Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Fast Evaluation of Singular BEM Integrals Based
on Tensor Approximations

by

Jonas Ballani

Preprint no.: 77

2010

Fast Evaluation of Singular BEM Integrals Based on Tensor Approximations

Jonas Ballani

December 21, 2010

In this paper we propose a method for the fast evaluation of integrals stemming from boundary element methods. Our method is based on the parametrisation of boundary elements in terms of a d -dimensional parameter tuple. We interpret the integral as a real-valued function f depending on d parameters and show that f is smooth in a d -dimensional box. A standard interpolation of f by polynomials leads to a d -dimensional tensor which is given by the values of f at the interpolation points. This tensor may be approximated in a low rank tensor format like the (CP) format or the $\mathcal{H}$ -Tucker format. The tensor approximation has to be done only once and allows us to evaluate interpolants in $\mathcal{O}(dr(m+1))$ operations in the (CP) format, or $\mathcal{O}(dk^3 + dk(m+1))$ operations in the $\mathcal{H}$ -Tucker format, where m denotes the interpolation order and the ranks r, k are small integers. We demonstrate that highly accurate integral values can be obtained at very moderate costs.

1 Introduction

In boundary element methods one often is required to evaluate integrals of the form

$$\int_{\sigma} \int_{\tau} \kappa(\|x - y\|) dy dx \quad (1)$$

where $\sigma, \tau \subset \Gamma$ are boundary elements on the boundary $\Gamma := \partial\Omega$ of a domain $\Omega \subset \mathbb{R}^{\ell}$ ($\ell = 2, 3$) and $\kappa : \mathbb{R} \rightarrow \mathbb{R}$ is a kernel function that has a singularity at 0 and is analytic elsewhere. Typically, a Galerkin discretisation of an integral equation over the boundary Γ with constant basis functions leads to this type of integrals. A significant part of the assembly time of the corresponding boundary element matrix is spent by calculating nearfield integrals of the form (1). In particular, this includes the calculation of integrals for which the closures of σ and τ are not disjoint and the integrand has at least one singularity. In this case, standard quadrature techniques can only be applied with high quadrature orders in order to achieve high accuracies which leads to high computational costs. It is therefore, in particular, necessary to accelerate the evaluation time of the singular integrals.

1.1 Evaluation by Interpolation

In this paper, we propose a new method that prepares suitable auxiliary data in a setup phase and then computes a large number of integral values very efficiently. The key idea behind our new approach is the investigation of functions of the type

$$f(a) := \int_{\sigma(a)} \int_{\tau(a)} \kappa(\|x - y\|) dy dx \quad (2)$$

where $a \in \mathbb{R}^d$ is a suitable parameter tuple describing the domains $\sigma(a)$ and $\tau(a)$. In practice, we can choose a suitable subset $Q \subseteq \mathbb{R}^d$ such that f is analytic in Q . Therefore we expect to

be able to approximate f on Q by, e.g., polynomials that can be evaluated very efficiently. A straightforward way to obtain an approximation of f by polynomials is to apply multivariate interpolation. The values of f in the interpolation points $\{\xi_{\mu,\nu}\}_{\mu=1,\dots,d,\nu=0,\dots,m}$ define a tensor $A \in \mathbb{R}^{(m+1)^d}$ given by

$$A_{(\nu_1,\dots,\nu_d)} := f(\xi_{1,\nu_1}, \dots, \xi_{d,\nu_d}), \quad \nu_\mu = 0, \dots, m, \mu = 1, \dots, d. \quad (3)$$

The interpolation on its own will not lead to an efficient scheme since the tensor A contains $(m+1)^d$ entries which even may exceed the available memory for $d > 6$. 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 in a data sparse way:

1. The Tucker format which represents a tensor in a storage complexity of $\mathcal{O}(k^d + dk(m+1))$.
2. The (CP) format (a.k.a. CANDECOMP/PARAFAC, r -term representation) which represents a tensor in a storage complexity of $\mathcal{O}(dr(m+1))$.
3. The $\mathcal{H}$ -Tucker format by Hackbusch and Kühn [7] which represents a tensor in a storage complexity of $\mathcal{O}(dk^3 + dk(m+1))$. This format also contains the (TT) format introduced by Tyrtysnikov and Oseledets [10] as a special case.

Here, r and k denote ranks in the specific format which depend on the tensor which has to be represented. If these ranks remain small, the last two formats lead to an efficient representation of a tensor even in high dimensions. The question is now how to find such a representation from (3).

The tensor approximation problem can be split into two cases. For moderate dimensions, say $d \leq 6$, it is possible to calculate all tensor entries from (3) since typically $m \leq 20$. For $d > 6$, this becomes intractable and one has to find an approximate tensor from (3) in a black box fashion.

For the (CP) format, we have to deal with an additional difficulty because the approximation problem is known to be ill-posed and a best approximation does not necessarily exist, see [8] for an introduction. Although the approximation problem may be regularised, it is not known how to find an approximation of the best approximation even if the tensor is available in its full representation. Nevertheless, one can look for a tensor that approximates A up to a given accuracy by gradient-like methods (ALS) or Newton-type methods. The construction of an approximate tensor in (CP) format in a black box fashion has been analysed by Espig, Grasedyck, and Hackbusch [5].

The situation changes completely if we look for an approximate tensor in $\mathcal{H}$ -Tucker format. In this case, an approximation may be calculated by a hierarchical singular value decomposition where a priori error bounds are available [6]. Similar results have been obtained by Oseledets and Tyrtysnikov [10] for the (TT) format. In [11], they also suggest a method for the approximation of a tensor in (TT) format in a black box fashion. We could further generalise these results to the $\mathcal{H}$ -Tucker format in [1].

1.3 Quadrature Techniques for Singular Integrals

We still have left open, how the entries of the tensor A can be computed. Each entry of A corresponds to the value of a singular integral which can be calculated by several techniques. Some of the most important are the following:

1. If σ and τ are sufficiently simple, it is possible to derive quadrature rules that take the particular singularity into account, e.g., by introducing suitable weight functions [15], [14]. This approach works only if appropriate quadrature rules can be derived and if the corresponding quadrature weights can be computed which again correspond to singular integrals.

2. If one of the integrals can be computed analytically in closed form and if the remaining integral is sufficiently regular, the second integral can be approximated by a standard quadrature rule [15], [9], [12].
3. A transformation may be applied that removes the singularity [4], [13]. This approach is very general and can be applied to a variety of important problem classes. Unfortunately, the domain of analyticity of the resulting integrand is not very large which requires the usage of high quadrature orders corresponding to high computational costs.

Since the (approximate) calculation of the entries of A is required only once, we can afford to use the method of [4], [13] using high quadrature orders. This enables us to treat a wide variety of kernel functions by our interpolation-based method.

1.4 This Paper

This paper is organised as follows. In Section 2 we introduce some basic definitions related to interpolation and tensor representations. In Section 3 we present the detailed approximation scheme which allows for an efficient evaluation of interpolants. The tensor approximation problem will be discussed in Section 4. Afterwards we will introduce concrete parametrisations of boundary elements for the practically relevant case $\Omega \subset \mathbb{R}^3$, where the boundary Γ is discretised by flat triangles. A proof will be given that shows that f is analytic in a box $Q \subset \mathbb{R}^d$. We will restrict our attention to the case of singular integrals but our techniques are also applicable to regular cases. In Section 6 we will briefly analyse the error caused by the approximation scheme. Finally, we illustrate the potential of our new approach by a number of numerical examples in Section 7.

2 Basic Definitions

2.1 Multivariate Interpolation

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

$$Q := [s_1, t_1] \times \dots \times [s_d, t_d].$$

Let us first define interpolation on intervals. For an interval $[s, t] \subset \mathbb{R}$, let $\{\xi_i\}_{i=0, \dots, m}$ be interpolation points with $s \leq \xi_i \leq t$, $i = 0, \dots, m$. The corresponding Lagrange polynomials have the form

$$\mathcal{L}_\nu(x) = \prod_{i=0, i \neq \nu}^m \frac{x - \xi_i}{\xi_\nu - \xi_i}, \quad \nu = 0, \dots, m.$$

The one-dimensional interpolation operator is given by

$$\mathfrak{I}_m^{[s, t]} : C[s, t] \rightarrow \mathcal{P}_m, \quad f \mapsto \sum_{\nu=0}^m f(\xi_\nu) \mathcal{L}_\nu,$$

where $\mathcal{P}_m$ is the space of polynomials of degree at most m . Due to the product structure of Q , we can define the d -dimensional interpolation operator

$$\mathfrak{I}_m^Q := \mathfrak{I}_m^{[s_1, t_1]} \otimes \dots \otimes \mathfrak{I}_m^{[s_d, t_d]} \quad (4)$$

as the tensor product of its one-dimensional counterparts. Now f can be approximated by the d -dimensional interpolation function $f_m := \mathfrak{I}_m^Q[f]$ which gives

$$f_m(x) = \sum_{\nu_1=0}^m \dots \sum_{\nu_d=0}^m f(\xi_{1, \nu_1}, \dots, \xi_{d, \nu_d}) \mathcal{L}_{1, \nu_1}(x_1) \dots \mathcal{L}_{d, \nu_d}(x_d)$$

where $\{\xi_{\mu,\nu}\}_{\mu=1,\dots,d,\nu=0,\dots,m}$ are interpolation points with $s_\mu \leq \xi_{\mu,\nu} \leq t_\mu$, $\mu = 1, \dots, d$, $\nu = 0, \dots, m$ and $\mathcal{L}_{\mu,\nu}$ are corresponding Lagrange polynomials.

