

**Comparison of
Fast Boundary Element Methods on
Parametric Surfaces**

H. Harbrecht, M. Peters

Institute of Mathematics
University of Basel
Rheinsprung 21
CH - 4051 Basel
Switzerland

Preprint No. 2013-01
January, 2013

www.math.unibas.ch

Comparison of Fast Boundary Element Methods on Parametric Surfaces[☆]

H. Harbrecht^a, M. Peters^a

*^aUniversity of Basel, Department of Mathematics and Computer Science, Rheinsprung
21, 4051 Basel, Switzerland*

Abstract

We compare fast black-box boundary element methods on parametric surfaces in \mathbb{R}^3 . These are the adaptive cross approximation, the multipole method based on interpolation, and the wavelet Galerkin scheme. The surface representation by a piecewise smooth parameterization is in contrast to the common approximation of surfaces by panels. Nonetheless, parametric surface representations are easily accessible from Computer Aided Design (CAD) and are recently topic of the studies in isogeometric analysis. Especially, we can apply two-dimensional interpolation in the multipole method. A main feature of this approach is that the cluster bases and the respective moment matrices are independent of the geometry. This results in a superior compression of the far field compared to other cluster methods.

Keywords: Non-local operators, fast boundary element methods, parametric surfaces.

1. Introduction

Many problems arising in science and engineering can be formulated in terms of boundary integral equations. Despite collocation and Nyström methods, the boundary element method (BEM) is a common way for the treatment of the occurring

[☆]This research has been supported by the Swiss National Science Foundation (SNSF) through the project “Rapid Solution of Boundary Value Problems on Stochastic Domains”.

Email addresses: `Helmut.Harbrecht@unibas.ch` (H. Harbrecht),
`Michael.Peters@unibas.ch` (M. Peters)

boundary integral operators, see [18, 32, 34]. Since in general the underlying boundary integral operator is non-local, as for instance for the Laplace equation, for the Helmholtz equation or for the heat equation, all referred methods will lead to linear systems with fully populated system matrices. Thus, the numerical solution of such problems requires large amounts of time and computation capacities.

To overcome this obstruction, in the last decades several ideas for the efficient approximation of the discrete system have been developed. They all exploit the fact that the system matrices exhibit a certain compressibility property. Most prominent examples of such methods are the fast multipole method [14], the panel clustering [19], the wavelet Galerkin scheme [5, 10, 30], and the adaptive cross approximation [2]. These discretization methods end up with linear or almost linear complexity, i.e. up to a poly-logarithmic factor, with respect to the number of boundary elements.

In this article, we compare the different approaches, namely the *Wavelet Galerkin Scheme* (WGS) [10], the interpolated *Fast Multipole Method* (FMM) as proposed in [12, 29, 15] and the *Adaptive Cross Approximation* (ACA) [2, 4]. The FMM and the ACA are used in the framework of the hierarchical matrix representation [16] for the low-rank approximation of the respective matrix blocks, whereas the WGS is strongly related with an adaptive sparse grid approach [7]. In the present comparison, we are interested in both, the computational performance and the approximation power, of the different approaches with respect to the solution of boundary integral equations.

Our particular realizations are based on a parametric surface representation by four-sided patches. Such parametric surface representations can be obtained directly from computer aided design (CAD). They are recently studied in the context of isogeometric analysis [27] and wavelet boundary element methods in [20, 21]. As we will see, one major advantage of parametric surfaces stems from the fact that more geometric information is available, which can therefore be exploited in the discretization. Especially, no difficulties arise if geometric entities occur in the kernel function, like the normal or tangents, as for example in the double layer operator or the adjoint double layer operator.

A further specialty of all of our fast boundary element methods is that they can be regarded as black-box algorithms for the discretization of Hilbert-Schmidt style operators since there is no explicit knowledge of the integral kernel presumed except for the smoothness apart from the diagonal. Specifically, we propose a completely new multipole algorithm based on the parametric representation of the surface. In this algorithm, we exploit the fact that the surface is a \mathbb{R}^2 -manifold and so the

problem is inherently two-dimensional. This results in a dramatic reduction of the computational effort. The achieved compression in the case of a polynomial expansion of the kernel function is even better than that of \mathcal{H}^2 -matrices, cf. [15].

This article is organized as follows. At first, in Section 2, we introduce the surface representation under consideration. Resulting from this representation, the mesh generation is straightforwardly. In Section 3, we consider the boundary integral equations and their properties. The respective Galerkin discretization is performed in Section 4. Then, Section 5 is dedicated to the FMM. Here, we present our new algorithm which perfectly fits the framework of parametric surfaces. Section 6 is concerned with the ACA and the specific consequences for the convergence theory of the method in view of the simplified interpolation on the two-dimensional reference domain. The subsequent Section 7 surveys then on the WGS. The numerical comparison of FMM, ACA and WGS is performed in Section 8. The conclusion is stated in Section 9. Finally, in the Appendix 10, we show that the standard kernel estimates imply our specific kernel estimates from Section 3.

In the following, in order to avoid the repeated use of generic but unspecified constants, by $C \lesssim D$ we mean that C can be bounded by a multiple of D , independently of parameters which C and D may depend on. Obviously, $C \gtrsim D$ is defined as $D \lesssim C$, and $C \sim D$ as $C \lesssim D$ and $C \gtrsim D$.

2. Surface Representation

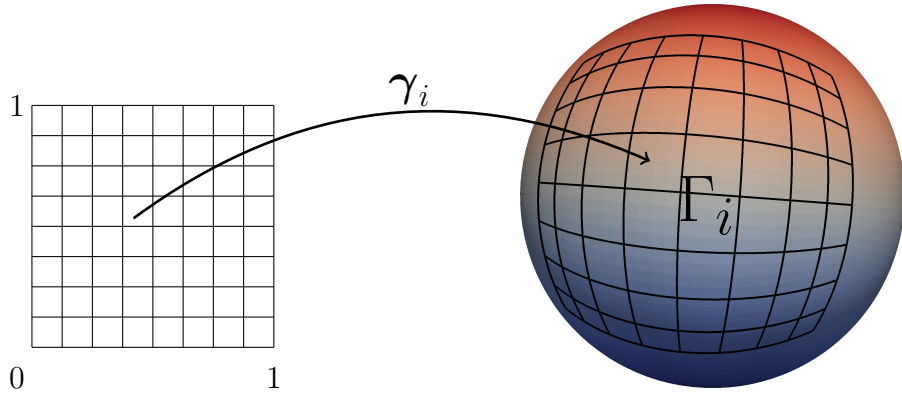


Figure 2.1: Surface representation and mesh generation

Let $\Omega \subset \mathbb{R}^3$ be the computational domain whose piecewise smooth surface $\Gamma := \partial\Omega$ is globally Lipschitz continuous. We construct a piecewise smooth parameterization of the surface Γ as follows. Let $\square := [0, 1]^2$ denote the unit square. We subdivide the given manifold Γ into several smooth *patches*

$$\Gamma = \bigcup_{i=1}^M \Gamma_i,$$

where the intersection $\Gamma_i \cap \Gamma_{i'}$ consists at most of a common vertex or a common edge for $i \neq i'$. Then, for each patch, there exists a smooth diffeomorphism

$$\gamma_i : \square \rightarrow \Gamma_i \quad \text{with} \quad \Gamma_i = \gamma_i(\square) \quad \text{for } i = 1, 2, \dots, M.$$

In fact, for our analysis, we will later need that these diffeomorphisms are also analytic functions.

In the following we will also refer to the *surface measure* of these diffeomorphisms. On the patch Γ_i , it will be denoted by

$$\kappa_i(\mathbf{s}) := \left\| \partial_{s_1} \gamma_i(\mathbf{s}) \times \partial_{s_2} \gamma_i(\mathbf{s}) \right\|_2. \quad (2.1)$$

The proposed surface representation yields an exact representation of the surface which is in contrast to the common approximation of surfaces by panels. Especially, there is no further approximation step required if the surface is given in this form. As a result, the rate of convergence is not limited by the accuracy of the surface approximation.

A whole bunch of these *parametric surfaces* are available as technical surfaces generated by tools from Computer Aided Design (CAD). The most common geometry representation in CAD is defined by the IGES (Initial Graphics Exchange Specification) standard. Here, the initial CAD object is a solid, bounded by a closed surface that is given as a collection of parametric surfaces which can be trimmed or untrimmed. An untrimmed surface is already a four-sided patch, parameterized over a rectangle. Whereas, a trimmed surface is just a piece of a supporting untrimmed surface, described by boundary curves. There are several representations of the parameterizations including B-splines, NURBS (nonuniform rational B-Splines), surfaces of revolution, and tabulated cylinders [26].

In [21], an algorithm has been developed to decompose a technical surface, described in the IGES format, into a collection of parameterized four-sided patches, fulfilling

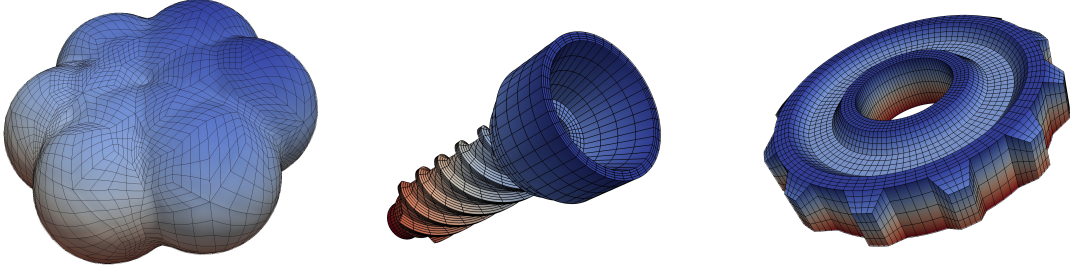


Figure 2.2: Different parametric surfaces

all the above requirements. In [20, 22], the algorithm has been extended to molecular surfaces. Figure 2.2 visualizes two parameterizations which satisfy the present requirements.

With the surface representation at hand, it is easily possible to generate a nested sequence of meshes on the surface Γ . A mesh \mathcal{Q}_j on level j for Γ is induced by dyadic subdivisions of depth j of the unit square into 4^j congruent squares, each of which is lifted to Γ by the associated parameterization γ_i (see Figure 2.1 for a visualization).

The above procedure yields a nested and quad-tree structured sequence $\mathcal{Q}_0 \subset \mathcal{Q}_1 \subset \dots \subset \mathcal{Q}_J$ of meshes consisting of $N_j = 4^j M$ elements on level j . We will refer to the particular elements as $\Gamma_{i,j,k}$ where i is the index of the applied parameterization γ_i , j is the level of the element and k is the index of the element in hierarchical order. To simplify the notation we will also denote the triple (i, j, k) by $\boldsymbol{\lambda} := (i, j, k)$ with $|\boldsymbol{\lambda}| := j$.

It is moreover convenient to refer to $\Gamma_{i,j,k}$ also as a *cluster*. In this case we think of $\Gamma_{i,j,k}$ as the union $\{\Gamma_{i,J,k'} : \Gamma_{i,J,k'} \subset \Gamma_{i,j,k}\}$, i.e. the set of all tree leafs appended to $\Gamma_{i,j,k}$ or its sons. Furthermore, we denote the collection of all clusters, the *cluster tree*, by \mathcal{T} . A scheme for the subdivisions of the patch Γ_i up to level 2 is shown in Figure 2.3.

3. Problem Formulation

We shall consider boundary integral equations on the closed, parametric surface $\Gamma := \partial\Omega$ of a given three-dimensional Lipschitz domain Ω :

$$(\mathcal{A}u)(\mathbf{x}) = \int_{\Gamma} k(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) d\sigma_{\mathbf{y}} = f(\mathbf{x}). \quad (3.2)$$

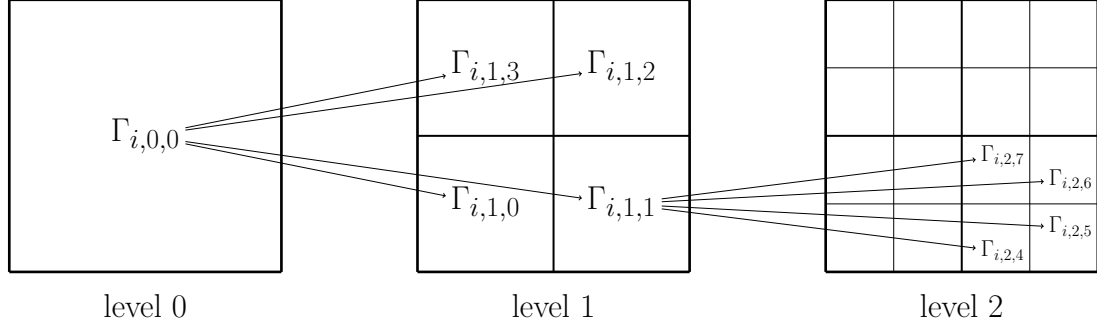


Figure 2.3: Visualization of the element tree

Herein, the boundary integral operator \mathcal{A} is an operator of order $2q$, which means that it maps $H^q(\Gamma)$ continuously and one-to-one onto $H^{-q}(\Gamma)$. The kernel functions under consideration are supposed to be smooth as functions in the variables \mathbf{x} and \mathbf{y} , apart from the diagonal $\{(\mathbf{x}, \mathbf{y}) \in \Gamma \times \Gamma : \mathbf{x} = \mathbf{y}\}$ and may have a singularity on the diagonal. Such kernel functions arise, for instance, by applying a boundary integral formulation to a second order elliptic problem [32, 34]. In general, they decay like a negative power of the distance of the arguments which depends on the order $2q$ of the operator and the spatial dimension.

