ represent integrals over point singularities. With the same splitting strategy as before, these terms can be expressed as a sum of the form (23). Now, only the first term in the sum is singular for which we can derive

$$f_{i,j}^{\Omega_1,(-1,1)} = c_{\mathrm{hom}} \sum_{s=0}^{m} \sum_{t=0}^{m} T_{i,s}^0 T_{j,t}^0 f_{s,t}^{\Omega_0,(-1,1)}.$$

And equivalent expression holds for $f_{i,j}^{\Omega_1,(1,-1)}$.

The relation between the different integral values can now be expressed by the linear system

$$(\mathbf{Id} - c_{\mathrm{hom}}\mathbf{M})\mathbf{f} = \mathbf{b} \tag{24}$$

with

$$\mathbf{M} := \begin{bmatrix} M^{1,1} & M^{1,2} & M^{1,3} \\ 0 & M^{2,2} & 0 \\ 0 & 0 & M^{3,3} \end{bmatrix}, \quad \mathbf{f} := \begin{bmatrix} f^{\Omega_0,(1,1)} \\ f^{\Omega_0,(-1,1)} \\ f^{\Omega_0,(1,-1)} \end{bmatrix}, \quad \mathbf{b} := \begin{bmatrix} 0 \\ b^2 \\ b^3 \end{bmatrix},$$

where

$$\begin{aligned} &M^{1,1} := T^0 \otimes T^0 + T^1 \otimes T^1, \quad M^{1,2} := JT^1 \otimes T^1, \quad M^{1,3} := T^1 \otimes JT^1, \\ &M^{2,2} = M^{3,3} := T^0 \otimes T^0, \end{aligned}$$

and

$$b_{i,j}^2 := f_{i,j}^{\Omega_0 \setminus \Omega_1,(-1,1)}, \qquad b_{i,j}^3 := f_{i,j}^{\Omega_0 \setminus \Omega_1,(1,-1)}.$$

Here, $J \in \mathbb{R}^{(m+1)\times(m+1)}$ with $J_{i,j} := \delta_{i,m-j}$, $i,j = 0,\dots,m$, denotes the standard flip matrix.

The sytem (24) is non-singular as long as $\alpha \neq -1,-2$ (cf. [10]). Note that the right-hand side $\mathbf{b}$ contains only regular integrals which can be computed by standard quadrature routines. The first block of the solution vector $\mathbf{f}$ comprises the integral values $f_{i,j}^{\Omega_0,(1,1)}$ for *all* indices $i,j = 0,\dots,m$.

Let us now consider the case of two elements sharing a single point, i.e. $\beta_x = \alpha_y$ or $\alpha_x = \beta_y$. In the first case, we can write

$$\begin{aligned} f_{i,j} &= c_{\mathrm{det}} \int_0^1 \int_0^1 L_i^X(\beta_x + \Phi(-x))\kappa(\beta_x + \Phi(-x), \alpha_y + \Psi(y)) L_j^Y(\alpha_y + \Psi(y)) \, \mathrm{d}y \, \mathrm{d}x \\ &= c_{\mathrm{det}} \int_0^1 \int_0^1 L_i(1-x)\kappa(\Phi(-x), \Psi(y)) L_j(y) \, \mathrm{d}y \, \mathrm{d}x \\ &= c_{\mathrm{det}} \int_0^1 \int_0^1 L_{m-i}(x)\kappa(\Phi(-x), \Psi(y)) L_j(y) \, \mathrm{d}y \, \mathrm{d}x \\ &= c_{\mathrm{det}} f_{m-i,j}^{\Omega_0,(-1,1)}. \end{aligned}$$

The values $f_{i,j}^{\Omega_0,(-1,1)}$ may be obtained from a simple linear system which corresponds to the second row of the block system (24).

Multidimensional Case

The splitting strategy from the one-dimensional example can be generalised to the n-dimensional case. For the details, we refer the reader to [10]. The linear system is then again of the form (24) where $\mathbf{M}$ is an upper triangular block matrix

$$\mathbf{M} = \begin{bmatrix} M^{1,1} & \cdots & \cdots & M^{1,s} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & M^{s,s} \end{bmatrix}$$

where each block $M^{\sigma,\tau}$ can be represented as

$$M^{\sigma,\tau} = \sum_{j=1}^{r} \bigotimes_{\mu=1}^{2n} M_{\mu,j}^{\sigma,\tau},$$

with $M_{\mu,j}^{\sigma,\tau} \in \{0, T^0, T^1, JT^1\}$. The scaling constant is then given by $c_{\text{hom}} = \left(\frac{1}{2}\right)^{2n+\alpha}$ such that (24) is non-singular for $\alpha \neq -2n + n'$ with $n' = 0, \ldots, n$. Note that by an alternative splitting strategy, the restrictions on the degree of the homogeneity α can be avoided, cf. [11].