We still have left open which interpolation points we want to use. For an optimal bound on the maximum norm of the error on the interval $[-1, 1]$, we choose the m -th order Chebyshev points given by

$$\xi_\nu := \cos\left(\frac{2\nu+1}{2m+2}\pi\right), \quad \nu = 0, \dots, m.$$

The interpolation points may be transformed to a given interval $[s, t]$ by the elementary relation

$$\xi_\nu^{[s,t]} := \frac{t+s}{2} + \frac{t-s}{2}\xi_\nu, \quad \nu = 0, \dots, m.$$

2.2 Tensor Representations

Notation 1 (Index set). Let $d \in \mathbb{N}$ and $n_1, \dots, n_d \in \mathbb{N}$. We consider tensors as vectors over product index sets. For this purpose, we introduce the d -fold product index set

$$\mathcal{I} := \mathcal{I}_1 \times \dots \times \mathcal{I}_d, \quad \mathcal{I}_\mu := \{1, \dots, n_\mu\}, \quad \mu \in \{1, \dots, d\}.$$

Definition 2 (Elementary tensor, order). A tensor $X \in \mathbb{R}^{\mathcal{I}}$ is called an *elementary tensor*, if it can be represented as the tensor product of vectors $x_\mu \in \mathbb{R}^{n_\mu}$, $\mu \in \{1, \dots, d\}$, i.e.

$$X = \bigotimes_{\mu=1}^d x_\mu. \quad (5)$$

The entries of X are given by

$$X_{(i_1, \dots, i_d)} = \prod_{\mu=1}^d (x_\mu)_{i_\mu}, \quad i_\mu \in \mathcal{I}_\mu.$$

Non-zero tensors of the form (5) are also called *rank 1* tensors. The integer d is the *order* or *dimension* of the tensor X .

Definition 3 (Tensor rank, (CP) format). The *rank* r of a tensor $X \in \mathbb{R}^{\mathcal{I}}$ is the minimal number $r \in \mathbb{N}_0$ such that there exist elementary tensors $X_1, \dots, X_r$ with

$$X = X_1 + \dots + X_r = \sum_{j=1}^r \bigotimes_{\mu=1}^d x_{j,\mu}, \quad x_{j,\mu} \in \mathbb{R}^{n_\mu}. \quad (6)$$

A tensor of the form (6) is said to be given in (CP) format. In the literature, the representation (6) is sometimes referred to as r -term representation, *PARAFAC* (parallel factors), *CANDECOMP* (canonical decomposition), or *Kronecker* format.

Definition 4 (Tucker rank, Tucker format). The *Tucker rank* of a tensor $X \in \mathbb{R}^{\mathcal{I}}$ is the tuple $(k_1, \dots, k_d)$ with minimal entries $k_\mu \in \mathbb{N}_0$ such that there exist orthonormal vectors $u_{j,\mu} \in \mathbb{R}^{n_\mu}$ and a so-called *core tensor* $C \in \mathbb{R}^{k_1 \times \dots \times k_d}$ with

$$X = \sum_{j_1=1}^{k_1} \dots \sum_{j_d=1}^{k_d} C_{(j_1, \dots, j_d)} \bigotimes_{\mu=1}^d u_{j_\mu, \mu}, \quad \langle u_{i,\mu}, u_{j,\mu} \rangle = \delta_{i,j}. \quad (7)$$

The representation of the form (7) is called the *Tucker format*.

A hierarchical representation of tensors has been introduced by Hackbusch and Kühn [7] and has been further analysed by Grasedyck [6]. The so-called *hierarchical Tucker format* is based on two major ingredients:

1. a hierarchy among the modes $1, \dots, d$ of a tensor corresponding to a hierarchy of subspaces,
2. a rank bound on certain matricisations of a tensor corresponding to the dimension of the subspaces.

For the definition of the hierarchical Tucker format, we will adopt the notation of [6].

Definition 5 (Matricisation). For a tensor $A \in \mathbb{R}^{\mathcal{I}}$, a collection $t \subset \{1, \dots, d\}$ of dimension indices and the complement $s := \{1, \dots, d\} \setminus t$ the matricisation

$$A^{(t)} \in \mathbb{R}^{\mathcal{I}_t \times \mathcal{I}_s}, \quad \mathcal{I}_t := \bigtimes_{\mu \in t} \mathcal{I}_\mu, \quad \mathcal{I}_s := \bigtimes_{\mu \in s} \mathcal{I}_\mu,$$

is defined by its entries

$$\left(A^{(t)}\right)_{(i_\mu)_{\mu \in t}, (i_\mu)_{\mu \in s}} := A_{(i_1, \dots, i_d)}.$$

Based on the matricisation of a tensor A with respect to several sets $t \subset \{1, \dots, d\}$ one can define the hierarchical rank and the hierarchical Tucker format. In order to be able to perform efficient arithmetics, we require the sets t to be organised in a tree.

Definition 6 (Dimension tree). A *dimension tree* $T_{\mathcal{I}}$ for dimension $d \in \mathbb{N}$ is a tree with root $\text{Root}(T_{\mathcal{I}}) = \{1, \dots, d\}$ and depth p such that each node $t \in T_{\mathcal{I}}$ is either

1. a leaf and singleton $t = \{\mu\}$ or
2. the union of two disjoint successors $S(t) = \{t_1, t_2\}$:

$$t = t_1 \dot{\cup} t_2.$$

The set of leaves of the tree is denoted by $\mathcal{L}(T_{\mathcal{I}})$.

Definition 7 (Hierarchical rank, $\mathcal{H}$ -Tucker). Let $T_{\mathcal{I}}$ be a dimension tree. The *hierarchical rank* $(k_t)_{t \in T_{\mathcal{I}}}$ of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ is defined by

$$\forall t \in T_{\mathcal{I}}: \quad k_t := \text{rank}(A^{(t)}).$$

The set of all tensors of hierarchical rank (node-wise) at most $(k_t)_{t \in T_{\mathcal{I}}}$, the so-called $\mathcal{H}$ -Tucker tensors, is denoted by

$$\mathcal{H}\text{-Tucker}((k_t)_{t \in T_{\mathcal{I}}}) := \{A \in \mathbb{R}^{\mathcal{I}} \mid \forall t \in T_{\mathcal{I}}: \text{rank}(A^{(t)}) \leq k_t\}.$$

Lemma 8 ($\mathcal{H}$ -Tucker format, [6]). Let $T_{\mathcal{I}}$ be a dimension tree and let $A \in \mathcal{H}\text{-Tucker}((k_t)_{t \in T_{\mathcal{I}}})$. Then A can be represented by transfer tensors $(B_t)_{t \in T_{\mathcal{I}} \setminus \mathcal{L}(T_{\mathcal{I}})}$ (for interior nodes) and mode frames $(U_t)_{t \in \mathcal{L}(T_{\mathcal{I}})}$ (for leaves), where $B_t \in \mathbb{R}^{k_t \times k_{t_1} \times k_{t_2}}$ for $S(t) = \{t_1, t_2\}$ and $U_t \in \mathbb{R}^{\mathcal{I}_t \times k_t}$.

The complete representation of a tensor $A \in \mathcal{H}\text{-Tucker}((k_t)_{t \in T_{\mathcal{I}}})$ is then given for all $t \in T_{\mathcal{I}} \setminus \mathcal{L}(T_{\mathcal{I}})$ with sons $S(t) = \{t_1, t_2\}$ by the recursive relation

$$(U_t)_\ell = \sum_{j_1=1}^{k_{t_1}} \sum_{j_2=1}^{k_{t_2}} (B_t)_{\ell, j_1, j_2} (U_{t_1})_{j_1} \otimes (U_{t_2})_{j_2}, \quad \ell = 1, \dots, k_t, \quad (8)$$

with $A = U_{\{1, \dots, d\}}$.