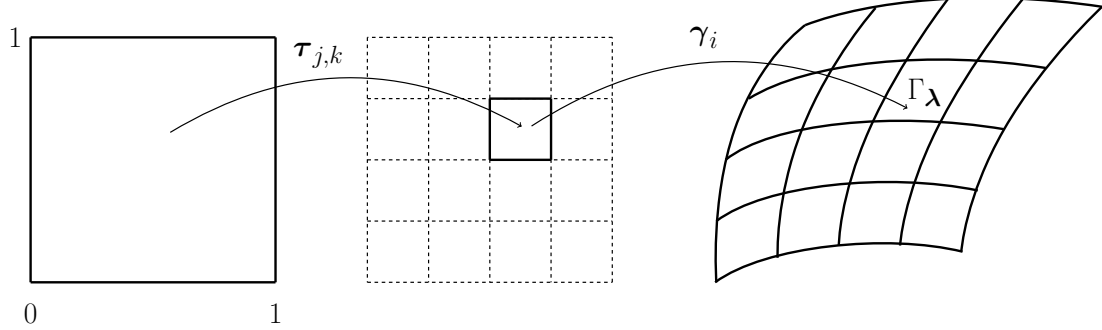


Figure 3.4: Localized parameterization

The variational formulation of the boundary integral equation (3.2) is given as follows:

$$\text{Find } u \in H^q(\Gamma) \text{ such that } (\mathcal{A}u, v)_{L^2(\Gamma)} = (f, v)_{L^2(\Gamma)} \text{ for all } v \in H^q(\Gamma). \quad (3.3)$$

If we insert the parameterizations, the bilinear form reads as

$$\begin{aligned} (\mathcal{A}u, v)_{L^2(\Gamma)} &= \int_{\Gamma} \int_{\Gamma} k(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) v(\mathbf{x}) \, d\sigma_{\mathbf{y}} \, d\sigma_{\mathbf{x}} \\ &= \sum_{i, i'=1}^M \int_{\square} \int_{\square} k_{i, i'}(\mathbf{s}, \mathbf{t}) u(\gamma_{i'}(\mathbf{t})) v(\gamma_i(\mathbf{s})) \, d\mathbf{t} \, d\mathbf{s} \end{aligned}$$

and the linear form reads as

$$\begin{aligned} (f, v)_{L^2(\Gamma)} &= \int_{\Gamma} f(\mathbf{x}) v(\mathbf{x}) \, d\sigma_{\mathbf{x}} \\ &= \sum_{i=1}^M \int_{\square} f(\gamma_i(\mathbf{s})) v(\gamma_i(\mathbf{s})) \kappa_i(\mathbf{s}) \, d\mathbf{s}. \end{aligned}$$

Here, the kernels $k_{i, i'}$ denote the *transported kernel functions*

$$\left. \begin{aligned} k_{i, i'}: \square \times \square &\longrightarrow \mathbb{R}, \\ k_{i, i'}(\mathbf{s}, \mathbf{t}) &:= k(\gamma_i(\mathbf{s}), \gamma_{i'}(\mathbf{t})) \kappa_i(\mathbf{s}) \kappa_{i'}(\mathbf{t}) \end{aligned} \right\} \quad i, i' = 1, 2, \dots, M. \quad (3.4)$$

Since the kernel $k(\mathbf{x}, \mathbf{y})$ is in general asymptotically smooth, the analyticity of the parameterizations $\{\gamma_i\}_{i=1}^M$ give raise to a decay estimate for the transported kernel function which is quite similar to (10.24).

Definition 3.1. A kernel function $k(\mathbf{x}, \mathbf{y})$ is called *analytically standard of order $2q$* if constants $c_k > 0$ and $r_k > 0$ exist such that the partial derivatives of the transported kernel functions $k_{i, i'}(\mathbf{s}, \mathbf{t})$ are uniformly bounded by

$$|\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} k_{i, i'}(\mathbf{s}, \mathbf{t})| \leq c_k \frac{(|\alpha| + |\beta|)!}{r_k^{|\alpha| + |\beta|}} \|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{-(2+2q+|\alpha|+|\beta|)} \quad (3.5)$$

provided that $2 + 2q + |\alpha| + |\beta| > 0$.

Remark 3.2. The parameterizations provide patchwise smoothness. Hence, under this assumptions, most kernels of boundary integral operators \mathcal{A} of order $2q$ are analytically standard of order $2q$. Indeed, in the Appendix 10, we present a proof of this statement.

In the context of the Galerkin scheme, it will be convenient to have also access to the *localized kernel functions*. Let $\square_{j, k} := \gamma_i^{-1}(\Gamma_{i, j, k})$ be the k -th element of the subdivided unit square on level j and define the affine mapping

$$\tau_{j, k}: \square \rightarrow \square_{j, k} \quad \text{for } j = 0, 1, \dots, J \text{ and } k = 0, 1, \dots, 4^j M - 1.$$

via dilatation and translation. Then, the localized kernel functions are given by

$$k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t}) := k(\gamma_\lambda(\mathbf{s}), \gamma_{\lambda'}(\mathbf{t})) \kappa_\lambda(\mathbf{s}) \kappa_{\lambda'}(\mathbf{t}) \quad (3.6)$$

with the *localized parameterizations* $\gamma_\lambda := \gamma_i \circ \tau_{j,k}$ and the corresponding surface measures $\kappa_\lambda := 2^{-2j} \kappa_i \circ \tau_{j,k}$ with κ_i defined in (2.1). An illustration of the mappings γ_λ is given by Figure 3.4.

In the following we will only consider the localized kernel functions. The following proposition is an immediate consequence of the fact that $\partial_{\mathbf{s}}^\alpha \tau_{j,k}(\mathbf{s}) = 2^{-j}$ if $|\alpha| = 1$ and $\partial_{\mathbf{s}}^\alpha \tau_{j,k}(\mathbf{s}) = 0$ if $|\alpha| > 1$.

Proposition 3.3. *Let the kernel function $k(\mathbf{x}, \mathbf{y})$ be analytically standard of order $2q$. Then, there exist constants $c_k > 0$ and $r_k > 0$ such that*

$$|\partial_{\mathbf{s}}^\alpha \partial_{\mathbf{t}}^\beta k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t})| \leq c_k \frac{(|\alpha| + |\beta|)!}{r_k^{|\alpha| + |\beta|}} \frac{2^{-|\lambda|(|\alpha|+2)} 2^{|\lambda'|(|\beta|+4)}}{\|\gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|_2^{2+2q+|\alpha|+|\beta|}} \quad (3.7)$$

holds uniformly for all λ, λ' provided that $2 + 2q + |\alpha| + |\beta| > 0$.

4. Galerkin Discretization

We shall be concerned with the Galerkin scheme for the discretization of the variational formulation (3.3). To this end, define

$$\hat{V}_j := \{\hat{\varphi}: \square \rightarrow \mathbb{R}: \varphi|_{\square_{j,k}} \text{ is a polynomial of order } d\} \subset L^2(\square).$$

Then, the ansatz space V_j on level j is given by

$$V_j := \{\gamma_i \circ \hat{\varphi}: \hat{\varphi} \in \hat{V}_j, i = 1, \dots, M\} \subset L^2(\Gamma).$$

This construction of the ansatz spaces obviously yields a nested sequence

$$V_0 \subset V_1 \subset \dots \subset V_J \subset H^t(\Gamma), \quad (4.8)$$

where the Sobolov smoothness t depends on the global smoothness of the functions $\varphi \in V_j$. Especially, for transported piecewise constant functions ($d = 1$), we have $t < 1/2$ and, for globally continuous, transported piecewise linear functions ($d = 2$), we have $t < 3/2$.

By replacing the energy space $H^q(\Gamma)$ in the variational formulation (3.3) by the finite dimensional ansatz space $V_J \subset H^q(\Gamma)$, we arrive at the Galerkin scheme for the boundary integral equation (3.2):

Find $u_J \in V_J$, such that

$$\int_{\Gamma} \int_{\Gamma} k(\mathbf{x}, \mathbf{y}) u_J(\mathbf{y}) v_J(\mathbf{x}) d\sigma_{\mathbf{y}} d\sigma_{\mathbf{x}} = \int_{\Gamma} f(\mathbf{x}) v_J(\mathbf{x}) d\sigma_{\mathbf{x}} \text{ for all } v_J \in V_J. \quad (4.9)$$

By setting $\hat{u}_{\lambda} := u_J \circ \gamma_{\lambda}$ and $\hat{v}_{\lambda} := v_J \circ \gamma_{\lambda}$, we might rewrite (4.9) and arrive at the equation

$$\sum_{|\lambda'|=J} \int_{\square} \int_{\square} k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t}) \hat{u}_{\lambda'}(\mathbf{t}) \hat{v}_{\lambda}(\mathbf{s}) d\mathbf{t} d\mathbf{s} = \int_{\square} f(\gamma_{\lambda}(\mathbf{s})) \hat{v}_{\lambda}(\mathbf{s}) \kappa_{\lambda}(\mathbf{s}) d\mathbf{s} \quad (4.10)$$

for all λ with $|\lambda| = J$. In the case of elementwise supported, piecewise polynomial basis functions for V_J , this yields immediately the system of linear equations

$$\mathbf{A} \mathbf{u} = \mathbf{f}. \quad (4.11)$$

Otherwise, for basis functions of higher global smoothness, straightforward but obvious modifications have to be made to arrive at the linear system (4.11).

For the Galerkin solution $u_J \in V_J$, we obtain the following error estimate by the use of the standard approximation property for ansatz functions of polynomial exactness d . Note that the rate of convergence doubles due to the Aubin-Nitsche trick.

Theorem 4.1. *Let $u \in H^q(\Gamma)$ be the solution of the boundary integral equation (3.2) and $u_J \in V_J$ the related Galerkin solution. Then, it holds the error estimate*

$$\|u - u_J\|_{H^{2q-d}(\Gamma)} \lesssim 2^{2J(q-d)} \|u\|_{H^d(\Gamma)}$$

provided that u and Γ are sufficiently regular.

5. Fast Multipole Method

In the chosen basis representation, i.e. in the single-scale basis for V_J , the system matrix \mathbf{A} in (4.11) is in general densely populated. This yields a rather high computational effort for the assembly and the matrix-vector multiplication. Fortunately the system matrix is block-wise of low rank, i.e. it is compressible in terms of an \mathcal{H} -matrix, cf. [16]. The computational complexity can thus be drastically reduced by a block-wise compression scheme. To determine compressible matrix blocks we employ the following *admissibility condition*.

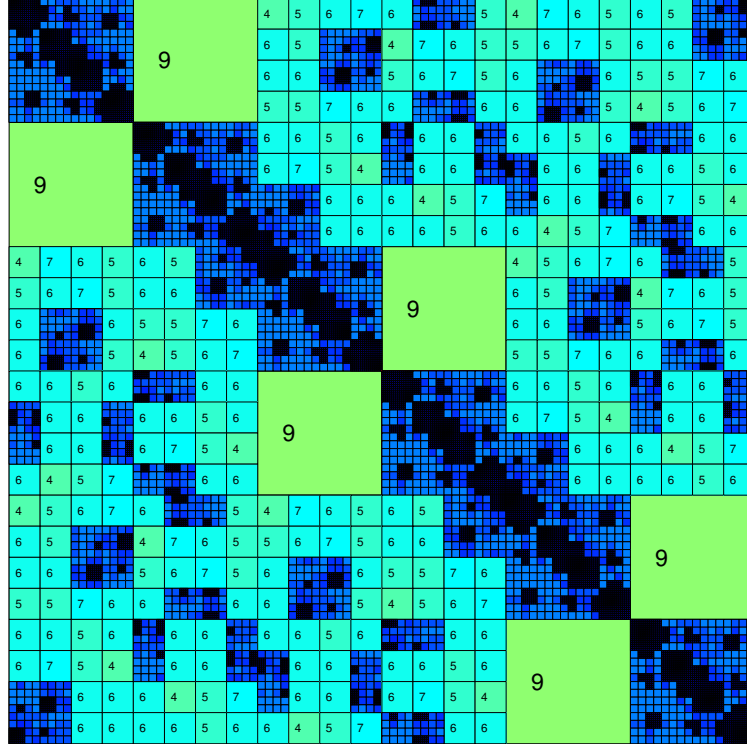


Figure 5.5: The special block partitioning of the \mathcal{H} -matrix.

Definition 5.1. *The clusters Γ_{λ} and $\Gamma_{\lambda'}$ with $|\lambda| = |\lambda'|$ are called admissible if*

$$\max \{ \text{diam}(\Gamma_{\lambda}), \text{diam}(\Gamma_{\lambda'}) \} \leq \eta \text{dist}(\Gamma_{\lambda}, \Gamma_{\lambda'}) \quad (5.12)$$

holds for some fixed $\eta \in (0, 1)$. The collection of admissible blocks $\Gamma_{\lambda} \times \Gamma_{\lambda'}$ forms the far-field of the operator. The remaining non-admissible blocks correspond to the near-field of the operator.

The quad-tree structure of the cluster tree \mathcal{T} yields thus a block partitioning of the system matrix with quadratic blocks and each block on a particular level contains exactly the same number of panel-panel interactions, see also Figure 5.5 for a visualization of this special block partitioning of an \mathcal{H} -matrix.

To compress the admissible matrix blocks, we impose two different approaches. On the one hand, we use the *Adaptive Cross Approximation* (ACA) [2], on the other hand, we develop a new version of the *Fast Multipole Method* (FMM) which is

adopted to our parametric surface representation. The latter one provides superior compression rates and computation times for the system matrix.

We develop first the black-box version of the FMM based on the interpolation of the kernel $k(\mathbf{x}, \mathbf{y})$ as firstly proposed in [12]. Note that, later on, this idea was also followed in [15] to construct \mathcal{H}^2 -matrices.