Solution of the Parameter Dependent System

So far, we have considered a fixed geometric configuration of the cuboids X and Y. If we include the dependence on a parameter vector $a \in I \subset \mathbb{R}^p$ into our formulation, we end up with a linear system of the form

$$(\mathbf{Id} - c_{\text{hom}}\mathbf{M})\mathbf{f}(a) = \mathbf{b}(a).$$

After a discretisation of the parameter a in I, each block of $\mathbf{b} = [b^1, \ldots, b^s]^\top$ can be interpreted as a tensor $b^\sigma \in \mathbb{R}^{\mathcal{I}}$ with index set $\mathcal{I}$ from (14). This tensor may be approximated in the hierarchical format $\mathcal{H}_{\boldsymbol{k}}$ by the techniques presented in Section 2.3.

Since each block of the upper triangular system matrix possesses Kronecker product structure, solution methods which explicitly exploit the tensor structure become applicable. For a detailed investigation of these methods we refer the reader to [13] and the references therein. In our application, we have used a truncated version of a GMRES method to obtain the (approximate) solution $\mathbf{f}$ such that each block of $\mathbf{f}$ is represented in the hierarchical format $\mathcal{H}_{\boldsymbol{k}}$. The first block of $\mathbf{f}$ then corresponds to the sought tensor A from (4).

6 Numerical Examples

In this section, we analyse the numerical properties of our approximation scheme for the evaluation of integrals of the form (1). The purpose of the chosen numerical examples is two-fold. First, we demonstrate that the geometric parametrisation of cuboids in terms of a parameter vector $a \in I \subset \mathbb{R}^p$ leads to functions of the form (12) which can be well approximated by interpolation. Second, we show that the resulting tensor $A \in \mathbb{R}^{\mathcal{I}}$ from (4) possesses a data-sparse representation in the hierarchical format $\mathcal{H}_{\boldsymbol{k}}$.

In all numerical examples, we fix the order of the Lagrange polynomials from (12) by $m_X = m_Y := 10$. Using the parametrisations from Section 4, we define the parameter space by $I := [0.5, 2]^p \subset \mathbb{R}^p$ which accounts for typical geometrical configurations in a volume mesh. In order not to overload the presentation, we restrict ourselves to kernel functions κ of the form

$$\kappa(x, y) = \|x - y\|^{\alpha}, \quad \alpha \in \mathbb{R}.$$

In the singular case, we can then compute the tensor entries of A from (4) by hierarchical quadrature. For the computation of the regular integrals and in the case of disjoint cuboids, we use Gauss quadrature of sufficiently large order such that the relative quadrature error lies below 10^{-8}.

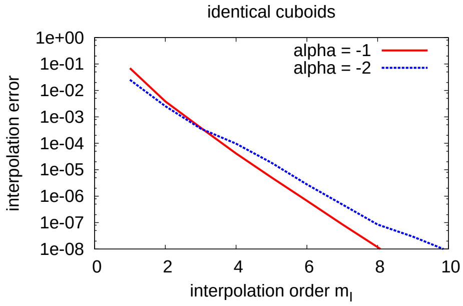
To test the interpolation quality within the set I, we compute test integrals of the form

$$\int_X \int_Y \frac{x_1 x_2 x_3 y_1 y_2 y_3}{\|x - y\|^{-\alpha}} \, \mathrm{d}x \, \mathrm{d}y$$

for 100 randomly chosen parameter samples from I. Note that for $\alpha = -1$, an analytic integration is possible [6] to which we can compare the results of our tensor structured approach. For other values of α, we use a relatively high interpolation order of $m_I := 15$ to test the interpolation error within I.