3 Approximation Scheme

Assume that we have found a parametrisation of $\sigma, \tau \subset \Gamma$ such that

$$f(a) := \int_{\sigma(a)} \int_{\tau(a)} \kappa(\|x - y\|) dy dx$$

is analytic in a box $Q := [s_1, t_1] \times \dots \times [s_d, t_d]$. How to find such a parametrisation will be discussed in Section 5. Since f is analytic in Q , we may approximate it by polynomials. The m -th order multivariate polynomial interpolation reads

$$f_m(a) = \sum_{\nu_1=0}^m \dots \sum_{\nu_d=0}^m f(\xi_{1,\nu_1}, \dots, \xi_{d,\nu_d}) \mathcal{L}_{1,\nu_1}(a_1) \dots \mathcal{L}_{d,\nu_d}(a_d) \quad (9)$$

where $\{\xi_{\mu,\nu}\}_{\mu=1,\dots,d, \nu=0,\dots,m}$ are interpolation points with $s_\mu \leq \xi_{\mu,\nu} \leq t_\mu$, $\mu = 1, \dots, d$, $\nu = 0, \dots, m$ and $\mathcal{L}_{\mu,\nu}$ are corresponding Lagrange polynomials. The values of f at the interpolation points define a d -dimensional tensor $A \in \mathbb{R}^{\mathcal{I}}$ with entries

$$A_{(\nu_1, \dots, \nu_d)} := f(\xi_{1,\nu_1}, \dots, \xi_{d,\nu_d}), \quad \nu_\mu = 0, \dots, m, \mu = 1, \dots, d,$$

where $\mathcal{I} = \mathcal{I}_1 \times \dots \times \mathcal{I}_d$, $\mathcal{I}_1 = \dots = \mathcal{I}_d = \{0, \dots, m\}$. The tensor A contains $(m+1)^d$ entries which may require too much storage and computing time for large d . We therefore want to approximate A by a tensor in a data sparse representation that allows us to evaluate f_m efficiently.

At this stage, it is in principle possible to choose any data sparse tensor format for the approximation of A . Here, we want to concentrate on the approximation by tensors given either in the (CP) or in the $\mathcal{H}$ -Tucker format because of the following considerations:

1. A tensor in (CP) format will allow us to evaluate interpolants in $\mathcal{O}(dr(m+1))$ operations but it is not easy to find such a tensor with low rank r for high accuracies.
2. A tensor in $\mathcal{H}$ -Tucker format will allow us to evaluate interpolants in $\mathcal{O}(dk^3 + dk(m+1))$ operations and there are reliable truncation procedures available to achieve high accuracies with still moderate $k := \max_{t \in T_{\mathcal{I}}} k_t$.

It is therefore reasonable to consider the approximation problem in both formats separately. This will be done in detail in Section 4. Let us now assume that A can be approximated by a tensor $\tilde{A} \in \mathbb{R}^{\mathcal{I}}$ given in (CP) format

$$\tilde{A} = \sum_{j=1}^r u_{j,1} \otimes \dots \otimes u_{j,d}, \quad u_{j,\mu} \in \mathbb{R}^{\mathcal{I}_\mu}, \quad j = 1, \dots, r, \mu = 1, \dots, d.$$

Now f_m from (9) may be approximated by

$$\begin{aligned} f_{m,r}(a) &:= \sum_{\nu_1=0}^m \dots \sum_{\nu_d=0}^m \tilde{A}_{(\nu_1, \dots, \nu_d)} \mathcal{L}_{1,\nu_1}(a_1) \dots \mathcal{L}_{d,\nu_d}(a_d) \\ &= \sum_{j=1}^r \left(\sum_{\nu_1=0}^m (u_{j,1})_{\nu_1} \mathcal{L}_{1,\nu_1} \right) (a_1) \dots \left(\sum_{\nu_d=0}^m (u_{j,d})_{\nu_d} \mathcal{L}_{d,\nu_d} \right) (a_d) \\ &= \sum_{j=1}^r \hat{u}_{j,1}(a_1) \dots \hat{u}_{j,d}(a_d) \end{aligned} \quad (10)$$

where

$$\hat{u}_{j,\mu} := \sum_{\nu=0}^m (u_{j,\mu})_\nu \mathcal{L}_{\mu,\nu}, \quad j = 1, \dots, r, \mu = 1, \dots, d.$$

This means that the approximation of the tensor A by a tensor $\tilde{A}$ given in (CP) format yields an approximation of the function f_m by a function $f_{m,r}$ which is the sum of tensor products of functions. Obviously, the functions $\hat{u}_{j,\mu}$ are interpolants of the values $(u_{j,\mu})_{\nu_\mu}$ in the interpolation points ξ_{μ,ν_μ} . Therefore they can be evaluated efficiently by standard techniques like Horner's scheme applied to Newton's divided differences. Once the scheme is set up, only $\mathcal{O}(dr(m+1))$ operations are required to evaluate $f_{m,r}$ for any given parameter vector a .

Equivalently, we may assume that A can be approximated by a tensor $A_{\mathcal{H}} \in \mathbb{R}^{\mathcal{I}}$ in $\mathcal{H}$ -Tucker format represented by transfer tensors $(B_t)_{t \in T_{\mathcal{I}} \setminus \mathcal{L}(T_{\mathcal{I}})}$ and mode frames $(U_t)_{t \in \mathcal{L}(T_{\mathcal{I}})}$. Then the interpolation has only to be performed in the leaves of the tensor. To see this, define for all leaves $t = \{\mu\} \in \mathcal{L}(T_{\mathcal{I}})$

$$(\hat{U}_\mu)_j := \sum_{\nu=0}^m (U_\mu)_{\nu,j} \mathcal{L}_{\mu,\nu}, \quad j = 1, \dots, k_\mu.$$

For all inner nodes $t \in T_{\mathcal{I}} \setminus \mathcal{L}(T_{\mathcal{I}})$ with sons $S(t) = \{t_1, t_2\}$ we define

$$(\hat{U}_t)_\ell ((a_\mu)_{\mu \in t}) := \sum_{j_1=1}^{k_{t_1}} \sum_{j_2=1}^{k_{t_2}} (B_t)_{\ell, j_1, j_2} (\hat{U}_{t_1})_{j_1} ((a_\mu)_{\mu \in t_1}) (\hat{U}_{t_2})_{j_2} ((a_\mu)_{\mu \in t_2}), \quad \ell = 1, \dots, k_t.$$

The final approximation is then given by

$$f_{m,\mathcal{H}} := \hat{U}_{\{1,\dots,d\}}.$$

Note that this construction is analogous to (8). The evaluation of the function $f_{m,\mathcal{H}}$ for a given parameter vector a in the leaves by Horner's scheme requires $\mathcal{O}((m+1) \sum_{\mu=1}^d k_\mu)$ operations. Due to the summations for the interior nodes, the overall complexity sums up to $\mathcal{O}(dk^3 + dk(m+1))$ where $k := \max_{t \in T_{\mathcal{I}}} k_t$.

4 Tensor Approximation

Let $A \in \mathbb{R}^{\mathcal{I}}$ and $\varepsilon > 0$. The aim of this section is to find a tensor $\tilde{A} \in \mathbb{R}^{\mathcal{I}}$ in a data sparse representation such that

$$\|A - \tilde{A}\| \leq \varepsilon. \quad (11)$$

The entries of A are given by the values of f at the interpolation points. For moderate dimensions, say $d \leq 6$, it is still possible to calculate and store all entries of A since typically $m \leq 20$. This full representation of A can then be used to find an approximation $\tilde{A}$ either given in (CP) or in $\mathcal{H}$ -Tucker format. For high dimensions, $d > 6$, it is no longer possible to store all entries of A . We therefore have to find an approximation of A *without* knowing all of its entries.

Here, we concentrate on the approximation in $\mathcal{H}$ -Tucker format for which a stable truncation procedure [6] is available and a heuristic approach has been developed to construct a tensor in a black box fashion [1]. We furthermore propose to use a variant of an ALS algorithm to approximate in the (CP) format with an input tensor given in $\mathcal{H}$ -Tucker format. This results in a still manageable complexity even in higher dimensions.