For a given polynomial degree $p \in \mathbb{N}$, let $\{x_0, x_1, \dots, x_p\} \subset [0, 1]$ be $p + 1$ pairwise distinct points. Furthermore, let $L_m(s)$ for $m = 0, \dots, p$ be the Lagrangian basis polynomials with respect to the interpolation points x_m for $m = 0, \dots, p$. By a tensor product construction we get the interpolation points $\mathbf{x}_{\mathbf{m}} := (x_{m_1}, x_{m_2})$ and the corresponding tensor product interpolation polynomials $\mathbf{L}_{\mathbf{m}}(\mathbf{s}) := L_{m_1}(s_1) \cdot L_{m_2}(s_2)$ for $m_1, m_2 = 0, \dots, p$. Then, in all admissible blocks $\Gamma_{\lambda} \times \Gamma_{\lambda'}$, we approximate

$$k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t}) \approx \sum_{\|\mathbf{m}\|_{\infty}, \|\mathbf{m}'\|_{\infty} \leq p} k_{\lambda, \lambda'}(\mathbf{x}_{\mathbf{m}}, \mathbf{x}_{\mathbf{m}'}) \mathbf{L}_{\mathbf{m}}(\mathbf{s}) \mathbf{L}_{\mathbf{m}'}(\mathbf{t}). \quad (5.13)$$

Consider now two basis functions $\hat{\varphi}_{\ell}, \hat{\varphi}_{\ell'} \in \hat{V}_{J-|\lambda|}$ of the ansatz space on the level $J - |\lambda|$. Since we employ quadrilateral meshes, we may exploit the tensor product structure of the ansatz functions. Therefore, let further $\hat{\varphi}_{\ell} = \hat{\varphi}_{\ell}^{(1)} \otimes \hat{\varphi}_{\ell}^{(2)}$ and $\hat{\varphi}_{\ell'} = \hat{\varphi}_{\ell'}^{(1)} \otimes \hat{\varphi}_{\ell'}^{(2)}$, respectively. From this and (5.13), we derive now

$$\begin{aligned} [\mathbf{A}_{\lambda, \lambda'}]_{\ell, \ell'} &\approx \int_{\square} \int_{\square} \sum_{\|\mathbf{m}\|_{\infty}, \|\mathbf{m}'\|_{\infty} \leq p} k_{\lambda, \lambda'}(\mathbf{x}_{\mathbf{m}}, \mathbf{x}_{\mathbf{m}'}) \mathbf{L}_{\mathbf{m}}(\mathbf{s}) \mathbf{L}_{\mathbf{m}'}(\mathbf{t}) \hat{\varphi}_{\ell}(\mathbf{s}) \hat{\varphi}_{\ell'}(\mathbf{t}) \, d\mathbf{s} \, d\mathbf{t} \\ &= \sum_{\|\mathbf{m}\|_{\infty}, \|\mathbf{m}'\|_{\infty} \leq p} k_{\lambda, \lambda'}(\mathbf{x}_{\mathbf{m}}, \mathbf{x}_{\mathbf{m}'}) \int_{\square} \mathbf{L}_{\mathbf{m}}(\mathbf{s}) \hat{\varphi}_{\ell}(\mathbf{s}) \, d\mathbf{s} \int_{\square} \mathbf{L}_{\mathbf{m}'}(\mathbf{t}) \hat{\varphi}_{\ell'}(\mathbf{t}) \, d\mathbf{t} \\ &=: [\mathbf{M}_{|\lambda|}^{\square} \mathbf{K}_{\lambda, \lambda'} (\mathbf{M}_{|\lambda'|}^{\square})^{\top}]_{\ell, \ell'}. \end{aligned}$$

By construction, each cluster on a particular level contains the same number of basis functions, namely $\dim(\hat{V}_{J-|\lambda|})$. Additionally, the *moment matrices* $\mathbf{M}_{|\lambda|}^{\square}$ are independent of the patch parameterization. This yields the

Proposition 5.2. *For $j = 1, 2, \dots, J$ and all $|\lambda| = |\lambda'| = j$, it holds*

$$\mathbf{M}_{|\lambda|}^{\square} = \mathbf{M}_{|\lambda'|}^{\square}. \quad (5.14)$$

As a consequence we have to compute and store only a single moment matrix

$$\mathbf{M}_{|\lambda|}^{\square} \in \mathbb{R}^{\dim(\hat{V}_{J-|\lambda|}) \times (p+1)^2}$$

for each particular level. These moment matrices can be decomposed further by exploiting the tensor product structure of the basis functions:

$$\begin{aligned}
\int_{\square} \mathbf{L}_{\mathbf{m}}(\mathbf{s}) \hat{\varphi}_{\ell}(\mathbf{s}) \, d\mathbf{s} &= \int_0^1 \int_0^1 L_{m_1}(s_1) \hat{\varphi}_{\ell}^{(1)}(s_1) L_{m_2}(s_2) \hat{\varphi}_{\ell}^{(2)}(s_2) \, ds_1 \, ds_2 \\
&= \int_0^1 L_{m_1}(s_1) \hat{\varphi}_{\ell}^{(1)}(s_1) \, ds_1 \int_0^1 L_{m_2}(s_2) \hat{\varphi}_{\ell}^{(2)}(s_2) \, ds_2 \\
&= [\mathbf{M}_{|\lambda|} \otimes \mathbf{M}_{|\lambda|}]_{\ell, (p+1)m_1+m_2}.
\end{aligned}$$

Since

$$\mathbf{M}_{|\lambda|} \in \mathbb{R}^{\sqrt{\dim(\hat{V}_{J-|\lambda|}) \times (p+1)}},$$

we end up with a major compression of the far-field.

Remark 5.3. *It is convenient to impose a lower threshold for the far-field. Therefore we consider matrix blocks with $\mathcal{O}(p^4)$ entries as near-field. This yields $\mathcal{O}(N_J p^{-2})$ near-field blocks with a storage cost of $\mathcal{O}(N_J p^2)$.*

Theorem 5.4. *The complexity for the computation and the storage of the far-field is given by $\mathcal{O}(N_J p^2)$.*

Proof. On level $j-1$, there exist $\mathcal{O}(1)$ clusters which do not satisfy the admissibility condition (5.12). For such clusters, we have to consider the son clusters on level j . Therefore, we face $\mathcal{O}(N_j)$ non-admissible and also $\mathcal{O}(N_j)$ admissible cluster-cluster interactions on level j . Furthermore, the maximum level to be computed is now $\lceil J - 2 \log_4 p \rceil$. Due to $N_j \sim 4^j M$, we thus may estimate

$$\sum_{j=0}^{\lceil J - 2 \log_4 p \rceil} N_j = \mathcal{O}(M 4^{\lceil J - 2 \log_4 p \rceil}) = \mathcal{O}(M 4^J p^{-2}) = \mathcal{O}(N_J p^{-2}).$$

This yields, together with Remark 5.3, overall $\mathcal{O}(N_J p^{-2})$ far-field blocks and accordingly $\mathcal{O}(N_J p^{-2})$ near-field blocks.

For each far-field block, we have to evaluate and store the localized kernel function in $\mathcal{O}(p^4)$ points. The complexity for assembly and storage of the moment matrices is $\mathcal{O}(\sqrt{N_J} p)$ in total. Hence, the far-field complexity is

$$\mathcal{O}(N_J p^{-2}) \cdot \mathcal{O}(p^4) + \mathcal{O}(\sqrt{N_J} p) = \mathcal{O}(N_J p^2).$$

□

Remark 5.5. To maintain the approximation property, we have to choose $p \sim \log N_J$. This yields an over-all complexity of $\mathcal{O}(N_J(\log N_J)^2)$ for the computation and the storage of the far-field. Nevertheless, if the integrals of the near-field cannot be evaluated with constant effort, then the computational effort of the near-field computation will in general dominate. For example, in the case of tensor product Gaussian quadrature rules and the Duffy trick, cf. [32], to regularize the singular integrals, one has to increase the degree of the quadrature for all singular integrals proportionally to $|\log h|$ where $h = 2^{-J}$ defines the mesh size. Thus, the computational effort is $\mathcal{O}((\log N_J)^4)$ for each entry, which results in a complexity of $\mathcal{O}(N_J(\log N_J)^4)$ for all singular integrals. However, it can be shown that this is also the overall complexity for the whole near-field if the quadrature degree is properly decreased with the distance of the elements.

With Definition 3.1 at hand, the proof of convergence for our FMM is straightforward. We present it here for the case that Chebyshev nodes on $I := [0, 1]$, i.e. the points

$$x_m := \frac{1}{2} \left[\cos \left(\frac{2m+1}{2(p+1)} \pi \right) + 1 \right], \quad m = 0, 1, \dots, p,$$

are used for the interpolation [12, 15].

Theorem 5.6. Let $k(\mathbf{x}, \mathbf{y})$ be an analytically standard kernel of order $2q$. Then, in an admissible block $\Gamma_\lambda \times \Gamma_{\lambda'}$, it holds

$$\begin{aligned} & \left\| k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t}) - \sum_{\|\mathbf{m}\|_\infty, \|\mathbf{m}'\|_\infty \leq p} k_{\lambda, \lambda'}(\mathbf{x}_\mathbf{m}, \mathbf{x}_{\mathbf{m}'}) \mathbf{L}_\mathbf{m}(\mathbf{s}) \mathbf{L}_{\mathbf{m}'}(\mathbf{t}) \right\|_{L^\infty(\square \times \square)} \\ & \lesssim \left(\frac{\eta}{r_k} \right)^{p+1} 2^{-4|\lambda|} \left\| \gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t}) \right\|_{L^\infty(\square \times \square)}^{-2(1+q)} \end{aligned}$$

with $r_k > 0$ being the constant from Definition 3.1.

Proof. We start with the one-dimensional interpolation error for the Chebyshev interpolation. It is well known that for a sufficiently smooth function $f: I \rightarrow \mathbb{R}$ the error estimate

$$\|f - \Pi_I^p f\|_{L^\infty(I)} \leq \frac{2 \cdot 4^{-(p+1)}}{(p+1)!} \|\partial^{p+1} f\|_{L^\infty(I)}$$

holds. The interpolation operator Π_I^p is defined by

$$\Pi_I^p f := \sum_{m=0}^p f(x_m) L_m(x).$$

According to [15], it satisfies the stability estimate

$$\|\Pi_I^p f\|_{L^\infty(I)} \leq c \log(p+1) \|f\|_{L^\infty(I)}$$

for some constant $c > 0$. By tensorization we obtain the d -dimensional interpolation operator $\Pi_{I^d}^p$ on I^d . In our case, we interpolate on $\square \times \square$ which is isomorphic to I^4 . From [15], we know for the polynomial interpolation of a function $f: I^d \rightarrow \mathbb{R}$ in the Chebyshev nodes that

$$\|f - \Pi_{I^d}^p\|_{L^\infty(I^d)} \lesssim \sum_{\ell=1}^d \frac{(\log(p+1))^{\ell-1}}{2(p+1)!4^p} \|\partial_{s_\ell}^{p+1} f\|_{L^\infty(I^d)}.$$

Therefore, in view of (3.7), we conclude

$$\begin{aligned} & \|k_{\lambda, \lambda'} - \Pi_{\square \times \square}^p k_{\lambda, \lambda'}\|_{L^\infty(\square \times \square)} \\ & \lesssim \sum_{\ell=1}^4 \frac{(\log(p+1))^{\ell-1}}{2(p+1)!4^p} \|\partial_{s_\ell}^{p+1} k_{\lambda, \lambda'}\|_{L^\infty(\square \times \square)} \\ & \lesssim \sum_{\ell=1}^4 \frac{(\log(p+1))^{\ell-1}}{2(p+1)!4^p} \frac{(p+1)!}{r_k^{p+1}} \|\gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|_{L^\infty(\square \times \square)}^{-2(1+q)-(p+1)} 2^{-|\lambda|((p+1)+4)} \\ & \lesssim \sum_{\ell=1}^4 \frac{(\log(p+1))^{\ell-1}}{2r_k^{p+1}4^p} \text{dist}(\Gamma_\lambda, \Gamma_{\lambda'})^{-(p+1)} 2^{-|\lambda|(p+5)} \|\gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|_{L^\infty(\square \times \square)}^{-2(1+q)} \\ & \lesssim \frac{2^{-|\lambda|(p+5)}}{r_k^{p+1} \|\gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|_{L^\infty(\square \times \square)}^{2(1+q)}} \text{dist}(\Gamma_\lambda, \Gamma_{\lambda'})^{-(p+1)}. \end{aligned}$$

The admissibility condition (5.12) provides

$$\text{dist}(\Gamma_\lambda, \Gamma_{\lambda'}) \geq \frac{\max\{\text{diam}(\Gamma_\lambda), \text{diam}(\Gamma_{\lambda'})\}}{\eta}.$$

Moreover, the Lipschitz continuity of the parameterizations and their inverses imply

$$\text{diam } \Gamma_\lambda \sim 2^{-|\lambda|} \quad \text{for all } |\lambda| = 1, 2, \dots, M.$$

Hence, we may bound

$$\text{dist}(\Gamma_\lambda, \Gamma_{\lambda'}) \gtrsim \frac{2^{-|\lambda|}}{\eta}.$$

Inserting this estimate into the above expression finally yields

$$\begin{aligned} \|k_{\lambda, \lambda'} - \Pi_{\square \times \square}^p k_{\lambda, \lambda'}\|_{L^\infty(\square \times \square)} &\lesssim \frac{2^{-|\lambda|(p+5)}}{r_k^{p+1} \|\gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|_{L^\infty(\square \times \square)}^{2(1+q)}} \left(\frac{2^{-|\lambda|}}{\eta} \right)^{-(p+1)} \\ &\lesssim 2^{-4|\lambda|} \left(\frac{\eta}{r_k} \right)^{p+1} \|\gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|_{L^\infty(\square \times \square)}^{-2(1+q)}. \end{aligned}$$

□

As in [12], we can directly derive from the previous theorem an error estimate for the bilinear form which is associated with the variational formulation (3.3).