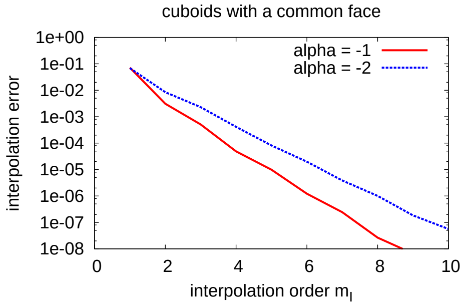
Once we have determined the required interpolation order m_I to reach a given relative accuracy of ε, we analyse the properties of the tensor A from (4). To this end, we first determine a highly accurate approximation of A in the hierarchical format $\mathcal{H}_{\boldsymbol{k}}$ from the solution of the linear system (24), where the right-hand side has been determined by the adaptive (but heuristic) black box strategy from [2]. This tensor is truncated to a tensor A_ε such that $\|A - A_\varepsilon\|_2 \leq \varepsilon \|A\|_2$. Depending on the accuracy ε, we monitor the behaviour of the hierarchical rank of A_ε. To facilitate the comparison of the hierarchical ranks for different tensors, we compute the so-called *effective rank* k_{eff} which corresponds to a storage complexity of A_ε of size $\mathcal{O}((d-1)k_{\text{eff}}^3 + dk_{\text{eff}} \sum_{\mu=1}^{d} \#\mathcal{I}_\mu)$.

For $\alpha = -1$ and $\alpha = -2$, we report our numerical results for the case of identical cuboids in Table 1, for the case of cuboids with a common face in Table 2, for the case of cuboids with a common edge in Table 3, for the case of cuboids with a common vertex in Table 4, and in case of disjoint cuboids in Table 5. On the left-hand side, we observe that in all five cases the interpolation error decays exponentially with the interpolation order m_I. On the right-hand side, we can see that the required effective rank k_{eff} for the representation of the truncated tensor A_ε increases only mildly with the accuracy ε. This means that even for high accuracies, functions of the type (12) can be approximated in a structured and data-sparse way.



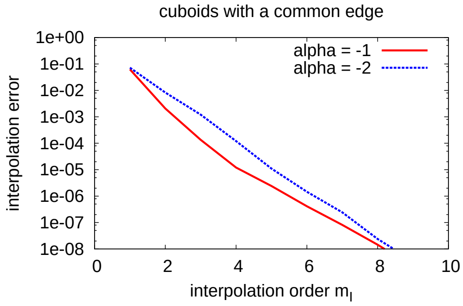
	$\alpha = -1$		$\alpha = -2$	
ε	m_I	k_{eff}	m_I	k_{eff}
1e-03	3	6.8	3	6.9
1e-04	4	7.4	4	7.6
1e-05	5	8.4	6	8.4
1e-06	6	9.0	7	9.3
1e-07	7	9.9	8	10.4

Table 1: Identical cuboids with $p = 3$. Left: interpolation error. Right: interpolation orders m_I and ranks k_{eff} of the tensor approximation to $\mathcal{H}_{\boldsymbol{k}}$ for different accuracies ε.



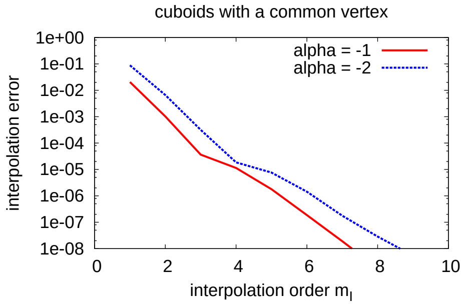
	$\alpha = -1$		$\alpha = -2$	
ε	m_I	k_{eff}	m_I	k_{eff}
1e-03	3	5.6	4	6.8
1e-04	4	6.8	5	8.0
1e-05	5	8.0	7	9.0
1e-06	7	9.7	9	10.2
1e-07	8	11.9	10	12.2

Table 2: Cuboids with a common face with $p = 4$. Left: interpolation error. Right: interpolation orders m_I and ranks k_{eff} of the tensor approximation to $\mathcal{H}_{\boldsymbol{k}}$ for different accuracies ε.