4.1 Approximation from Full Representation to $\mathcal{H}$ -Tucker

The approximation of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ to a tensor $A_{\mathcal{H}} \in \mathcal{H}$ -Tucker has been investigated by [7], [6]. Assume that for $\varepsilon > 0$ there exists a best approximation $A_{\mathcal{H}}^{\text{best}} \in \mathcal{H}$ -Tucker of hierarchical rank $(k_t)_{t \in T_{\mathcal{I}}}$ with the property

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

Then Theorem 24 of [6] shows that by means of a hierarchical singular value decomposition it is possible to find an approximation $A_{\mathcal{H}} \in \mathcal{H}$ -Tucker of the best approximation with hierarchical rank $(k_t)_{t \in T_{\mathcal{I}}}$ and the property

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

The complexity for the hierarchical singular value decomposition is bounded by

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

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

4.2 Black Box Approximation in $\mathcal{H}$ -Tucker

In [1] we presented a heuristic algorithm for the approximation of a tensor $A \in \mathbb{R}^{\mathcal{I}}$ by $A_{\mathcal{H}} \in \mathcal{H}$ -Tucker which is given by the evaluation of a function

$$A_{(\nu_1, \dots, \nu_d)} := f(\xi_{1, \nu_1}, \dots, \xi_{d, \nu_d}), \quad \nu_{\mu} = 0, \dots, m, \mu = 1, \dots, d.$$

The algorithm is based on the construction of low rank approximations of the matricisations of A by inspecting only very few tensor entries. The number of function evaluations required is bounded by

$$\mathcal{O}(dk^3 + d \log(d)k^2(m+1)).$$

The algorithm is rank-adaptive in the sense that it finds the necessary hierarchical rank $(k_t)_{t \in T_{\mathcal{I}}}$ for the representation of $A_{\mathcal{H}}$ automatically to guarantee an accuracy of

$$\|A - A_{\mathcal{H}}\|_{\infty} \leq \varepsilon.$$

The overall complexity for the setup of $A_{\mathcal{H}}$ is bounded by

$$\mathcal{O}(dk^4 + d \log(d)k^2(m+1)).$$

4.3 Approximation from $\mathcal{H}$ -Tucker to (CP) format

Assume that we have found a tensor $A_{\mathcal{H}} \in \mathcal{H}$ -Tucker that approximates A up to a given accuracy ε . We now want to find a tensor $\tilde{A} \in \mathbb{R}^{\mathcal{I}}$ in (CP) format with fixed representation rank r that approximates $A_{\mathcal{H}}$. A straightforward approach is to apply an alternating least squares algorithm (ALS). For a given approximation $\tilde{A} = \sum_{j=1}^r \bigotimes_{\mu=1}^d u_{j,\mu}$, $u_{j,\mu} \in \mathbb{R}^{\mathcal{I}_{\mu}}$, we may fix all $u_{j,\mu}$ except for $\mu = \nu$. Then the problem of minimising $\|A_{\mathcal{H}} - \tilde{A}\|^2$ reduces to a linear least squares problem for the unknown values $u_{j,\nu}$. Since $A_{\mathcal{H}}$ is already given in a data sparse representation, the computational effort for one ALS iteration can be dramatically reduced compared to the full representation. To see this, let us define $u_j^{(\nu)} := \bigotimes_{\mu \in [\nu]} u_{j,\mu}$ for $j = 1, \dots, r$, where $[\nu] := \{1, \dots, d\} \setminus \{\nu\}$. Moreover, let $\mathcal{I}_{[\nu]} := \times_{\mu \in [\nu]} \mathcal{I}_{\mu}$ and define $y_i := (A_{\mathcal{H}, i, \ell})_{\ell \in \mathcal{I}_{[\nu]}}$ for all $i \in \mathcal{I}_{\nu}$. If the $u_j^{(\nu)}$ are linearly independent, a minimiser of the least-squares problem may be found by the solution of the associated normal equations $Vx_i = b_i$ for all $i \in \mathcal{I}_{\nu}$ where

$$\begin{aligned} V_{j,\ell} &= \langle u_j^{(\nu)}, u_{\ell}^{(\nu)} \rangle, & j &= 1, \dots, r, \ell = 1, \dots, r, \\ (b_i)_j &= \langle u_j^{(\nu)}, y_i \rangle, & j &= 1, \dots, r, \\ (x_i)_j &= (u_{j,\nu})_i, & j &= 1, \dots, r. \end{aligned}$$

The computational load is dominated by the calculation of the vector b_i . The cost for the evaluation of one entry of b_i may be bounded by the cost for a scalar product between the tensor $A_{\mathcal{H}}$ and an elementary tensor. To verify this, let $e_{i_{\nu}} \in \mathbb{R}^{\mathcal{I}_{\nu}}$ be given by $(e_{i_{\nu}})_{i_{\mu}} = \delta_{i_{\nu}, i_{\mu}}$ and define $v \in \mathbb{R}^{\mathcal{I}}$ by

$$v := \bigotimes_{\mu=1}^{\nu-1} u_{j,\mu} \otimes e_{i_{\nu}} \otimes \bigotimes_{\mu=\nu+1}^d u_{j,\mu}.$$

Now it is easy to see that

$$\begin{aligned}
(b_{i_\nu})_j &= \left\langle u_j^{(\nu)}, y_{i_\nu} \right\rangle = \sum_{(i_\mu)_{\mu \in [\nu]} \in \mathcal{I}_{[\nu]}} A_{\mathcal{H}, (i_1, \dots, i_d)} \prod_{\mu=1, \mu \neq \nu}^d (u_{j, \mu})_{i_\mu} \\
&= \sum_{(i_\mu)_{\mu \in \{1, \dots, d\}} \in \mathcal{I}} A_{\mathcal{H}, (i_1, \dots, i_d)} \prod_{\mu=1}^d (v_{j, \mu})_{i_\mu} = \langle v, A_{\mathcal{H}} \rangle.
\end{aligned}$$

A scalar product between an elementary tensor and a tensor in $\mathcal{H}$ -Tucker can be performed in $\mathcal{O}(dk^3 + dk(m+1))$ operations, where $k := \max_{t \in T_{\mathcal{I}}} k_t$. Hence, all entries of the vectors b_i may be calculated in $\mathcal{O}(drk^3(m+1) + drk(m+1)^2)$. The computation of the entries of V requires at most $\mathcal{O}(dr^2(m+1))$ operations, whereas the solution of the normal equations can be done in at most $\mathcal{O}(r^3)$. Therefore the computational complexity for one full ALS cycle is bounded by

$$\mathcal{O}(d^2rk^3(m+1) + d^2rk(m+1)^2 + d^2r^2(m+1) + dr^3).$$

Remark 9. Instead of approximating the complete $\mathcal{H}$ -Tucker tensor $A_{\mathcal{H}}$, one can also restrict to approximating the structured core tensor represented by the transfer tensors $(B_t)_{t \in T_{\mathcal{I}} \setminus \mathcal{L}(T_{\mathcal{I}})}$. The core tensor may be interpreted as an $\mathcal{H}$ -Tucker tensor with mode sizes k_μ instead of $(m+1)$. The complexity for one ALS cycle applied to the structured core tensor is then bounded by

$$\mathcal{O}(d^2rk^4 + d^2r^2k + dr^3).$$

5 Parametrisation of Boundary Elements

Let $\sigma, \tau \subset \Gamma$ be boundary elements. The aim of this section is to find parametrisations $\sigma(a), \tau(a)$ such that the function f given by (2) is analytic for all $a \in Q \subset \mathbb{R}^d$. Since, in practice, it is the most relevant setting, we will restrict our attention to the case when Γ is the boundary of a three-dimensional domain and σ, τ are flat triangles of a conforming triangulation of Γ .

First note that if for two pairs of triangles (σ, τ) and (σ', τ') there exists an orthogonal matrix $T \in \mathbb{R}^{3 \times 3}$ and a displacement vector $c \in \mathbb{R}^3$ such that $\sigma' = \phi(\sigma)$ and $\tau' = \phi(\tau)$, where $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the bijective linear map given by

$$\phi(x) = Tx + c, \quad x \in \mathbb{R}^3,$$

then we have

$$\int_{\sigma} \int_{\tau} \kappa(\|x - y\|) dy dx = \int_{\sigma'} \int_{\tau'} \kappa(\|x - y\|) dy dx. \quad (12)$$

W.l.o.g. we may therefore assume that the first triangle σ lies in the x - y -plane and is defined by the corners $P_1 = (0, 0, 0)^\top$, $P_2 = (p_{21}, 0, 0)^\top$, $P_3 = (p_{31}, p_{32}, 0)^\top$ and the second triangle τ is given by the corners $Q_1 = (q_{11}, q_{12}, q_{13})^\top$, $Q_2 = (q_{21}, q_{22}, q_{23})^\top$, $Q_3 = (q_{31}, q_{32}, q_{33})^\top$. For a conforming triangulation, the following four situations may occur:

- the case of a common face, where $\sigma = \tau$ which leads to a parameter space of dimension $d = 3$,
- the case of a common edge, where $\bar{\sigma} \cap \bar{\tau} = [0, p_{21}]$ which leads to a parameter space of dimension $d = 6$,
- the case of a common vertex, where $\bar{\sigma} \cap \bar{\tau} = \{(0, 0, 0)^\top\}$ which leads to a parameter space of dimension $d = 9$, and finally
- the case of disjoint triangles, where $\bar{\sigma} \cap \bar{\tau} = \emptyset$ which leads to a parameter space of dimension $d = 12$.

In some cases, the parameter space can be further reduced. To see this, first note the following definition.