Theorem 5.7. *Let $\sigma > 0$ be arbitrarily but fixed. Then, for the integral operator \mathcal{A}_J which results from an interpolation of degree $p > 0$ of the kernel function in every admissible block and the exact representation of the kernel in all other blocks, there holds*

$$|(\mathcal{A}u, v)_{L^2(\Gamma)} - (\mathcal{A}_J u, v)_{L^2(\Gamma)}| \lesssim 2^{-J\sigma} \|u\|_{L^1(\Gamma)} \|v\|_{L^1(\Gamma)}$$

provided that $p \sim J(2 + 2q + \sigma)$.

Proof. From Theorem 5.6, one infers for admissible clusters $\Gamma_\lambda \times \Gamma_{\lambda'}$ that

$$\text{dist}(\Gamma_\lambda, \Gamma_{\lambda'}) \geq \frac{2^{-|\lambda|}}{\eta} \geq 2^{-J}$$

since $\eta < 1$ and $|\lambda| \leq J$. Therefore, it holds

$$\|k_{\lambda, \lambda'} - \Pi_{\square \times \square}^p k_{\lambda, \lambda'}\|_{L^\infty(\square \times \square)} \lesssim 2^{-4|\lambda|} \left(\frac{\eta}{r_k} \right)^{p+1} 2^{2J(1+q)}$$

for all λ, λ' with $|\lambda| = |\lambda'|$, because the kernel representation is exact in non-admissible clusters. Now, denote by $\mathcal{B} \subset \mathcal{T} \times \mathcal{T}$ the set of all matrix blocks, i.e. the

union of all admissible and of all non-admissible blocks. Then, we may write

$$\begin{aligned}
& |(\mathcal{A}u, v)_{L^2(\Gamma)} - (\mathcal{A}_J u, v)_{L^2(\Gamma)}| \\
&= \left| \sum_{(\lambda, \lambda') \in \mathcal{B}} \int_{\square} \int_{\square} (k_{\lambda, \lambda'} - \Pi_{\square \times \square}^p k_{\lambda, \lambda'})(\mathbf{s}, \mathbf{t}) u_{\lambda'}(\mathbf{t}) v_{\lambda}(\mathbf{s}) \, d\mathbf{t} \, d\mathbf{s} \right| \\
&\leq \left| \sum_{(\lambda, \lambda') \in \mathcal{B}} \int_{\square} \int_{\square} \|k_{\lambda, \lambda'} - \Pi_{\square \times \square}^p k_{\lambda, \lambda'}\|_{L^\infty(\square \times \square)} u_{\lambda'}(\mathbf{t}) v_{\lambda}(\mathbf{s}) \, d\mathbf{t} \, d\mathbf{s} \right| \\
&\lesssim 2^{-4|\lambda|} \left(\frac{\eta}{r_k} \right)^{p+1} 2^{2J(1+q)} \left| \sum_{(\lambda, \lambda') \in \mathcal{B}} \int_{\square} \int_{\square} u_{\lambda'}(\mathbf{t}) v_{\lambda}(\mathbf{s}) \, d\mathbf{t} \, d\mathbf{s} \right| \\
&\lesssim \left(\frac{\eta}{r_k} \right)^{p+1} 2^{2J(1+q)} \|u\|_{L^1(\Gamma)} \|v\|_{L^1(\Gamma)}.
\end{aligned}$$

In view of

$$\left(\frac{\eta}{r_k} \right)^{p+1} 2^{2J(1+q)} = 2^{-J\sigma} \iff p+1 = \left\lfloor \frac{J(2+2q+\sigma)}{\log_2(\eta) - \log_2(r_k)} \right\rfloor,$$

we obtain the assertion. \square

6. Adaptive Cross Approximation

We shall briefly introduce the ACA for the compression of admissible matrix blocks. As a starting point, we employ again the admissibility condition (5.12) to partition the system matrix. Then, in each admissible matrix block, we approximate $\mathbf{A}_{\lambda, \lambda'} \in \mathbb{R}^{n \times n}$ with $n = \dim(\hat{V}_{J-|\lambda|})$ by a truncated, partially pivoted Gaussian elimination, cf. [2]. To this end, we define the vectors $\ell_m, \mathbf{u}_m \in \mathbb{R}^n$ by the following iterative scheme, where $\mathbf{A}_{\lambda, \lambda'} = [a_{i,j}]_{i,j=1}^n$ is the matrix-block under consideration:

for $m = 1, 2, \dots$ set $\mathbf{u}_m = \hat{\mathbf{u}}_m / [\hat{\mathbf{u}}_m]_{j_m}$ with

$$\hat{\mathbf{u}}_m = [a_{i_m, j}]_{j=1}^n - \sum_{j=1}^{m-1} [\ell_j]_{i_m} \mathbf{u}_j \text{ and } \ell_m = [a_{i, j_m}]_{i=1}^n - \sum_{i=1}^{m-1} [u_i]_{j_m} \ell_i.$$

A criterion to guarantee the convergence of the algorithm is to choose the pivot element located in (i_m, j_m) -position as the maximum element in modulus of the

remainder $\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_{m-1} \mathbf{U}_{m-1}$. This would require the assembly of the whole matrix block $\mathbf{A}_{\lambda, \lambda'}$, which is not feasible in practice. Therefore, we employ another pivoting strategy which performs quite well in most cases. We choose j_m such that $[\hat{\mathbf{u}}_m]_{j_m}$ is the largest element in modulus of the row $\hat{\mathbf{u}}_m$.

We finally stop the iteration if the criterion

$$\|\ell_{m+1}\|_2 \|\mathbf{u}_{m+1}\|_2 \leq \varepsilon \|\mathbf{L}_m \mathbf{U}_m\|_F \quad (6.15)$$

for some desired accuracy $\varepsilon > 0$ is met. Here and in the following, $\|\cdot\|_F$ denotes the Frobenius norm. Under the assumption that

$$\|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_{m+1} \mathbf{U}_{m+1}\|_F \leq \vartheta \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F$$

holds uniformly for a fixed $\vartheta < 1$, we arrive at

$$\begin{aligned} \|\ell_{m+1}\|_2 \|\mathbf{u}_{m+1}\|_2 &= \|\mathbf{L}_{m+1} \mathbf{U}_{m+1} - \mathbf{L}_m \mathbf{U}_m\|_F \\ &\leq \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_{m+1} \mathbf{U}_{m+1}\|_F + \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F \\ &\leq (1 + \vartheta) \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F. \end{aligned}$$

On the other hand, we find

$$\begin{aligned} \|\mathbf{L}_{m+1} \mathbf{U}_{m+1} - \mathbf{L}_m \mathbf{U}_m\|_F &\geq \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F - \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_{m+1} \mathbf{U}_{m+1}\|_F \\ &\geq (1 - \vartheta) \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F. \end{aligned}$$

Therefore, we conclude that the approximation error is proportional to the norm $\|\ell_{m+1}\|_2 \|\mathbf{u}_{m+1}\|_2$ of the update vectors

$$(1 - \vartheta) \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F \leq \|\ell_{m+1}\|_2 \|\mathbf{u}_{m+1}\|_2 \leq (1 + \vartheta) \|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F.$$

Thus, together with (6.15), we can guarantee a relative error bound

$$\|\mathbf{A}_{\lambda, \lambda'} - \mathbf{L}_m \mathbf{U}_m\|_F \lesssim \varepsilon \|\mathbf{A}_{\lambda, \lambda'}\|_F. \quad (6.16)$$

Theorem 6.1. *Let \mathbf{A} be the uncompressed system matrix and $\tilde{\mathbf{A}}$ be the system matrix which is compressed by the adaptive cross approximation. Then, with respect to the Frobenius norm, there holds the error estimate*

$$\|\mathbf{A} - \tilde{\mathbf{A}}\|_F \lesssim \varepsilon \|\mathbf{A}\|_F$$

provided that the blockwise error satisfies (6.16).

Proof. In view of (6.16), we have

$$\begin{aligned}
\|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 &= \sum_{j=0}^J \sum_{|\boldsymbol{\lambda}|, |\boldsymbol{\lambda}'|=j} \|\mathbf{A}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}'} - \tilde{\mathbf{A}}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}'}\|_F^2 \\
&\lesssim \varepsilon^2 \sum_{j=0}^J \sum_{|\boldsymbol{\lambda}|, |\boldsymbol{\lambda}'|=j} \|\mathbf{A}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}'}\|_F^2 \\
&= \varepsilon^2 \|\mathbf{A}\|_F^2.
\end{aligned}$$

Taking square roots on both sides yields the assertion. \square

Obviously, the complexity for the computation of the rank- m -approximation $\mathbf{L}_m \mathbf{U}_m$ to the block $\mathbf{A}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}'}$ is $\mathcal{O}(2m^2n)$ and the storage cost is $\mathcal{O}(2mn)$. The latter one can be further reduced by the application of a singular value decomposition and neglecting non-relevant singular values.

The theoretical foundation of ACA is the successive interpolation of asymptotically smooth functions, cf. [2]. Traditionally, ACA employs the three-dimensional interpolation theory for estimating the interpolation error relative to the surface Γ . Since then the interpolation points may lie on a hyperplane for which the interpolation is not unique anymore, cf. [31], the traditional ACA may fail to converge. We refer the reader to [6] and [3], respectively, for a specific example where this happens. Nevertheless, in our framework, such situations are excluded since only the two-dimensional interpolation theory on the unit square is employed.

In the following, we restate the convergence result from [2] and adopt everything to the case that the interpolation is performed on the unit square \square and $\square \times \square$, respectively.

Let the function $f: \Gamma \times \Gamma \rightarrow \mathbb{R}$ satisfy Definition 3.1 and let $\Gamma_{\boldsymbol{\lambda}} \times \Gamma_{\boldsymbol{\lambda}'}$ be an admissible block. Consider the sequences $\{s_k\}_k, \{r_k\}_k$ given as follows. Set

$$r_0(\mathbf{s}, \mathbf{t}) := f_{\boldsymbol{\lambda}, \boldsymbol{\lambda}'}(\mathbf{s}, \mathbf{t}) \text{ and } s_0(\mathbf{s}, \mathbf{t}) := 0,$$

and compute for $k = 0, 1, \dots$

$$\begin{aligned}
r_{k+1}(\mathbf{s}, \mathbf{t}) &= r_k(\mathbf{s}, \mathbf{t}) - r_k(\mathbf{s}_{i_{k+1}}, \mathbf{t}_{j_{k+1}})^{-1} r_k(\mathbf{s}, \mathbf{t}_{j_{k+1}}) r_k(\mathbf{s}_{i_{k+1}}, \mathbf{t}), \\
s_{k+1}(\mathbf{s}, \mathbf{t}) &= s_k(\mathbf{s}, \mathbf{t}) + r_k(\mathbf{s}_{i_{k+1}}, \mathbf{t}_{j_{k+1}})^{-1} r_k(\mathbf{s}, \mathbf{t}_{j_{k+1}}) r_k(\mathbf{s}_{i_{k+1}}, \mathbf{t}).
\end{aligned}$$

Here, we have to assume explicitly that the points $\mathbf{s}_{i_{k+1}}, \mathbf{t}_{j_{k+1}} \in \square$ are chosen such that

$$r_k(\mathbf{s}_{i_{k+1}}, \mathbf{t}_{j_{k+1}})^{-1} \neq 0.$$

Then, with partial pivoting, i.e. $\mathbf{s}_{i_{k+1}}$ is chosen such that

$$|r_k(\mathbf{s}_{i_{k+1}}, \mathbf{t}_{j_{k+1}})| \geq |r_k(\mathbf{s}, \mathbf{t}_{j_{k+1}})| \text{ for all } \mathbf{s} \in \square,$$

the following error estimate can be proven, cf. [2],

$$|r_k(\mathbf{s}, \mathbf{t})| \lesssim 2^k \text{dist}(\Gamma_\lambda, \Gamma_{\lambda'})^{-2(1+q)} \eta^{\sqrt{k}}.$$

Consequently, for sufficiently small η , the remainders $|r_k(\mathbf{s}, \mathbf{t})|$ decay exponentially. According to [2] factor 2^k is not observed in most of the practical applications. Therefore, we will also omit it here for the complexity considerations which improves the results.

Theorem 6.2. *Assume that for admissible clusters Γ_λ and $\Gamma_{\lambda'}$, the remainder $r_k(\mathbf{s}, \mathbf{t})$ satisfies the estimate*

$$|r_k(\mathbf{s}, \mathbf{t})| \lesssim \text{dist}(\Gamma_\lambda, \Gamma_{\lambda'})^{-2(1+q)} \eta^{\sqrt{k}}. \quad (6.17)$$

Then, for $\varepsilon > 0$, it holds $|r_k(\mathbf{s}, \mathbf{t})| \lesssim \varepsilon$ provided that $k \sim (|\log \varepsilon| + J(2 + 2q))^2$.

Proof. Analogously to the proof of Theorem 5.7 holds

$$\text{dist}(\Gamma_\lambda, \Gamma_{\lambda'}) \geq 2^{-J}.$$

Therefore, the assertion immediately follows from

$$\varepsilon = 2^{2J(1+q)} \eta^{\sqrt{k}} \iff k = \left(\frac{\log_2 \varepsilon - 2J(1+q)}{\log_2 \eta} \right)^2.$$

□

Remark 6.3. *For the particular choice $\varepsilon = 2^{-J\sigma}$ in the above theorem, we observe that the rank k of the ACA behaves like the rank p^2 for the FMM. In fact this result is in concordance with the respective results from [12] and [6].*

Although it is not necessary to, introduce a threshold parameter for the far-field in the ACA, as discussed in Remark 5.3 for the FMM, we will consider it here. Hence, we arrive at the following Theorem, which can be proven rather analogously to Theorem 5.4.