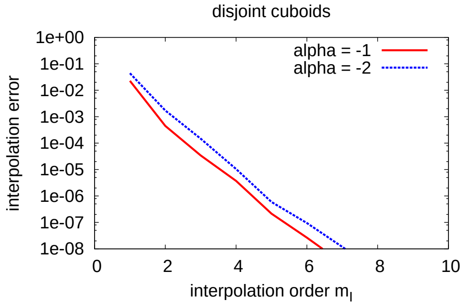
	$\alpha = -1$		$\alpha = -2$	
ε	m_I	k_{eff}	m_I	k_{eff}
1e-03	3	4.3	4	5.8
1e-04	4	5.8	5	7.7
1e-05	5	7.5	6	9.1
1e-06	6	8.8	7	10.5
1e-07	7	11.7	8	12.2

Table 3: Cuboids with a common edge with $p = 5$. Left: interpolation error. Right: interpolation orders m_I and ranks k_{eff} of the tensor approximation to $\mathcal{H}_{\boldsymbol{k}}$ for different accuracies ε.



ε	$\alpha=-1$ m_I	k_{eff}	$\alpha=-2$ m_I	k_{eff}
1e-03	3	3.4	3	4.7
1e-04	4	4.9	4	6.6
1e-05	5	6.6	5	8.4
1e-06	6	8.2	7	10.1
1e-07	7	10.1	8	11.6

Table 4: Cuboids with a common vertex with $p = 6$. Left: interpolation error. Right: interpolation orders m_I and ranks k_{eff} of the tensor approximation to $\mathcal{H}_{\boldsymbol{k}}$ for different accuracies ε.



ε	$\alpha=-1$ m_I	k_{eff}	$\alpha=-2$ m_I	k_{eff}
1e-03	2	2.8	3	3.0
1e-04	3	3.4	4	4.1
1e-05	4	4.2	5	5.0
1e-06	5	5.2	6	6.0
1e-07	6	6.5	7	7.3

Table 5: Disjoint cuboids with $p = 9$. Left: interpolation error. Right: interpolation orders m_I and ranks k_{eff} of the tensor approximation to $\mathcal{H}_{\boldsymbol{k}}$ for different accuracies ε.

7 Conclusion

In this article, we have introduced a tensor-structured scheme for the evaluation of singular volume integral over cuboids. The key ingredient of our approach was the interpretation of singular integrals as high-dimensional functions to which data-sparse tensor approximations can be applied. For a number of standard numerical examples, we have shown that data-sparsity can be maintained even for high accuracies. This means that highly accurate integral values can be obtained by a small number of simple arithmetic operations in a fast and stable way.

As was already shown in [1], it is possible to include further parameters into our formulation, as e.g. the wave number for Helmholtz problems. This approach can be used for a simultaneous approximation of singular integral values for a fixed parameter range. Moreover, we would like to stress that our scheme is not restricted to integrals over cuboids. One can also generalise our method to singular integrals over tetrahedra. However, in this case the appropriate choice of the geometric parameters is much more challenging and is left for further investigations.

References

[1] J. Ballani. Fast evaluation of singular BEM integrals based on tensor approximations. *Numer. Math.*, 121(3):433–460, 2012.

[2] J. Ballani, L. Grasedyck, and M. Kluge. Black box approximation of tensors in hierarchical Tucker format. *Linear Algebra Appl.*, to appear.

[3] C. Bertoglio and B. N. Khoromskij. Low-rank quadrature-based tensor approximation of the Galerkin projected Newton/Yukawa kernels. *Comput. Phys. Comm.*, 183(4):904–912, 2012.

[4] S. Börm and W. Hackbusch. Hierarchical quadrature of singular integrals. *Computing*, 74:75–100, 2005.

[5] L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31:2029–2054, 2010.

[6] W. Hackbusch. Direct integration of the Newton potential over cubes. *Computing*, 68(3):193–216, 2002.

[7] W. Hackbusch. Entwicklungen nach Exponentialsummen. Preprint 4/2005, Max Planck Institute for Mathematics in the Sciences, 2005.

[8] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer, Berlin, 2012.

[9] W. Hackbusch and S. Kühn. A new scheme for the tensor representation. *J. Fourier Anal. Appl.*, 15(5):706–722, 2009.

[10] P. Meszmer. Hierarchical quadrature for multidimensional singular integrals. *J. Numer. Math.*, 18(2):91–117, 2010.

[11] P. Meszmer. Hierarchical quadrature for multidimensional singular integrals – part II. Preprint xx/2012, Max Planck Institute for Mathematics in the Sciences, 2012.

[12] I. V. Oseledets and E. E. Tyrtyshnikov. Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM J. Sci. Comp.*, 31(5):3744–3759, 2009.

[13] C. Tobler. *Low-rank Tensor Methods for Linear Systems and Eigenvalue Problems.* PhD thesis, ETH Zürich, 2012.