Definition 10 (Homogeneity). The kernel function $\kappa : \mathbb{R} \rightarrow \mathbb{R}$ is called *homogeneous* of *degree* α if for all $\lambda > 0$,

$$\kappa(\lambda x) = \lambda^\alpha \kappa(x) \quad \text{for all } x \in \mathbb{R}.$$

For triangles σ, τ defined as above, let $\Lambda := \text{diag}\{p_{21}, p_{21}, p_{21}\}$ be a scaling matrix and define

$$\sigma' := \{\Lambda^{-1}x \mid x \in \sigma\}, \quad \tau' := \{\Lambda^{-1}x \mid x \in \tau\}.$$

For a homogeneous kernel function κ of degree α , the integral (12) is determined by

$$\int_{\sigma} \int_{\tau} \kappa(\|x - y\|) dy dx = |\det \Lambda|^2 \int_{\sigma'} \int_{\tau'} \kappa(\|\Lambda(x - y)\|) dy dx = |p_{21}|^{6+\alpha} \int_{\sigma'} \int_{\tau'} \kappa(\|x - y\|) dy dx.$$

In this particular situation, it hence suffices to consider triangles with $p_{21} = 1$ which reduces the parameter dimension for all cases by one. In the following we will propose parameter tuples $a = (a_1, \dots, a_d)$ such that the function $f(a)$ given by (2) is analytic in a box $Q = [s_1, t_1] \times \dots \times [s_d, t_d]$. We will restrict our attention to the first three cases, where the integrand contains a singularity but it is also possible to apply our technique to integrals over disjoint triangles.

5.1 The Case of a Common Face

If $\sigma = \tau$, we have $Q_1 = P_1, Q_2 = P_2, Q_3 = P_3$, and the dimension of the parameter space reduces to $d = 3$. A straightforward way to describe the remaining points by a parameter tuple $a \in \mathbb{R}^3$ is to set $P_1 := (0, 0, 0)^\top, P_2 := (a_1, 0, 0)^\top, P_3 := (a_1 a_2, a_1 a_3, 0)^\top$. The very simple geometrical setting is depicted in Figure 1.

Figure 1: triangles with a common face

5.2 The Case of a Common Edge

If $\bar{\sigma} \cap \bar{\tau} = [0, p_{21}]$, we have $Q_1 = P_1, Q_2 = P_2$, and the parameter space is of dimension $d = 6$. As in the previous case, we will set $P_1 := (0, 0, 0)^\top, P_2 := (a_1, 0, 0)^\top, P_3 := (a_1 a_2, a_1 a_3, 0)^\top$. It remains to parametrise the point Q_3 . As it is geometrically meaningful, we suggest to incorporate the angle $\phi := \angle(E_\sigma, E_\tau)$ into the parametrisation, where E_σ, E_τ are the planes for which $\sigma \subset E_\sigma, \tau \subset E_\tau$. Let us therefore introduce a rotation matrix

$$R_\phi := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix}$$

and an auxiliary point $\hat{Q}_3 := (a_1 a_4, a_1 a_5, 0)^\top$. Then we simply set $Q_3 := R_\phi \hat{Q}_3$ and define $a_6 := \phi$. The geometrical setting is depicted in Figure 2. In order to avoid pathological cases, we require that $a_1, a_3, a_5 \geq 0$ and $\phi \in (0, \pi]$.

Figure 2: triangles with a common edge

5.3 The Case of a Common Vertex

If $\bar{\sigma} \cap \bar{\tau} = \{(0, 0, 0)^\top\}$, we have $Q_1 = P_1$, and the parameter space is of dimension $d = 9$. To simplify the presentation, we will just consider cases in which the two triangles lie almost in the same plane which is true for all parts of the boundary where the curvature is still moderate. In fact, this corresponds to a setting very often encountered in practice as many parts of the surface are locally flat. As in the previous case, we will set $P_1 := (0, 0, 0)^\top$ and $P_2 := (a_1, 0, 0)^\top$. The point P_3 may be parametrised by polar coordinates, $P_3 := a_1 a_2 (\cos \alpha, \sin \alpha, 0)^\top$, which will enable us later to restrict the possible range of α . In order to parametrise the remaining two points Q_2 and Q_3 , we will first define two auxiliary points by $\hat{Q}_2 := a_1 a_3 (1, 0, \sin \phi)^\top$ and $\hat{Q}_3 := a_1 a_3 a_4 (\cos \beta, -\sin \beta, \sin \psi)^\top$. We now introduce a rotation matrix

$$R_\delta := \begin{pmatrix} \cos \delta & \sin \delta & 0 \\ -\sin \delta & \cos \delta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and set $Q_2 := R_\delta \hat{Q}_2$, $Q_3 := R_\delta \hat{Q}_3$. The geometrical setting is depicted in Figure 3.

Figure 3: triangles with a common vertex

Note that for $\phi, \psi \in [-\varepsilon, \varepsilon]$ with small $\varepsilon > 0$, the two triangles lie almost in the same plane. Moreover, let $\Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be the orthogonal projection onto the x_1 - x_2 -plane. Then we have $\beta = \angle(\Pi(Q_2) - P_1, \Pi(Q_3) - P_1)$. To avoid pathological cases, it is hence reasonable to require that $\delta > 0$ and that $\alpha/2 + \beta/2 + \delta \leq \pi$. To guarantee this two conditions, we introduce two auxiliary variables $\lambda_\alpha \in [0, 1]$ and $\lambda_\beta \in [0, 1]$ and let $0 < \delta_{\min} < \delta < \delta_{\max} < \pi$, $\alpha := \alpha_{\min} + \lambda_\alpha(\alpha_{\max} - \alpha_{\min})$,

$\beta := \beta_{\min} + \lambda_\beta(\beta_{\max} - \beta_{\min})$ where $\alpha_{\min}, \beta_{\min} > 0$, $\alpha_{\max} = \beta_{\max} := \pi - \delta$. This prevents the triangles from coming too close. It is therefore possible to choose the rest of the parameters by $a_5 := \phi$, $a_6 := \psi$, $a_7 := \delta$, $a_8 := \lambda_\alpha$, $a_9 := \lambda_\beta$ such that $a \in Q$ for a box $Q \subset \mathbb{R}^9$.

5.4 Analyticity of the function f

The analysis for functions f of the type (2) with singular kernel functions κ has been carried out in detail in [2]. Here, we summarise the results for the case of a weakly singular kernel function κ . Given an initial state $a_0 \in Q \subset \mathbb{R}^d$, the dependence of the domains σ and τ on a parameter vector $a \in Q$ can be expressed by a map $\Phi : \mathbb{R}^\ell \times Q \rightarrow \mathbb{R}^\ell$ defined by

$$\begin{aligned}\Phi(x, a) &:= x_a \in \sigma(a) \quad \text{for all } x \in \sigma(a_0), a \in Q, \\ \Phi(y, a) &:= y_a \in \tau(a) \quad \text{for all } y \in \tau(a_0), a \in Q,\end{aligned}$$

where

$$\Phi(x, a_0) = x \quad \text{for all } x \in \sigma(a_0), \quad \Phi(y, a_0) = y \quad \text{for all } y \in \tau(a_0).$$

We assume that for all $a \in Q$ the map $\Phi(\cdot, a) : \mathbb{R}^\ell \rightarrow \mathbb{R}^\ell$ is a diffeomorphism between $\sigma(a_0) \cup \tau(a_0)$ and $\sigma(a) \cup \tau(a)$. Moreover, we assume that for all $x \in \sigma(a_0) \cup \tau(a_0)$ the map $\Phi(x, \cdot) : Q \rightarrow \mathbb{R}^\ell$ is smooth with respect to the parameter vector $a \in Q$. Note that the parametrisations of triangles introduced in the sections above fulfil the smoothness requirements on Φ . This is due to the fact that all triangles are parametrised by their corners which smoothly depend on $a \in Q$. Moreover, any $x \in \sigma(a_0)$ can be associated with its barycentric coordinates which again define a point $x_a \in \sigma(a)$. This leads to a natural diffeomorphism $\Phi(\cdot, a)$ for all $a \in Q$.