Theorem 6.4. *Assume that (6.17) holds uniformly for all k . Furthermore, let p denote the threshold parameter from Remark 5.3. Then, the complexity for the computation of the far-field in the ACA is given by $\mathcal{O}(\lceil J - 2\log_4 p \rceil k^2 N_J)$ and the storage by $\mathcal{O}(\lceil J - 2\log_4 p \rceil k N_J)$.*

Proof. In accordance with the proof of Theorem 5.4, the complexity for the far-field computation is given by

$$\begin{aligned} \sum_{j=0}^{\lceil J-2\log_4 p \rceil} \mathcal{O}(N_j) \cdot \mathcal{O}(k^2 N_{J-j}) &= \sum_{j=0}^{\lceil J-2\log_4 p \rceil} \mathcal{O}(M4^j) \cdot \mathcal{O}(k^2 M4^{J-j}) \\ &= \mathcal{O}(\lceil J - 2\log_4 p \rceil k^2 M^2 4^J) \\ &= \mathcal{O}(\lceil J - 2\log_4 p \rceil k^2 N_J). \end{aligned}$$

A similar computation yields the complexity for the storage. \square

7. Wavelet Galerkin Scheme

The idea of the *Wavelet Galerkin Scheme* (WGS) is the use of an appropriate wavelet basis instead of the single-scale basis for the discretization of the Galerkin scheme (4.9). Instead of using only a single scale j , the idea of wavelets is to keep track to the increment of information between two adjacent scales $j-1$ and j . That is, one chooses complement spaces

$$W_j := V_j \ominus V_{j-1}$$

which are generated by the (compactly supported) wavelets of the associated level j , i.e.

$$W_j = \text{span}\{\psi_{j,k} : k \in \nabla_j\}.$$

The set ∇_j denotes an appropriate index set. By setting $W_0 := V_0$, we recursively obtain the multiscale decomposition

$$V_J = \bigoplus_{j=0}^J W_j$$

and the associated wavelet basis

$$\Psi_J := \{\psi_{j,k} : k \in \nabla_j, j \leq J\}.$$

For all further details concerning wavelet analysis, we refer the reader to the survey article [9].

In the context of wavelet matrix compression, the wavelets are required to be compactly supported

$$\text{diam}(\text{supp } \psi_{j,k}) \sim 2^{-j}$$

and to provide *vanishing moments* of order $\tilde{d} > d - 2q$ which means that

$$|(v, \psi_{j,k})_{L^2(\Gamma)}| \lesssim 2^{-j(1+\tilde{d})} |v|_{W^{\tilde{d},\infty}(\text{supp } \psi_{j,k})}. \quad (7.18)$$

Here, $|v|_{W^{\tilde{d},\infty}(\Omega)} := \sup_{|\alpha|=\tilde{d}} \|\partial^\alpha v\|_{L^\infty(\Omega)}$ denotes the usual semi-norm in $W^{\tilde{d},\infty}(\Omega)$. Piecewise constant and bilinear wavelets which match all these requirements have been constructed in [23, 25].

If we discretize the bilinear form $(\mathcal{A}u, v)_{L^2(\Gamma)}$ in wavelet coordinates, the system matrix becomes quasi-sparse. In fact, by combining (3.5) and (7.18), we arrive at the decay estimate

$$(\mathcal{A}\psi_{j,k}, \psi_{j',k'})_{L^2(\Gamma)} \lesssim \frac{2^{-(j+j')(1+\tilde{d})}}{\text{dist}(\text{supp } \psi_{j,k}, \text{supp } \psi_{j',k'})^{2(1+q+\tilde{d})}} \quad (7.19)$$

which is the main foundation of compression estimates [10]. Based on (7.19), we can set all matrix entries to zero, for which the distance of the supports between the associated trial and test functions is larger than a level dependent cut-off parameter $\mathcal{B}_{j,j'}$. A further compression, reflected by a cut-off parameter $\mathcal{B}_{j,j'}^s$, is achieved by neglecting some of those matrix entries, for which the corresponding trial and test functions have overlapping supports.

To formulate this result, we introduce the abbreviation $\Omega_\lambda^s := \text{singsupp } \psi_\lambda$ which denotes the *singular support* of the wavelet ψ_λ , i.e. that subset of Γ where the wavelet is not smooth.

Theorem 7.1 (A-priori compression [10]). *Let Ω_λ and Ω_λ^s be given as above and define the compressed system matrix \mathbf{A}_J , corresponding to the boundary integral operator \mathcal{A} , by*

$$[\mathbf{A}_J]_{\lambda,\lambda'} := \begin{cases} 0, & \text{dist}(\Omega_\lambda, \Omega_{\lambda'}) > \mathcal{B}_{|\lambda|,|\lambda'|} \text{ and } |\lambda|, |\lambda'| > 0, \\ 0, & \text{dist}(\Omega_\lambda, \Omega_{\lambda'}) \leq 2^{-\min\{|\lambda|, |\lambda'|\}} \text{ and} \\ & \text{dist}(\Omega_\lambda^s, \Omega_{\lambda'}^s) > \mathcal{B}_{|\lambda|,|\lambda'|}^s \text{ if } |\lambda'| > |\lambda| \geq 0, \\ & \text{dist}(\Omega_\lambda, \Omega_{\lambda'}^s) > \mathcal{B}_{|\lambda|,|\lambda'|}^s \text{ if } |\lambda| > |\lambda'| \geq 0, \\ (\mathcal{A}\psi_{\lambda'}, \psi_\lambda)_{L^2(\Gamma)}, & \text{otherwise.} \end{cases}$$

Fixing

$$a > 1, \quad d < \delta < \tilde{d} + 2q, \quad (7.20)$$

the cut-off parameters $\mathcal{B}_{j,j'}$ and $\mathcal{B}_{j,j'}^s$ are set as follows

$$\begin{aligned} \mathcal{B}_{j,j'} &= a \max \left\{ 2^{-\min\{j,j'\}}, 2^{\frac{2J(\delta-q)-(j+j')(\delta+\tilde{d})}{2(d+q)}} \right\}, \\ \mathcal{B}_{j,j'}^s &= a \max \left\{ 2^{-\max\{j,j'\}}, 2^{\frac{2J(\delta-q)-(j+j')\delta-\max\{j,j'\}\tilde{d}}{d+2q}} \right\}. \end{aligned}$$

Then, the system matrix \mathbf{A}_J has only $\mathcal{O}(N_J)$ nonzero coefficients. Moreover, the error estimate

$$\|u - u_J\|_{H^{2q-d}(\Gamma)} \lesssim 2^{2J(q-d)} \|u\|_{H^d(\Gamma)}$$

holds for the solution u_J of the compressed Galerkin system provided that u and Γ are sufficiently regular.

The compressed system matrix can be assembled in linear complexity if one employs the exponentially convergent *hp*-quadrature method proposed in [24]. Moreover, for performing faster matrix-vector multiplications, an additional a-posteriori compression might be applied which reduces again the number of nonzero coefficients by a factor 2–5 [10]. The pattern of the compressed system matrix exhibit the typical *finger structure*, compare Figure 7.6.

8. Numerical Results

8.1. General Setup

Beyond all the presented theory about the considered methods, this section can be seen as the essence of this article. Our main goal was to compare the presented methods, i.e. the FMM, the ACA, and the WGS, for the numerical solution of integral equations in a common framework. Here, the first task was to adopt the cluster methods to the parametric surface representation, since this surface representation is a key-ingredient for the construction of wavelets [23, 25]. The second task was to eliminate differences in the numerical implementation of the methods. Therefore, we based all three implementations on the same single-scale boundary element code. Especially, the three methods use the same quadrature routines for the occurring singular integrals in the near-field. Only the degree of quadrature is chosen method-dependent. This also points out the major difference between the cluster methods

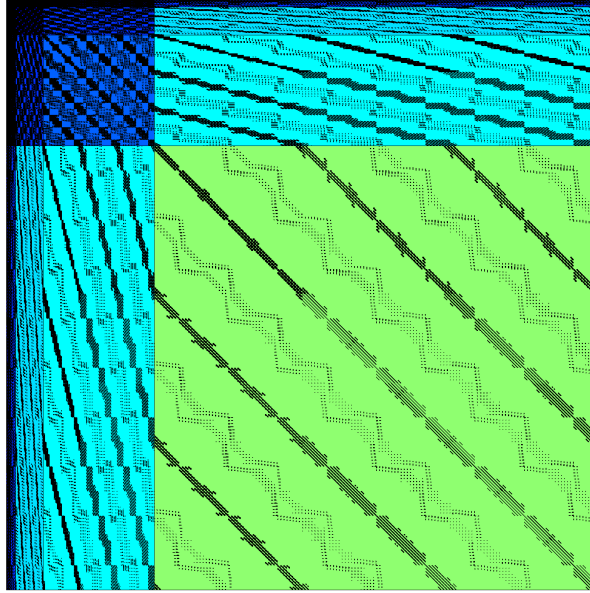


Figure 7.6: Compression pattern in case of the gearwheel geometry.

and the WGS: The cluster methods provide a compression of the stiffness matrix in the single-scale representation and use, thus, the same degrees of quadrature as in the single-scale case. Since the WGS induces a multilevel splitting of the kernel approximation, the degrees of quadrature have to be chosen differently. On each level only the discretization error accuracy has to be realized. As a consequence, the approximation in the WGS is per se different from the approximation in the single-scale case.

So, although we created circumstances to make this methods comparable, the question arises: How to compare the methods now in practice? We chose the following approach: We solve the Dirichlet problem for the Laplacian on a sphere and on a gearwheel (seen on the right in Figure 8.8) by the indirect formulation based on the single layer potential and also based on the double layer potential (see e.g. [18, 32, 34]). The sphere is represented by six patches and the gearwheel is represented by 336 patches. For the Galerkin discretization, we employ piecewise constant ansatz functions. We equilibrate the approximation errors and measure the time for assembling the stiffness matrix, the doubles per degree of freedom and the time for the solution of the linear system. As error measure, we evaluate the respective potentials in certain points. In addition, on the sphere, we also present

the L^2 -error of the computed densities.

The doubles per degree of freedom are denoted by μ_{full} for the cluster methods. In the case of the WGS they are split into μ_{prior} , the doubles per degree of freedom after the a-priori compression and μ_{post} , the doubles per degree of freedom remaining after the a-posteriori compression. Since there are big deviations in the measured times between the cluster methods and the WGS, we found it illuminating to present also the doubles per degree of freedom only consumed by the near-field (μ_{near}) in the case of the cluster methods. As it turns out, we already have $\mu_{\text{near}} \sim \mu_{\text{post}}$, no matter which cluster method and which geometry is considered. This means that overall the WGS results in a system matrix which has approximately the same number of entries as already the near-field of the cluster methods.

For the cluster methods, we split the assembly time (t_{assembly}) into the time for only compressing the far-field (t_{far}) and the time for the computation of the near-field (t_{near}), since the increasing degrees of quadrature in the near-field may camouflage the expected scaling of the methods. Additionally we provide the solution times (t_{solve}) and the required number of iterations ($\#_{\text{iter}}$).

All computations are carried out on a single core of an Intel(R) Xeon(R) X5570 2.93 Ghz compute server with 48 GB main memory. Moreover, in the numerical experiments, the parameters in the different approaches are set as follows.

ACA. The accuracy of the ACA is increased in accordance with the level J and we chose $\sigma = 3$, i.e. we choose the parameter ε in (6.15) as $\varepsilon \sim 2^{-3J}$ for Symm's integral equation and as $\varepsilon \sim 2^{-2J}$ for the double layer equation. It was necessary to employ a recompression for the ACA, realized by a singular value decomposition in the approximated matrix blocks, to keep storage costs acceptable. Furthermore, we exploit a sharpened admissibility condition in the ACA: If $\Gamma_{\lambda} \cap \Gamma_{\lambda'} = \emptyset$, we consider Γ_{λ} and $\Gamma_{\lambda'}$ to be admissible. This is justified since, due to our surface representation, there exists a fixed parameter $\eta > 0$ which depends only on the particular geometry, such that we obtain the same admissibility condition. Nevertheless, we could not observe an increase of the ranks in the approximated blocks compared to the admissibility condition (5.12). Additionally, we treat blocks of size 4×4 always as near-field independently of the discretization level J .

FMM. For the FMM, we increased the polynomial degree if necessary to maintain the convergence behavior. It is tabulated in the following tables in the column entitled " p ". The parameter η in the admissibility condition (5.12) is set to $\eta = 1.6$ except for the single layer equation on the sphere where $\eta = 1.4$ gives slightly better results. To save redundant computations we stored the matrices $\mathbf{M}_{|\lambda|}^{\square}$, rather than

$\mathbf{M}_{|\lambda|}$, once for each level. The size of the near-field is increased in accordance with the polynomial degrees, i.e. we have 4×4 blocks for $p = 3$ and 16×16 blocks for $p = 4, 5$. Compared to the other methods, the FMM takes more time to solve the linear system. Potential improvement is possible here: Since the cluster bases are only depending on the particular level, a multilevel representation of the moment matrices can be applied via the Haar basis. This yields hierarchical cluster bases which essentially results in the \mathcal{H}^2 -matrix representation, cf. [15], with the difference that the cluster transfer matrices need not to be explicitly stored.

WGS. For the WGS, we use piecewise constant wavelets with $\tilde{d} = 3$ vanishing moments as constructed in [23]. Moreover, the bandwidth parameter a in (7.20) is chosen equal to $a = 1$ on the sphere and equal to $a = 2.5$ on the gearwheel. The parameter δ in (7.20) is always chosen as $\delta = 1.25$.

Iterative Solution. In case of the symmetric system matrices which arise from the single layer operator, we assemble and store only the lower trigonal part of the system matrices. For the solution of the resulting linear systems, we employ the preconditioned conjugate gradient method, cf. [13], in the case of the single layer potential and the generalized maximum residual method, cf. [13], in the case of the double layer potential. For the WGS, a diagonal scaling is sufficient to maintain a constant number of iterations for increasing number of unknowns [10]. For the cluster methods, we implemented a wavelet preconditioner based on the Haar transform which performs a level-dependent scaling of the system matrix. Note that this wavelet preconditioning employs the norm equivalences of wavelet bases [11, 28, 33].

8.2. Symm's Integral Equation

We consider the numerical solution of Symm's integral equation

$$\int_{\Gamma} \frac{u(\mathbf{y})}{4\pi\|\mathbf{x} - \mathbf{y}\|_2} d\sigma_{\mathbf{y}} = f(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma := \partial\Omega. \quad (8.21)$$

It computes the solution

$$U(\mathbf{x}) = \int_{\Gamma} \frac{u(\mathbf{y})}{4\pi\|\mathbf{x} - \mathbf{y}\|_2} d\sigma_{\mathbf{y}} \in H^1(\Omega)$$

of the Laplace equation

$$\Delta U = 0 \text{ in } \Omega, \quad U = f \text{ on } \Gamma. \quad (8.22)$$

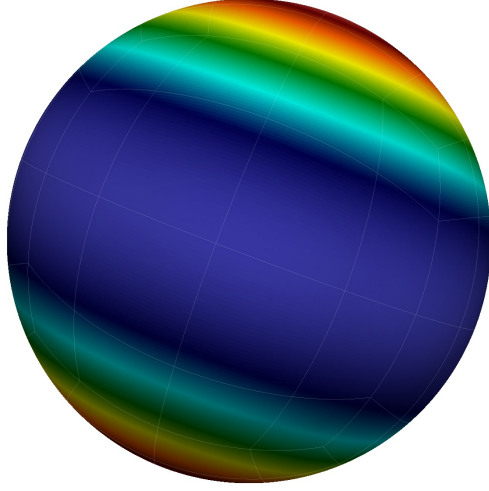


Figure 8.7: Solution of the integral equations (8.21) and (8.23) on the sphere.

Here, Ω is either the unit ball or the gearwheel seen in Figure 8.8. The right hand side f is chosen as the spherical harmonic $f(\mathbf{x}) = Y_2^0(\mathbf{x})$ in case of the sphere and as $f(\mathbf{x}) = 4x_1^2 - 3x_2^2 - x_3^2$ in case of the gearwheel. Since the spherical harmonics are the eigenfunctions of the single layer operator on the unit sphere, we know the exact density in case of the unit sphere and can thus measure the related L^2 -error. A visualization of the spherical harmonics Y_2^0 is given in Figure 8.7. Moreover, we find $U(\mathbf{x}) = \|\mathbf{x}\|_2^2 Y_2^0(\mathbf{x}/\|\mathbf{x}\|_2)$ in case of the ball and $U(\mathbf{x}) = 4x_1^2 - 3x_2^2 - x_3^2$ in case of the gearwheel.

For the unit sphere, the numerical results are tabulated in the Tables 8.1–8.3, and, for the gearwheel, the numerical results are tabulated in the Tables 8.4–8.6. We compute the potential U in 1793 and 648 evaluation points in the interior of the sphere and the gearwheel, respectively, and compute the maximum deviation to the exact solution. These numbers are found in the columns entitled “ ℓ^∞ -error”. In addition, for the unit sphere, the columns entitled “ L^2 -error” contain the error of the approximate density with respect to the exact eigenfunction Y_2^0 .

It can be seen that all methods exhibit essentially the same rates of convergence. Especially, in case of the sphere, the expected rate of convergence 2^{-3J} for the potential error and the rate of convergence 2^{-J} for the density error are achieved. Whereas, in case of the gearwheel, the rate of convergence is reduced due to the lack of smoothness.

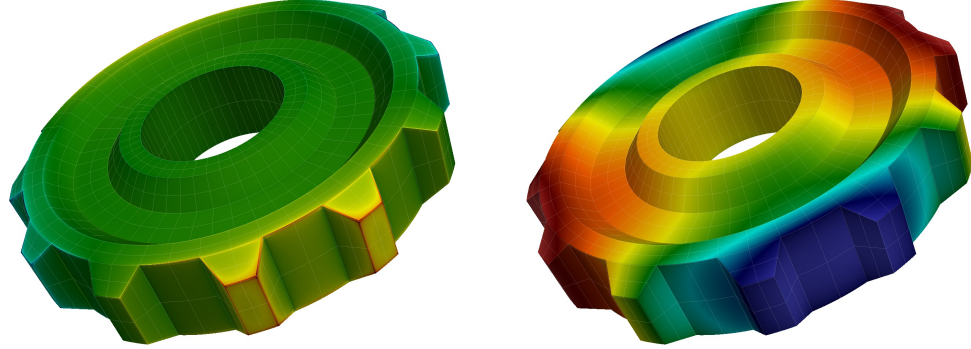


Figure 8.8: Solutions for the single layer operator (left) and for the double layer operator (right) on the gearwheel.

As mentioned already before and in concordance with Remark 5.5, the computation and the storage of the near-field in the cluster methods exceed the effort for the WGS. Indeed, the cluster methods suffer from denser populated system matrices, which is also reflected in the solution times. Nevertheless, we would like to point out the very fast computation times for the far-field of the FMM. The very drastic increase in the number of near-field entries in the FMM stems from the increase of the polynomial degree from levels five to six and levels seven to eight.

8.3. Second Kind Equation for the Double Layer Operator

In order to solve the Laplace equation (8.22) with the indirect formulation for the double layer potential, we arrive at the following Fredholm integral equation of the second kind:

$$\int_{\Gamma} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{n}_{\mathbf{y}} \rangle}{4\pi \|\mathbf{x} - \mathbf{y}\|_2^3} u(\mathbf{y}) \, d\sigma_{\mathbf{y}} - \frac{1}{2} u(\mathbf{x}) = f(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma. \quad (8.23)$$

We choose the same right hand sides as before. The spherical harmonics are also the eigenfunctions of the double layer operator on the sphere such that the density is known in this case. The density in case of the gearwheel is unknown and depicted in the right plot of Figure 8.8. The potentials, now given by

$$U(\mathbf{x}) = \int_{\Gamma} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{n}_{\mathbf{y}} \rangle}{4\pi \|\mathbf{x} - \mathbf{y}\|_2^3} u(\mathbf{y}) \, d\sigma_{\mathbf{y}} \in H^1(\Omega),$$

J	N	μ_{prior}	μ_{post}	$\ell^\infty\text{-error}$	$L^2\text{-error}$	t_{assembly}	t_{solve}	$\#\text{iter}$
1	24	13	7	$1.555 \cdot 10^{-1}$	$9.119 \cdot 10^{-1}$	< 1s	< 1s	1
2	96	28	24	$1.818 \cdot 10^{-2}$	$3.749 \cdot 10^{-1}$	< 1s	< 1s	16
3	384	58	41	$1.101 \cdot 10^{-3}$	$1.837 \cdot 10^{-1}$	< 1s	< 1s	28
4	1536	79	53	$3.026 \cdot 10^{-4}$	$9.150 \cdot 10^{-2}$	< 1s	< 1s	38
5	6144	94	65	$1.323 \cdot 10^{-5}$	$4.560 \cdot 10^{-2}$	4s	< 1s	46
6	24576	107	77	$4.211 \cdot 10^{-6}$	$2.284 \cdot 10^{-2}$	25s	< 1s	53
7	98304	117	85	$2.754 \cdot 10^{-7}$	$1.145 \cdot 10^{-2}$	136s	3s	60
8	393216	124	91	$4.000 \cdot 10^{-8}$	$5.730 \cdot 10^{-3}$	727s	13s	67
9	1572864	129	96	$7.768 \cdot 10^{-9}$	$2.873 \cdot 10^{-3}$	4034s	63s	74

Table 8.1: Results of the WGS for Symm's integral equation on the sphere.

J	N	p	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	$L^2\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	24	3	14	14	$1.679 \cdot 10^{-1}$	$8.857 \cdot 10^{-1}$	< 1s	< 1s	< 1s	1
2	96	3	50	50	$2.064 \cdot 10^{-2}$	$3.716 \cdot 10^{-1}$	< 1s	< 1s	< 1s	19
3	384	3	194	194	$1.076 \cdot 10^{-3}$	$1.830 \cdot 10^{-1}$	< 1s	< 1s	< 1s	27
4	1536	3	770	196	$1.816 \cdot 10^{-4}$	$9.081 \cdot 10^{-2}$	1s	< 1s	< 1s	32
5	6144	3	922	200	$2.534 \cdot 10^{-5}$	$4.529 \cdot 10^{-2}$	4s	< 1s	3s	37
6	24576	4	1239	798	$2.850 \cdot 10^{-6}$	$2.263 \cdot 10^{-2}$	49s	< 1s	13s	42
7	98304	4	1251	776	$3.212 \cdot 10^{-7}$	$1.131 \cdot 10^{-2}$	360s	4s	73s	46
8	393216	5	1743	765	$9.664 \cdot 10^{-9}$	$5.655 \cdot 10^{-3}$	1538s	33s	545s	50
9	1572864	5	1725	760	$1.719 \cdot 10^{-9}$	$2.827 \cdot 10^{-3}$	10942s	130s	2754s	54

Table 8.2: Results of the FMM for Symm's integral equation on the sphere.

J	N	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	$L^2\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	24	14	14	$1.679 \cdot 10^{-1}$	$8.857 \cdot 10^{-1}$	< 1s	< 1s	< 1s	1
2	96	45	42	$1.784 \cdot 10^{-2}$	$3.745 \cdot 10^{-1}$	< 1s	< 1s	< 1s	22
3	384	107	66	$7.860 \cdot 10^{-4}$	$1.831 \cdot 10^{-1}$	< 1s	< 1s	< 1s	27
4	1536	230	72	$1.916 \cdot 10^{-4}$	$9.081 \cdot 10^{-2}$	1s	< 1s	< 1s	32
5	6144	448	74	$2.371 \cdot 10^{-5}$	$4.529 \cdot 10^{-2}$	4s	2s	< 1s	37
6	24576	776	74	$2.791 \cdot 10^{-6}$	$2.263 \cdot 10^{-2}$	35s	17s	3s	41
7	98304	1250	74	$3.621 \cdot 10^{-7}$	$1.131 \cdot 10^{-2}$	279s	156s	20s	46
8	393216	1888	74	$4.194 \cdot 10^{-8}$	$5.655 \cdot 10^{-3}$	1188s	1303s	120s	50
9	1572864	2681	74	$6.269 \cdot 10^{-9}$	$2.827 \cdot 10^{-3}$	8531s	9392s	734s	54

Table 8.3: Results of the ACA for Symm's integral equation on the sphere.

J	N	μ_{prior}	μ_{post}	$\ell^\infty\text{-error}$	t_{assembly}	t_{solve}	$\#\text{iter}$
1	1344	151	107	$3.805 \cdot 10^{-1}$	< 1s	< 1s	81
2	5376	233	166	$4.184 \cdot 10^{-1}$	2s	< 1s	48
3	21504	359	217	$7.244 \cdot 10^{-3}$	19s	1s	66
4	86016	450	167	$1.692 \cdot 10^{-3}$	134s	6s	89
5	344064	505	154	$3.735 \cdot 10^{-4}$	719s	28s	108
6	1376256	543	159	$4.316 \cdot 10^{-5}$	4159s	142s	129

Table 8.4: Results of the WGS for Symm's integral equation on the gearwheel.

J	N	p	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	1344	3	674	674	$2.130 \cdot 10^{-1}$	< 1s	< 1s	< 1s	42
2	5376	3	2690	357	$2.130 \cdot 10^{-2}$	1s	2s	11s	63
3	21504	3	2005	255	$7.091 \cdot 10^{-3}$	9s	6s	51s	78
4	86016	3	1453	220	$1.478 \cdot 10^{-3}$	96s	16s	223s	90
5	344064	4	1633	880	$3.125 \cdot 10^{-4}$	557s	38s	1170s	102
6	1376256	4	1402	803	$6.168 \cdot 10^{-5}$	4621s	122s	5769s	114

Table 8.5: Results of the FMM for Symm's integral equation on the gearwheel.

J	N	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	1344	674	674	$2.130 \cdot 10^{-1}$	< 1s	< 1s	< 1s	42
2	5376	417	76	$2.882 \cdot 10^{-2}$	< 1s	3s	2s	72
3	21504	679	75	$1.686 \cdot 10^{-2}$	8s	16s	9s	80
4	86016	1256	74	$2.546 \cdot 10^{-3}$	90s	88s	41s	91
5	344064	1742	74	$4.795 \cdot 10^{-4}$	345s	503s	204s	102
6	1376256	2460	74	$7.030 \cdot 10^{-5}$	3213s	2386s	1199s	114

Table 8.6: Results of the ACA for Symm's integral equation on the gearwheel.

J	N	μ_{prior}	μ_{post}	$\ell^\infty\text{-error}$	$L^2\text{-error}$	t_{assembly}	t_{solve}	$\#\text{iter}$
1	24	24	10	$2.275 \cdot 10^{-1}$	$5.130 \cdot 10^{-1}$	< 1s	< 1s	1
2	96	54	23	$1.338 \cdot 10^{-2}$	$2.431 \cdot 10^{-1}$	< 1s	< 1s	4
3	384	108	37	$1.143 \cdot 10^{-2}$	$1.210 \cdot 10^{-1}$	< 1s	< 1s	11
4	1536	140	42	$8.258 \cdot 10^{-4}$	$6.032 \cdot 10^{-2}$	1s	< 1s	16
5	6144	157	46	$3.183 \cdot 10^{-4}$	$3.015 \cdot 10^{-2}$	5s	< 1s	20
6	24576	163	47	$5.926 \cdot 10^{-5}$	$1.508 \cdot 10^{-2}$	23s	< 1s	24
7	98304	166	47	$1.244 \cdot 10^{-5}$	$7.537 \cdot 10^{-3}$	107s	< 1s	25
8	393216	167	47	$3.417 \cdot 10^{-6}$	$3.768 \cdot 10^{-3}$	475s	3s	26
9	1572864	168	46	$1.660 \cdot 10^{-6}$	$1.884 \cdot 10^{-3}$	2107s	13s	26

Table 8.7: Results of the WGS for the double layer equation on the sphere.