To analyse the sensitivity of the function f with respect to changes in the parameter vector $a \in Q$, we fix all components of a except for a fixed direction $\mu \in \{1, \dots, d\}$ and consider variations in a_μ . W.l.o.g. we may assume that the initial state is attained at $a_\mu = 0$. The *transformation velocity* $\theta : \mathbb{R}^\ell \rightarrow \mathbb{R}^\ell$ at $a_\mu = 0$ is then defined by

$$\theta(y) := \left. \frac{\partial}{\partial a_\mu} \Phi(y, a) \right|_{a_\mu=0} \quad \text{for all } y \in \sigma(a_0) \cup \tau(a_0).$$

The *material derivative* $\dot{g}(y)$ of a function $g : \mathbb{R}^\ell \times Q \rightarrow \mathbb{R}$ at $a_\mu = 0$ is defined as

$$\dot{g}(y) := \left. \frac{\partial}{\partial a_\mu} g(\Phi(y, a), a) \right|_{a_\mu=0} = \left. \frac{\partial}{\partial a_\mu} g(y_a, a) \right|_{a_\mu=0} + \nabla_y g(y_a, a) \cdot \theta(y).$$

The material derivative of an *integral* over the parametrised domain $\tau(a)$ can be obtained by the formula (cf. [2])

$$\left. \frac{\partial}{\partial a_\mu} \int_{\tau(a)} g \, ds \right|_{a_\mu=0} = \int_{\tau(a_0)} (\dot{g} + g \operatorname{div} \theta) \, ds. \quad (13)$$

We now want to apply this result to our particular setting, where the integrand is given by the kernel function $\kappa = \kappa(x, y) = \kappa(x - y)$. First note that, since κ does not depend on a_μ and x and y are transformed by the same map Φ , the material derivative is given by

$$\dot{\kappa}(x - y) = \nabla \kappa(x - y) \cdot \left. \frac{\partial(x_a - y_a)}{\partial a_\mu} \right|_{a_\mu=0} = \nabla \kappa(x - y) \cdot (\theta(x) - \theta(y)).$$

The application of (13) with $g = \kappa$ yields

$$\left. \frac{\partial}{\partial a_\mu} \int_{\tau(a)} \kappa(x - y) \, ds_y \right|_{a_\mu=0} = \int_{\tau(a_0)} (\nabla \kappa(x - y) \cdot (\theta(x) - \theta(y)) + \kappa(x - y) \operatorname{div} \theta(y)) \, ds_y. \quad (14)$$

Note that due to the smoothness of Φ we have

$$\|\theta(x) - \theta(y)\| \leq C \|x - y\| \quad \text{for all } x \in \sigma(a_0), y \in \tau(a_0),$$

for some $C > 0$. Therefore, the integrand in (14) is again at most weakly singular and the integral on the right-hand side exists. To establish the existence of the first derivate of f , we first define for all $x \in \sigma(a)$

$$g_{\tau(a)}(x) := \int_{\tau(a)} \kappa(x - y) \, ds_y. \quad (15)$$

The application of (13) with $g = g_{\tau(a)}$ then yields

$$\frac{\partial f}{\partial a_\mu} = \frac{\partial}{\partial a_\mu} \int_{\sigma(a)} g_{\tau(a)}(x) \, ds_x = \int_{\sigma(a_0)} (\dot{g}_{\tau(a)} + g_{\tau(a)} \operatorname{div} \theta) \, ds_x. \quad (16)$$

The material derivative $\dot{g}_{\tau(a)}$ is given by (14) which leads to an integrand that is at most weakly singular. Hence the first derivate of f with respect to a_μ exists for all directions $\mu \in \{1, \dots, d\}$. From (16) it can readily be seen that higher derivatives of f involve only regular terms or expressions of the type (15). We therefore deduce that f is smooth with respect to the parameter vector $a \in \mathbb{R}^d$.

A similar but slightly more involved analysis can be performed for the case of strongly and hypersingular kernel functions κ , see [2].

6 Error Analysis

In this section we briefly analyse the error for the approximation of the function f by the scheme introduced in Section 3. In order to approximate the function f by interpolation, we need to be able to evaluate f at the interpolations points. Since the value of f corresponds to the evaluation of an integral, in most cases it will not be possible to evaluate f exactly. Instead, we have to rely on quadrature rules that give us an approximation of f up to a certain precision. We therefore have to deal with three different types of errors:

1. the interpolation error induced by substituting f by its interpolant f_m ,
2. the quadrature error by the approximate calculation of the entries of the tensor A ,
3. the tensor approximation error when substituting A by a tensor $\tilde{A}$ in (CP) or $\mathcal{H}$ -Tucker format.

To estimate the interpolation error, let Λ_m be a constant satisfying the stability estimate

$$\left\| \mathfrak{I}_m^{[-1,1]}[f] \right\|_{\infty, [-1,1]} \leq \Lambda_m \|f\|_{\infty, [-1,1]} \quad \text{for all } f \in C[-1, 1]. \quad (17)$$

Note that in the case of Chebyshev interpolation, one can show that

$$\Lambda_m := \frac{2}{\pi} \ln(m+1) + 1 \leq m+1$$

is a good choice for all $m \in \mathbb{N}$ which means that the stability constant grows very slowly depending on m . Since $\mathfrak{I}_m^{[-1,1]}$ is a projection onto the set $\mathcal{P}_m$ of m -th order polynomials, we have for a function $f \in C^{m+1}[-1, 1]$,

$$\left\| f - \mathfrak{I}_m^{[-1,1]}[f] \right\|_{\infty, [-1,1]} \leq \frac{2^{-m}}{(m+1)!} \left\| f^{(m+1)} \right\|_{\infty, [-1,1]}. \quad (18)$$

We now can formulate the following theorem.

Theorem 11 ([3] Theorem 3.10). *Let $Q = [s_1, t_1] \times \dots \times [s_d, t_d]$ and $m \in \mathbb{N}$. Let $\mathfrak{I}_m^{[-1,1]}$ satisfy the stability estimate (17) and the error estimate (18). Let $\mathfrak{I}_m^Q$ be the m -th order tensor-product interpolation operator defined by (4). Then we have*

$$\begin{aligned} \left\| \mathfrak{I}_m^Q[f] \right\|_{\infty, Q} &\leq \Lambda_m^d \|f\|_{\infty, Q}, & f \in C(Q), \\ \left\| f - \mathfrak{I}_m^Q[f] \right\|_{\infty, Q} &\leq \frac{2d\Lambda_m^{d-1}}{(m+1)!} \left(\frac{\operatorname{diam}(Q)}{4} \right)^{m+1} \max_{\mu \in \{1, \dots, d\}} \left\{ \left\| \partial_\mu^{m+1} f \right\|_{\infty, Q} \right\}, & f \in C^{m+1}(Q). \end{aligned}$$

Due to the application of a quadrature rule, the entries of A are not determined exactly. Let $A^{\text{quad}} \in \mathbb{R}^{\mathcal{I}}$ denote the tensor that determines the entries of A by a quadrature rule and assume

$$\left\| A - A^{\text{quad}} \right\|_{\infty} \leq \varepsilon_{\text{quad}}$$

for some $\varepsilon_{\text{quad}} > 0$. The tensor approximation is then based on the tensor A^{quad} which leads to an additional error $\varepsilon_{\text{ten}} > 0$, i.e.

$$\left\| A^{\text{quad}} - \tilde{A} \right\|_{\infty} \leq \varepsilon_{\text{ten}}.$$

We therefore have

$$\left\| A - \tilde{A} \right\|_{\infty} \leq \varepsilon_{\text{quad}} + \varepsilon_{\text{ten}}.$$

Since $f_{m,r}$ from (10) was defined as the interpolation polynomial belonging to the perturbed values $\tilde{A}$, we have that

$$\|f_m - f_{m,r}\|_{\infty,Q} \leq \Lambda_m^d \left\| A - \tilde{A} \right\|_{\infty}.$$

The total error can now be estimated by

$$\begin{aligned} \|f - f_{m,r}\|_{\infty,Q} &= \|f - f_m + f_m - f_{m,r}\|_{\infty,Q} \\ &\leq \|f - \mathfrak{I}_m^Q[f]\|_{\infty,Q} + \Lambda_m^d \left\| A - \tilde{A} \right\|_{\infty} \\ &\leq \|f - \mathfrak{I}_m^Q[f]\|_{\infty,Q} + \Lambda_m^d (\varepsilon_{\text{quad}} + \varepsilon_{\text{ten}}). \end{aligned}$$

An analogous estimate holds for $f_{m,\mathcal{H}}$.

7 Numerical Examples

In this section, we want to verify if the parametrisation of boundary elements leads to acceptable numerical results. In particular we are interested in the following questions:

- How do the ranks of the approximations of the tensor A in the (CP) and in the $\mathcal{H}$ -Tucker format behave for increasing accuracies?
- Which interpolation orders are needed to reach a certain accuracy?
- How does the scheme perform compared to the standard evaluation of integrals?

In the following, we will analyse these three points for the case of the Laplace single layer potential and for the Helmholtz single layer potential. The (approximate) evaluation of the entries of the tensor A will be done by the method introduced in [4], [13] which relies on a transformation of the integrand which removes the singularity. The regularised integral can be computed numerically by a standard Gauss quadrature of sufficiently high order.

7.1 Laplace Single Layer Potential

The Laplace single layer potential is related to the kernel $\kappa(\|x - y\|) = 1/(4\pi \|x - y\|)$. A Galerkin discretisation with constant basis functions thus requires the evaluation of integrals of the form

$$f(a) := \int_{\sigma(a)} \int_{\tau(a)} \frac{1}{4\pi \|x - y\|} dy dx.$$

Since the kernel function is homogeneous of degree -1, the parameter dimension may be reduced by one as pointed out in Section 5.

Example 12 (Laplace: common face). Using the same notation as in Section 5.1, we may w.l.o.g set $a_1 := 1$. We now let $a_2 \in [0.5, 1]$ and $a_3 \in [0.5, 2]$. In this particular case, the (CP) format and the $\mathcal{H}$ -Tucker format are equivalent and may be determined by a simple SVD. For the evaluation of the entries of A by [13], we use Gauss quadrature of order 15 which is accurate up to $\approx 10^{-12}$. Our first numerical test concerns the required interpolation order m . Since we cannot test the interpolation quality on the whole box Q , we randomly choose $L := 10^3$ points $q_\ell \in Q$, $\ell = 1, \dots, L$, which we collect in a set $\tilde{Q} := \{q_\ell \mid 1 \leq \ell \leq L\}$. We then measure the relative errors

$$\begin{aligned} \text{err}_2 &:= \|f_{m,k} - f\|_{2,\tilde{Q}} / \|f\|_{2,\tilde{Q}}, & \|g\|_{2,\tilde{Q}} &:= \left(\sum_{\ell=1}^L |g(q_\ell)|^2 \right)^{1/2}, \\ \text{err}_0 &:= \|f_{m,k} - f\|_{\infty,\tilde{Q}} / \|f\|_{\infty,\tilde{Q}}, & \|g\|_{\infty,\tilde{Q}} &:= \max_{1 \leq \ell \leq L} |g(q_\ell)|. \end{aligned}$$

The rank r of the tensor approximation has been chosen adaptively to guarantee $\|A - \tilde{A}\|_2 / \|A\|_2 \leq \varepsilon_{\text{ten}} := 10^{-12}$. The relative interpolation error is summarised in Table 1. As predicted by the theoretical estimate, the interpolation error decreases exponentially with the interpolation order.

Secondly, we are interested in the behaviour of the rank r when we require a certain accuracy ε_{ten} for the approximation of the tensor A . In order to not losing the accuracy of the interpolation, we have to require that the tensor approximation error is at least as small as the interpolation error when using a highly accurate tensor approximation. We therefore choose the largest possible interpolation order m which still guarantees $\varepsilon_{\text{ten}} < \text{err}_0$ and look at the ranks r for the tensor approximation. As Table 1 illustrates, the ranks grow only mildly when the accuracy is increased.

In a third test, we compare the timings for the evaluation of 10^6 integrals by [13] using Gauss quadrature of order q with n_q quadrature points in time t_q to our new interpolation-based method. For the evaluation of an integral by interpolation two time measurements are relevant. First, we need to find the parameter vector $a \in \mathbb{R}^d$ from the triangle coordinates on the grid in time t_Φ . Second, we need to evaluate the function f at a in time t_r . The results are summarised in Table 2. Note that the order of the Gauss quadrature has been adapted to meet the accuracy ε . The speed-up $\rho := t_q / (t_r + t_\Phi)$ typically exceeds three orders of magnitude.

ε_{ten}	$m + 1$	r
1e-03	5	2
1e-04	7	2
1e-05	8	3
1e-06	10	3
1e-07	12	4
1e-08	14	4
1e-09	16	5
1e-10	18	6

Table 1: Laplace: common face. Left: interpolation error for $\varepsilon_{\text{ten}} = 10^{-12}$. Right: interpolation orders and ranks of the tensor approximation in (CP) format for different accuracies ε_{ten} .

Example 13 (Laplace: common edge). Using the notation of Section 5.2, we will restrict the parameter space to the values depicted in Table 3. The tensor approximation can be done either in $\mathcal{H}$ -Tucker format or in the (CP) format. We suggest to first approximate in $\mathcal{H}$ -Tucker from which we can obtain an approximation in (CP) format using an ALS method. The interpolation error is measured for an interpolation function based on a tensor approximated in $\mathcal{H}$ -Tucker format with an accuracy of $\varepsilon_{\text{ten}} = 10^{-12}$. For the evaluation of the entries of A by [13], we use

ε	q	n_q	t_q	$m+1$	t_Φ	t_r	ρ
1e-03	4	1152	38s	5	0.05s	0.06s	345
1e-04	5	2250	77s	7	0.05s	0.07s	642
1e-05	6	3888	134s	8	0.05s	0.14s	705
1e-06	8	9216	316s	10	0.05s	0.13s	1756
1e-07	9	13122	452s	12	0.05s	0.25s	1507
1e-08	10	18000	621s	14	0.05s	0.30s	1774
1e-09	11	23958	832s	16	0.05s	0.44s	1698
1e-10	13	39546	1467s	18	0.05s	0.53s	2529

Table 2: Laplace: common face. Timings for the evaluation of 10^6 integrals with accuracy ε .

	a_1	a_2	a_3	a_4	a_5	a_6
min	1.0	0.0	0.5	0.0	0.5	$\pi/2$
max	1.0	1.0	2.0	1.0	2.0	π

Table 3: Laplace: parameter space for the case of a common edge.

Gauss quadrature of order 15 which is accurate up to $\approx 10^{-12}$. The error and the necessary ranks for the tensor approximation are summarised in Table 4. Here, k_{eff} denotes the effective rank of a tensor in $\mathcal{H}$ -Tucker format which corresponds to a storage complexity of $\mathcal{O}((d-1)k_{\text{eff}}^3 + dk_{\text{eff}}(m+1))$. The timings for the evaluation by [13] using Gauss quadrature and for the evaluation by interpolation are compared in Table 5. Here, $t_{\mathcal{H}}$ and t_r denote the evaluation time for 10^6 integrals when A is approximated in $\mathcal{H}$ -Tucker format and (CP) format, respectively. The speed-up $\rho := t_q / (\min\{t_{\mathcal{H}}, t_r\} + t_\Phi)$ typically reaches two orders of magnitude.

ε_{ten}	$m+1$	k_{eff}	r
1e-03	4	2.7	5
1e-04	6	3.4	11
1e-05	8	4.8	20
1e-06	9	6.3	41
1e-07	11	8.1	77
1e-08	13	9.8	—
1e-09	15	12.0	—
1e-10	17	14.5	—

Table 4: Laplace: common edge. Left: interpolation error for $\varepsilon_{\text{ten}} = 10^{-12}$. Right: interpolations orders and ranks of the tensor approximation in $\mathcal{H}$ -Tucker format and in (CP) format for different accuracies ε_{ten} .

ε	q	n_q	t_q	$m+1$	t_Φ	$t_{\mathcal{H}}$	t_r	ρ
1e-03	4	960	33s	4	0.2s	1.08s	0.24s	75
1e-04	5	1875	63s	6	0.2s	1.44s	0.76s	65
1e-05	6	3240	111s	8	0.2s	2.28s	1.83s	55
1e-06	7	5145	176s	9	0.2s	3.65s	4.21s	46
1e-07	8	7680	263s	11	0.2s	6.04s	8.59s	42
1e-08	10	15000	512s	13	0.2s	9.58s	—	52
1e-09	11	19965	682s	15	0.2s	16.43s	—	41
1e-10	13	32955	1203s	17	0.2s	29.28s	—	41

Table 5: Laplace: common edge. Timings for the evaluation of 10^6 integrals with accuracy ε .

Example 14 (Laplace: common vertex). As discussed in Section 5.2, we just consider the case when the two triangles lie almost in the same plane. Using the notation of Section 5.2, we therefore require that $\phi, \psi \in [-\pi/16, \pi/16]$. To avoid unnatural configurations, we allow $\delta \in [\pi/5, 4\pi/5]$ and let $\alpha = \alpha_{\min} + \lambda_{\alpha}(\alpha_{\max} - \alpha_{\min})$, $\beta = \beta_{\min} + \lambda_{\beta}(\beta_{\max} - \beta_{\min})$, where $\alpha_{\min} = \beta_{\min} := \pi/5$, $\alpha_{\max} = \beta_{\max} := \pi - \delta$. We have summarised the parameter space in Table 6. As

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
min	1.0	0.7	0.7	0.7	$-\pi/16$	$-\pi/16$	$\pi/5$	0.0	0.0
max	1.0	1.4	1.4	1.4	$\pi/16$	$\pi/16$	$4\pi/5$	1.0	1.0

Table 6: Laplace: parameter space for the case of a common vertex.

some numerical experiments have shown, it is advantageous to subdivide the admissible interval for $\delta \in [\pi/5, 4\pi/5]$ into several smaller intervals in order to obtain sufficiently low tensor ranks. The resulting tensors can easily be considered separately as only one parameter is concerned. We demonstrate the growth of the ranks towards smaller values of δ by means of three examples. Let a subdivision of the interval $[\pi/5, 4\pi/5]$ be given by $D_i = [\pi/5 + i\pi/20, \pi/5 + (i+1)\pi/20]$, $i = 0, \dots, 11$. We now consider values of δ for $\delta_0 \in D_0$, $\delta_6 \in D_6$, $\delta_{11} \in D_{11}$.