J	N	p	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	$L^2\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	24	3	24	24	$2.512 \cdot 10^{-1}$	$4.920 \cdot 10^{-1}$	< 1s	< 1s	< 1s	4
2	96	3	96	96	$2.971 \cdot 10^{-2}$	$2.365 \cdot 10^{-1}$	< 1s	< 1s	< 1s	4
3	384	3	389	368	$5.776 \cdot 10^{-3}$	$1.199 \cdot 10^{-1}$	< 1s	< 1s	< 1s	3
4	1536	3	1483	364	$1.524 \cdot 10^{-3}$	$6.021 \cdot 10^{-2}$	4s	< 1s	< 1s	3
5	6144	3	1741	369	$3.872 \cdot 10^{-4}$	$3.014 \cdot 10^{-2}$	16s	2s	< 1s	3
6	24576	3	1824	357	$4.912 \cdot 10^{-5}$	$1.507 \cdot 10^{-2}$	151s	8s	< 1s	2
7	98304	3	1799	350	$2.929 \cdot 10^{-6}$	$7.537 \cdot 10^{-3}$	1776s	29s	4s	2
8	393216	3	1769	346	$4.048 \cdot 10^{-6}$	$3.768 \cdot 10^{-3}$	5463s	114s	17s	2
9	1572864	3	1745	344	$1.303 \cdot 10^{-6}$	$1.884 \cdot 10^{-3}$	42374s	446s	78s	2

Table 8.8: Results of the FMM for the double layer equation on the sphere.

J	N	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	$L^2\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	24	24	24	$2.512 \cdot 10^{-1}$	$4.920 \cdot 10^{-1}$	< 1s	< 1s	< 1s	4
2	96	82	80	$2.932 \cdot 10^{-2}$	$2.364 \cdot 10^{-1}$	< 1s	< 1s	< 1s	4
3	384	174	128	$5.774 \cdot 10^{-3}$	$1.199 \cdot 10^{-1}$	< 1s	< 1s	< 1s	3
4	1536	344	140	$1.526 \cdot 10^{-3}$	$6.022 \cdot 10^{-2}$	4s	< 1s	< 1s	3
5	6144	546	143	$3.623 \cdot 10^{-4}$	$3.014 \cdot 10^{-2}$	15s	2s	< 1s	3
6	24576	887	144	$2.272 \cdot 10^{-5}$	$1.507 \cdot 10^{-2}$	145s	16s	< 1s	3
7	98304	1439	144	$1.533 \cdot 10^{-5}$	$7.537 \cdot 10^{-3}$	1194s	195s	< 1s	2
8	393216	2285	144	$9.625 \cdot 10^{-6}$	$3.768 \cdot 10^{-3}$	5039s	2699s	4s	2
9	1572864	3588	144	$3.151 \cdot 10^{-6}$	$1.884 \cdot 10^{-3}$	41950s	23601s	426s	2

Table 8.9: Results of the ACA for the double layer equation on the sphere.

J	N	μ_{prior}	μ_{post}	$\ell^\infty\text{-error}$	t_{assembly}	t_{solve}	$\#\text{iter}$
1	1344	281	88	$8.001 \cdot 10^{-1}$	1s	< 1s	13
2	5376	328	90	$1.778 \cdot 10^{-1}$	6s	< 1s	14
3	21504	435	95	$5.214 \cdot 10^{-2}$	40s	< 1s	22
4	86016	481	88	$9.237 \cdot 10^{-3}$	340s	1s	31
5	344064	473	76	$6.193 \cdot 10^{-3}$	1761s	4s	37
6	1376256	453	64	$6.165 \cdot 10^{-4}$	8465s	20s	40

Table 8.10: Results of the WGS for the double layer equation on the gearwheel.

J	N	p	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	1344	3	1344	1344	$4.680 \cdot 10^{-1}$	1s	< 1s	< 1s	15
2	5376	3	5132	710	$1.674 \cdot 10^{-1}$	6s	9s	2s	16
3	21504	3	3885	507	$4.868 \cdot 10^{-2}$	6s	28s	10s	16
4	86016	3	2726	436	$1.448 \cdot 10^{-2}$	683s	78s	35s	15
5	344064	3	2023	398	$5.706 \cdot 10^{-3}$	3357s	346s	136s	15
6	1376256	3	1582	375	$2.717 \cdot 10^{-3}$	25394s	1623s	549s	15

Table 8.11: Results of the FMM for the double layer equation on the gearwheel.

J	N	μ_{full}	μ_{near}	$\ell^\infty\text{-error}$	t_{near}	t_{far}	t_{solve}	$\#\text{iter}$
1	1344	1344	1344	$4.680 \cdot 10^{-1}$	1s	< 1s	< 1s	15
2	5376	801	149	$1.706 \cdot 10^{-1}$	3s	7s	< 1s	16
3	21504	1050	145	$4.160 \cdot 10^{-2}$	58s	22s	1s	16
4	86016	1604	144	$1.172 \cdot 10^{-2}$	638s	116s	5s	15
5	344064	2421	144	$5.258 \cdot 10^{-3}$	3108s	653s	26s	15
6	1376256	3230	144	$3.718 \cdot 10^{-3}$	23796s	3850s	117s	15

Table 8.12: Results of the ACA for the double layer equation on the gearwheel.

satisfy again $U(\mathbf{x}) = \|\mathbf{x}\|_2^2 Y_2^0(\mathbf{x}/\|\mathbf{x}\|_2)$ in case of the ball and $U(\mathbf{x}) = 4x_1^2 - 3x_2^2 - x_3^2$ in case of the gearwheel. The numerical results are tabulate in the Tables 8.7–8.9 for the unit sphere and in the Tables 8.10–8.12 for the gearwheel.

The potentials are evaluated in the same points as in the previous examples, i.e. in 1793 points within the unit ball and in 648 points within the gearwheel. The related errors are found again in the columns entitled “ ℓ^∞ -error”. For the unit sphere, we also give the L^2 -error of the approximate density in the columns entitled “ L^2 -error”. Again, all methods produce essentially the same rates of convergence. In case of the sphere, the expected rate of convergence 2^{-2J} for the potential error and the rate of convergence 2^{-J} for the density error are achieved. In case of the gearwheel, the rate of convergence for the potential error is again slightly reduced.

For the double layer operator, we have to assemble the whole system matrix since no symmetry can be exploited which leads to a higher storage consumption. In particular, we observe that the computational effort of the cluster methods is again mainly driven by the near-field computation. It becomes expensive due to the level dependent increase of accuracy. On the other hand, the accuracies increase much slower, so that a polynomial degree of $p = 3$ is sufficient on all levels for the FMM to achieve comparable errors. Thus, in the numerical results, found in the Table 8.8 for the unit sphere and in the Table 8.11 for the gearwheel, we clearly see the expected linear behavior of the FMM for the fixed polynomial degree in the far-field and the solution times.

For the ACA on the unit sphere, we have a drastic increase from level eight to nine in the memory per degree of freedom to obtain an error of the same order of magnitude as the other methods. On the gearwheel geometry, the ACA profits from the occurring zero blocks of the double layer operator. They appear due to the fact that many patches of the gearwheel lie on a common hyperplane and the kernel becomes zero since $\mathbf{n}_\mathbf{y} \perp (\mathbf{x} - \mathbf{y})$ for all \mathbf{x} and \mathbf{y} in this hyperplane. In such blocks, the ACA truncates in the first step without storing anything. This is in contrast to the representation of the other two methods. However, despite this improvement of the compression rate, both, the WGS and the FMM, still have a smaller storage consumption.

9. Conclusion

In this article, fast boundary element methods for parametric surfaces have been presented and compared. The WGS was already available for parametric surfaces

(see [10, 24]) since the construction of wavelets relies on such surfaces. The interpolation based FMM yields an extremely efficient far-field approximation since only the cluster bases on the interval have to be provided. The ACA for parametric surfaces works also for kernels which involve geometrical entities like normals or tangents.

It turned out from the numerical experiments that the WGS yields superior compression rates of the system matrix. Indeed, the numbers of relevant matrix coefficients are less or equal to the numbers of near-field coefficients of the cluster methods. The FMM and the ACA behave quite similar, which is no surprise due to Remark 6.3. Nevertheless, the computation of the far-field and its storage is much more efficient in the FMM compared to the ACA, but the matrix-vector multiplication and, thus, the solution time is slower.

The drawback of the WGS is the complicated numerical quadrature. Here, quadrature formulae are required which compute the integrals on the coarse levels with high accuracy.

10. Appendix

Usually, for fast boundary element methods, it is assumed that the kernel is asymptotically smooth in the *space*. This means

$$|\partial_{\mathbf{x}}^{\alpha} \partial_{\mathbf{y}}^{\beta} k(\mathbf{x}, \mathbf{y})| \leq c_k \frac{(|\alpha| + |\beta|)!}{r_k^{|\alpha| + |\beta|}} \|\mathbf{x} - \mathbf{y}\|_2^{-2-2q-|\alpha|-|\beta|} \quad (10.24)$$

for some constants $c_k > 0$ and $r_k > 0$ which are independent of α and β .

We show in the following theorem that the decay estimate (10.24) implies the condition (3.5) provided that the parameterization is piecewise analytically.

Theorem 10.1. *Let the kernel function $k(\mathbf{x}, \mathbf{y})$ fulfill the decay estimate (10.24) and let the parameterizations γ_i and $\gamma_{i'}$ be analytic functions. Then, for all $i, i' = 1, 2, \dots, M$, there exist constants $R_{i,i'} > 0$ and $c > 0$ such that the transported kernel (3.4) satisfies the estimate*

$$|\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} k_{i,i'}(\mathbf{s}, \mathbf{t})| \leq c_k \frac{(|\alpha| + |\beta|)!}{R_{i,i'}^{|\alpha| + |\beta|}} \frac{\|\kappa_i\|_{L^{\infty}(\square)} \|\kappa_{i'}\|_{L^{\infty}(\square)}}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha|+|\beta|}} \quad (10.25)$$

uniformly for all α, β provided that $2 + 2q + |\alpha| + |\beta| > 0$.

Proof. For the following, it is convenient to compute an estimate of

$$\tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t}) := k(\gamma_i(\mathbf{s}), \gamma_{i'}(\mathbf{t})).$$

To this end, without the loss of generality, we assume that the boundary Γ is scaled such that $\text{diam}(\Gamma) \leq 1$ and thus

$$\|\gamma_i\|_{L^\infty(\square)} \leq 1, \quad \|\gamma_{i'}\|_{L^\infty(\square)} \leq 1. \quad (10.26)$$

Then, the partial derivatives of $\tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})$ can be expressed in terms of the Faa di Bruno formula (see [8]) in accordance with

$$\begin{aligned} \partial_{\mathbf{s}}^\alpha \partial_{\mathbf{t}}^\beta \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t}) &= \sum_{\substack{1 \leq |\alpha'| \leq |\alpha| \\ 1 \leq |\beta'| \leq |\beta|}} \partial_{\mathbf{x}}^{\alpha'} \partial_{\mathbf{y}}^{\beta'} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t}) \left(\sum_{p(\alpha, \alpha')} \alpha! \prod_{a=1}^{|\alpha|} \frac{(\partial_{\mathbf{s}}^{\nu_a} \gamma_i(\mathbf{s}))^{\mu_a}}{(\mu_a!)(\nu_a!)^{|\mu_a|}} \right) \\ &\quad \cdot \left(\sum_{p(\beta, \beta')} \beta! \prod_{b=1}^{|\beta|} \frac{(\partial_{\mathbf{t}}^{\nu_b} \gamma_{i'}(\mathbf{t}))^{\mu_b}}{(\mu_b!)(\nu_b!)^{|\mu_b|}} \right) \end{aligned} \quad (10.27)$$

with multi-indices $\alpha, \beta \in \mathbb{N}^2$ and $\alpha', \beta' \in \mathbb{N}^3$. Moreover, the set $p(\alpha, \alpha')$ is given by

$$\begin{aligned} p(\alpha, \alpha') &:= \left\{ (\mu_a, \nu_a)_{a=1}^{|\alpha|} \in (\mathbb{N}^3, \mathbb{N}^2)^{|\alpha|} \text{ with } \sum_a \mu_a = \alpha' \text{ and } \sum_a |\mu_a| \nu_a = \alpha : \right. \\ &\quad \text{there exists an } s \in [1, |\alpha|) \text{ such that } |\mu_a| = |\nu_a| = 0 \ \forall 1 \leq a \leq s \\ &\quad \left. \text{and } |\mu_a| \neq 0 \ \forall s+1 \leq a \leq |\alpha| \wedge \mathbf{0} \prec \nu_{s+1} \prec \cdots \prec \nu_{|\alpha|} \right\}. \end{aligned}$$

Here, the relation $\nu \prec \nu'$ means either $|\nu| < |\nu'|$ or, if $|\nu| = |\nu'|$, it denotes the lexicographical order which means in the present two-dimensional setting $\nu_1 < \nu'_1$.