The interpolation error is measured for an interpolation function based on a tensor approximated in $\mathcal{H}$ -Tucker-format which has been obtained in a black box fashion with an accuracy of $\varepsilon_{\text{ten}} = 10^{-8}$. For the evaluation of the entries of A by [13], we use Gauss quadrature of order 13 which is accurate up to $\approx 10^{-10}$. The interpolation error and the necessary ranks for the tensor approximation are summarised in Figure 4 and Table 7. From Table 7 one can clearly see that the tensor ranks and the interpolation order grow as δ gets smaller. This may be due to the fact that the effect of the singularity becomes stronger when the two triangles approach each other. The timings for the evaluation by [13] using Gauss quadrature and for the evaluation by interpolation are compared in Table 8. The speed-up $\rho := t_q / (\min\{t_{\mathcal{H}}, t_r\} + t_{\Phi})$ typically reaches one order of magnitude but may even be larger if the two triangles are not too close.

Figure 4: Laplace: common vertex. Interpolation error for $\varepsilon_{\text{ten}} = 10^{-8}$ and different ranges of δ .

7.2 Helmholtz Single Layer Potential

The Helmholtz single layer potential is related to the kernel $\kappa(\|x - y\|) = e^{ik\|x - y\|} / (4\pi \|x - y\|)$ where k denotes the wave number. For simplicity, we will just consider the real part $\text{Re } \kappa = \cos(k \|x - y\|) / (4\pi \|x - y\|)$. A Galerkin discretisation with constant basis functions thus requires the evaluation of integrals of the form

$$f(a) := \int_{\sigma(a)} \int_{\tau(a)} \frac{\cos(k \|x - y\|)}{4\pi \|x - y\|} dy dx.$$

Let us consider the high frequency case $k \sim 1/h$ where h is the meshwidth of the grid. If, e.g., $h \sim 1$, we may choose $k := 1/32$.

ε_{ten}	δ_0			δ_6			δ_{11}		
	$m+1$	k_{eff}	r	$m+1$	k_{eff}	r	$m+1$	k_{eff}	r
1e-03	4	3.1	9	3	2.2	6	3	1.7	5
1e-04	6	4.2	25	4	3.5	12	4	2.5	8
1e-05	7	6.4	69	5	4.4	30	5	3.3	16
1e-06	9	9.0	—	6	5.8	73	6	4.2	35
1e-07	11	12.3	—	7	7.7	—	7	5.0	—

Table 7: Laplace: common vertex. Interpolation orders and ranks of the tensor approximation in $\mathcal{H}$ -Tucker format and in (CP) format for different accuracies ε_{ten} and different ranges of δ .

ε	q	n_q	t_q	δ_0				δ_6			δ_{11}		
				t_Φ	$t_{\mathcal{H}}$	t_r	ρ	$t_{\mathcal{H}}$	t_r	ρ	$t_{\mathcal{H}}$	t_r	ρ
1e-03	3	162	3.60s	0.5s	1.92s	0.65s	3.1	1.37s	0.36s	4.2	1.23s	0.29s	4.6
1e-04	4	384	8.58s	0.5s	2.71s	2.39s	3.0	2.19s	0.85s	6.4	1.55s	0.57s	8.0
1e-05	6	1296	28.86s	0.5s	5.38s	7.58s	4.9	2.84s	2.52s	9.6	1.97s	1.34s	15.7
1e-06	7	2058	45.82s	0.5s	11.57s	—	3.8	4.38s	7.03s	9.4	2.74s	3.32s	14.1
1e-07	8	3072	68.31s	0.5s	24.92s	—	2.7	8.00s	—	8.0	3.37s	—	17.7

Table 8: Laplace: common vertex. Timings for the evaluation of 10^6 integrals with accuracy ε for different ranges of δ .

Example 15 (Helmholtz: common face). Since the Helmholtz kernel is no longer homogeneous, the parameter space is now 3-dimensional. We use the same parameter bounds as in Example 12 but let $a_1 \in [0.5, 2]$. The interpolation error and the necessary ranks for the tensor approximation are summarised in Table 9.

ε_{ten}	$m+1$	k_{eff}	r
1e-03	6	1.5	2
1e-04	8	1.6	2
1e-05	11	2.7	4
1e-06	13	2.7	5
1e-07	15	3.2	6
1e-08	17	3.8	8
1e-09	19	4.3	—

Table 9: Helmholtz: common face. Left: interpolation error for $\varepsilon_{\text{ten}} = 10^{-12}$. Right: interpolation orders and ranks of the tensor approximation in $\mathcal{H}$ -Tucker format and in (CP) format for different accuracies ε_{ten} .

Example 16 (Helmholtz: common edge). We use the same parameter space as in Table 3 but let $a_1 \in [0.5, 2.0]$. The interpolation error is measured for an interpolation function based on a tensor approximated in $\mathcal{H}$ -Tucker format which has been obtained in a black box fashion with an accuracy of $\varepsilon_{\text{ten}} = 10^{-8}$. The interpolation error and the necessary ranks for the tensor approximation are summarised in Table 10.

Example 17 (Helmholtz: common vertex). We use the same parameter space as in Table 6 but let $a_1 \in [0.5, 2.0]$. Again, the interpolation error is measured for an interpolation function based on a tensor approximated in $\mathcal{H}$ -Tucker format which has been obtained in a black box fashion

ε_{ten}	$m + 1$	k_{eff}	r
1e-03	5	3.2	11
1e-04	7	4.7	29
1e-05	9	7.2	75
1e-06	11	10.0	185
1e-07	13	13.0	–

Table 10: Helmholtz: common edge. Left: interpolation error for $\varepsilon_{\text{ten}} = 10^{-8}$. Right: interpolation orders and ranks of the tensor approximation in $\mathcal{H}$ -Tucker format and in (CP) format for different accuracies ε_{ten} .

with an accuracy of $\varepsilon_{\text{ten}} = 10^{-8}$. The interpolation error and the necessary ranks for the tensor approximation are summarised in Table 11.

ε_{ten}	$m + 1$	δ_0	δ_6	δ_{11}
		k_{eff}	k_{eff}	k_{eff}
		r	r	r
1e-03	6	2.9	9	5
1e-04	8	4.3	29	10
1e-05	11	6.7	79	24
1e-06	13	9.2	–	62
1e-07	15	13.2	–	–

Table 11: Helmholtz: common vertex. Left: Interpolation error for $\varepsilon_{\text{ten}} = 10^{-8}$ and different ranges of δ . Right: Interpolation orders and ranks of the tensor approximation in $\mathcal{H}$ -Tucker format and in (CP) format for different accuracies ε_{ten} and different ranges of δ .

Acknowledgements

The author thanks Prof. Dr. Dr. h.c. Wolfgang Hackbusch (Max Planck Institute for Mathematics in the Sciences, Leipzig) and Prof. Dr. Lars Grasedyck (RWTH Aachen) for helpful discussions.

References

- [1] J. Ballani, L. Grasedyck, and M. Kluge. Black box approximation of tensors in hierarchical Tucker format. Preprint 57/2010, Max Planck Institute of Mathematics in the Sciences, 2010.
- [2] M. Bonnet. Differentiability of strongly singular and hypersingular boundary integral formulations with respect to boundary perturbations. *Comp. Mech.*, 19(3):240–246, 1997.
- [3] S. Börm, L. Grasedyck, and W. Hackbusch. Hierarchical Matrices. Lecture Note 21 of the Max Planck Institute for Mathematics in the Sciences, 2003.

- [4] S. Erichsen and S. A. Sauter. Efficient automatic quadrature in 3-d Galerkin BEM. *Comput. Meth. Appl. Mech. Eng.*, 157:215–224, 1998.
- [5] M. Espig, L. Grasedyck, and W. Hackbusch. Black box low tensor rank approximation using fibre-crosses. *Constr. Approx.*, 30:557–597, 2009.
- [6] Lars Grasedyck. Hierarchical singular value decomposition of tensors. Technical Report 27, Max Planck Institute for Mathematics in the Sciences, 2009. accepted for SIMAX.
- [7] Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *J. Fourier Anal. Appl.*, 15(5):706–722, 2009.
- [8] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.
- [9] C. Lage and S. A. Sauter. On the efficient computation of singular and nearly singular integrals arising in 3D Galerkin BEM. *ZAMM*, 76:273–275, 1996.
- [10] 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.
- [11] I.V. Oseledets and E.E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432:70–88, 2010.
- [12] S. Rjasanow and O. Steinbach. *The Fast Solution of Boundary Integral Equations*. Mathematical and Analytical Techniques with Applications to Engineering. Springer, 2007.
- [13] S. A. Sauter and C. Schwab. *Randelementmethoden*. Teubner, 2004.
- [14] C. Schwab and W. L. Wendland. Kernel properties and representations of boundary integral operators. *Math. Nachr.*, 156:187–218, 1992.
- [15] J. Stoer. *Einführung in die Numerische Mathematik I*. Springer, 5th edition edition, 1989.