Since the parameterizations are analytic and in view of our scaling (10.26), the Cauchy integral formula, cf. [1], implies

$$[\partial_{\mathbf{s}}^\alpha \gamma_i(\mathbf{s})]_j \leq \frac{\alpha!}{\rho_i^{|\alpha|}}, \quad [\partial_{\mathbf{t}}^\beta \gamma_{i'}(\mathbf{t})]_j \leq \frac{\beta!}{\rho_{i'}^{|\beta|}}, \quad j = 1, 2, 3 \quad (10.28)$$

for some $\rho_i, \rho_{i'} \in (0, 1]$. Inserting these estimates into (10.27) yields

$$\begin{aligned}
|\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})| &\leq \sum_{\substack{1 \leq |\alpha'| \leq |\alpha| \\ 1 \leq |\beta'| \leq |\beta|}} |\partial_{\mathbf{x}}^{\alpha'} \partial_{\mathbf{y}}^{\beta'} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})| \left(\sum_{p(\alpha, \alpha')} \alpha! \prod_{a=1}^{|\alpha|} \frac{\rho_i^{-|\nu_a|} \mu_a!}{\mu_a!} \right) \\
&\quad \cdot \left(\sum_{p(\beta, \beta')} \beta! \prod_{b=1}^{|\beta|} \frac{\rho_{i'}^{-|\nu_b|} \mu_b!}{\mu_b!} \right) \\
&= \frac{\alpha! \beta!}{\rho_i^{|\alpha|} \rho_{i'}^{|\beta|}} \sum_{\substack{1 \leq |\alpha'| \leq |\alpha| \\ 1 \leq |\beta'| \leq |\beta|}} |\partial_{\mathbf{x}}^{\alpha'} \partial_{\mathbf{y}}^{\beta'} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})| \sum_{p(\alpha, \alpha')} \prod_{a=1}^{|\alpha|} \frac{1}{\mu_a!} \sum_{p(\beta, \beta')} \prod_{b=1}^{|\beta|} \frac{1}{\mu_b!}.
\end{aligned}$$

We shall next determine upper bounds of the two last terms of this expression. To this end, we employ the identity

$$\sum_{p(\alpha, \alpha')} \prod_{b=1}^{|\alpha|} \frac{1}{\mu_b!} = \frac{|s^+(\alpha, \alpha')|}{\alpha'!},$$

provided by [8], where

$$s^+(\alpha, \alpha') := \left\{ (\eta_1, \dots, \eta_{|\alpha'|}) : |\eta_a| \neq 0 \text{ and } \sum_a \eta_a = \alpha \right\}.$$

To bound the cardinality of the set $s^+(\alpha, \alpha')$ we use the estimates for the number of weak integer compositions, cf. [17], in each of the two components of $\alpha = (\alpha_1, \alpha_2)$. This yields the trivial combinatorial estimate

$$|s^+(\alpha, \alpha')| \leq \binom{\alpha_1 + |\alpha'| - 1}{|\alpha'| - 1} \binom{\alpha_2 + |\alpha'| - 1}{|\alpha'| - 1} \leq 2^{|\alpha| + 2(|\alpha'| - 1)}.$$

Thus, we arrive at

$$|\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})| \leq \frac{2^{|\alpha| + |\beta|} \alpha! \beta!}{16 \rho_i^{|\alpha|} \rho_{i'}^{|\beta|}} \sum_{\substack{1 \leq |\alpha'| \leq |\alpha| \\ 1 \leq |\beta'| \leq |\beta|}} \frac{4^{|\alpha'| + |\beta'|}}{\alpha'! \beta'!} |\partial_{\mathbf{x}}^{\alpha'} \partial_{\mathbf{y}}^{\beta'} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})|.$$

By inserting the assertion (10.24), we deduce

$$\begin{aligned} |\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})| &\leq \frac{c_k}{16} \frac{2^{|\alpha|+|\beta|}}{\rho_i^{|\alpha|} \rho_{i'}^{|\beta|}} \alpha! \beta! \\ &\cdot \sum_{\substack{1 \leq |\alpha'| \leq |\alpha| \\ 1 \leq |\beta'| \leq |\beta|}} \frac{(|\alpha'| + |\beta'|)!}{\alpha'! \beta'!} \left(\frac{4}{r_k}\right)^{|\alpha'|+|\beta'|} \frac{1}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha'|+|\beta'|}}. \end{aligned}$$

Since $\text{diam}(\Gamma) \leq 1$, we find

$$\frac{1}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha'|+|\beta'|}} \leq \frac{1}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha|+|\beta|}} \quad (10.29)$$

for all $|\alpha'| \leq |\alpha|$, $|\beta'| \leq |\beta|$ and $\mathbf{s}, \mathbf{t} \in \square$. Thus, it holds

$$\begin{aligned} |\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})| &\leq \frac{c_k}{16} \alpha! \beta! \frac{2^{|\alpha|+|\beta|} \rho_i^{-|\alpha|} \rho_{i'}^{-|\beta|}}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha|+|\beta|}} \\ &\cdot \underbrace{\sum_{\substack{1 \leq |\alpha'| \leq |\alpha| \\ 1 \leq |\beta'| \leq |\beta|}} \frac{(|\alpha'| + |\beta'|)!}{\alpha'! \beta'!} \left(\frac{4}{r_k}\right)^{|\alpha'|+|\beta'|}}_{=: R(\alpha, \beta)}. \end{aligned}$$

Next, we shall derive an estimation for $R(\alpha, \beta)$. Obviously, it is

$$(|\alpha| + |\beta|)! = \binom{|\alpha| + |\beta|}{|\beta|} |\alpha|! |\beta|! = \binom{|\alpha| + |\beta|}{|\alpha|} |\alpha|! |\beta|!. \quad (10.30)$$

Additionally, we have by means of multinomial coefficients

$$(|\alpha| + |\beta|)! = \binom{|\alpha| + |\beta|}{|\beta|} \binom{|\alpha|}{\alpha} \binom{|\beta|}{\beta} \alpha! \beta!.$$

Further, we know

$$(x_1 + \dots + x_m)^{|\alpha|} = \sum_{|\alpha'|=|\alpha|} \binom{|\alpha|}{\alpha'} x_1^{\alpha'_1} \dots x_m^{\alpha'_m}$$

for multi-indices $\alpha, \alpha' \in \mathbb{N}^m$ and $x_1, \dots, x_m \in \mathbb{R}$. Therefore, we arrive at

$$\begin{aligned}
R(\alpha, \beta) &= \sum_{\substack{1 \leq |\alpha'| \leq |\alpha| \\ 1 \leq |\beta'| \leq |\beta|}} \frac{(|\alpha'| + |\beta'|)!}{\alpha'! \beta'!} \left(\frac{4}{r_k} \right)^{|\alpha'| + |\beta'|} = \sum_{a=1}^{|\alpha|} \sum_{b=1}^{|\beta|} \sum_{\substack{|\alpha'|=a \\ |\beta'|=b}} \frac{(a+b)!}{\alpha'! \beta'!} \left(\frac{4}{r_k} \right)^{a+b} \\
&= \sum_{a=1}^{|\alpha|} \sum_{b=1}^{|\beta|} \binom{a+b}{b} \sum_{\substack{|\alpha'|=a \\ |\beta'|=b}} \binom{a}{\alpha'} \binom{b}{\beta'} \left(\frac{4}{r_k} \right)^{a+b} \\
&\leq \sum_{a=1}^{|\alpha|} \sum_{b=1}^{|\beta|} 2^{a+b} \left(\frac{12}{r_k} \right)^{a+b} = \sum_{a=1}^{|\alpha|} \left(\frac{24}{r_k} \right)^a \sum_{b=1}^{|\beta|} \left(\frac{24}{r_k} \right)^b \\
&\leq \left(\frac{24}{24 - r_k} \right)^2 \left(\frac{24}{r_k} \right)^{|\alpha| + |\beta|} \leq 2 \left(\frac{24}{r_k} \right)^{|\alpha| + |\beta|}.
\end{aligned}$$

Altogether, this yields

$$|\partial_s^\alpha \partial_t^\beta \tilde{k}_{i,i'}(s, t)| \leq \frac{c_k}{8} \alpha! \beta! \left(\frac{48}{r_k} \right)^{|\alpha| + |\beta|} \frac{\rho_i^{-|\alpha|} \rho_{i'}^{-|\beta|}}{\|\gamma_i(s) - \gamma_{i'}(t)\|_2^{2(1+q) + |\alpha| + |\beta|}}. \quad (10.31)$$

Now, we can easily estimate the decay of the transported kernel function (3.4) by the Leibniz formula. It holds

$$\partial_s^\alpha \partial_t^\beta k_{i,i'}(s, t) = \sum_{\substack{\alpha' \leq \alpha \\ \beta' \leq \beta}} \binom{\alpha}{\alpha'} \binom{\beta}{\beta'} \partial_s^{\alpha'} \partial_t^{\beta'} \tilde{k}_{i,i'}(s, t) \partial_s^{\alpha - \alpha'} \kappa_i(s) \partial_t^{\beta - \beta'} \kappa_{i'}(t). \quad (10.32)$$

Since the surface measures are analytic, we can estimate them also by the Cauchy integral formula. It is

$$|\partial_s^\alpha \kappa_i(s)| \leq \frac{\alpha!}{\tilde{\rho}_i^{|\alpha|}} \|\kappa_i\|_{L^\infty(\square)}, \quad |\partial_t^\beta \kappa_{i'}(t)| \leq \frac{\beta!}{\tilde{\rho}_{i'}^{|\beta|}} \|\kappa_{i'}\|_{L^\infty(\square)}$$

for some $\tilde{\rho}_i, \tilde{\rho}_{i'} > 0$. We insert these estimates into (10.32) and arrive at

$$|\partial_s^\alpha \partial_t^\beta k_{i,i'}(s, t)| \leq \alpha! \beta! \|\kappa_i\|_{L^\infty(\square)} \|\kappa_{i'}\|_{L^\infty(\square)} \sum_{\substack{\alpha' \leq \alpha \\ \beta' \leq \beta}} \frac{|\partial_s^{\alpha'} \partial_t^{\beta'} \tilde{k}_{i,i'}(s, t)|}{\alpha'! \beta'!} \tilde{\rho}_i^{-|\alpha - \alpha'|} \tilde{\rho}_{i'}^{-|\beta - \beta'|}.$$

Next, we use (10.31) to replace the derivatives $\partial_{\mathbf{s}}^{\alpha'} \partial_{\mathbf{t}}^{\beta'} \tilde{k}_{i,i'}(\mathbf{s}, \mathbf{t})$ which, in view of (10.29) and with $R_{i,i'} := \min\{\rho_i, \rho_{i'}, \tilde{\rho}_i, \tilde{\rho}_{i'}, 1\}$, leads to

$$\begin{aligned} |\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} k_{i,i'}(\mathbf{s}, \mathbf{t})| &\leq \frac{c_k}{8} \alpha! \beta! \|\kappa_i\|_{L^{\infty}(\square)} \|\kappa_{i'}\|_{L^{\infty}(\square)} \\ &\quad \cdot \sum_{\substack{\alpha' \leq \alpha \\ \beta' \leq \beta}} \frac{\alpha'! \beta'!}{\alpha! \beta'!} \left(\frac{48}{r_k}\right)^{|\alpha'|+|\beta'|} \frac{\rho_i^{-|\alpha'|} \tilde{\rho}_i^{-|\alpha-\alpha'|} \rho_{i'}^{-|\beta'|} \tilde{\rho}_{i'}^{-|\beta-\beta'|}}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha|+|\beta|}} \\ &\leq \frac{c_k}{8} \alpha! \beta! \frac{\|\kappa_i\|_{L^{\infty}(\square)} \|\kappa_{i'}\|_{L^{\infty}(\square)}}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha|+|\beta|}} \cdot \sum_{\substack{\alpha' \leq \alpha \\ \beta' \leq \beta}} \left(\frac{48}{R_{i,i'} r_k}\right)^{|\alpha'|+|\beta'|} \end{aligned}$$

The sum remaining sum can be estimated via

$$\sum_{\substack{\alpha' \leq \alpha \\ \beta' \leq \beta}} q^{|\alpha'|+|\beta'|} \leq \frac{q^{\alpha_1+1} - 1}{q - 1} \cdot \frac{q^{\alpha_2+1} - 1}{q - 1} \cdot \frac{q^{\beta_1+1} - 1}{q - 1} \cdot \frac{q^{\beta_2+1} - 1}{q - 1} \leq \left(\frac{q}{q - 1}\right)^4 q^{|\alpha|+|\beta|}.$$

For $q = 48/(R_{i,i'} r_k)$, we have

$$\left(\frac{48}{48 - R_{i,i'} r_k}\right)^4 \leq 2.$$

With the help of (10.30), it finally follows

$$|\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} k_{i,i'}(\mathbf{s}, \mathbf{t})| \leq \frac{c_k}{4} (|\alpha| + |\beta|)! \left(\frac{48}{R_{i,i'} r_k}\right)^{|\alpha|+|\beta|} \frac{\|\kappa_i\|_{L^{\infty}(\square)} \|\kappa_{i'}\|_{L^{\infty}(\square)}}{\|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|_2^{2(1+q)+|\alpha|+|\beta|}}.$$

This is the desired estimate (10.25). \square

References

- [1] H. Amann and J. Escher. *Analysis II*. Birkhäuser, Basel, 2006.
- [2] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86:565–589, 2000.
- [3] M. Bebendorf and R. Grzhibovskis. Accelerating Galerkin BEM for Linear Elasticity using Adaptive Cross Approximation. *Math. Methods Appl. Sci.*, 29:1721–1747, 2006.

- [4] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70:1–24, 2003.
- [5] G. Beylkin, R. Coifman, and V. Rokhlin. The fast wavelet transform and numerical algorithms. *Comm. Pure and Appl. Math.*, 44:141–183, 1991.
- [6] S. Börm and L. Grasedyck. Hybrid cross approximation of integral operators. *Numer. Math*, 101:221–249, 2005.
- [7] H.J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.
- [8] G.M. Constantine and T.H. Savits. A multivariate Faà di Bruno formula with applications. *Trans. Amer. Math. Soc.*, 248:503–520, 1996.
- [9] W. Dahmen. Wavelet and multiscale methods for operator equations. *Acta Numerica*, 6:55–228, 1997.
- [10] W. Dahmen, H. Harbrecht, and R. Schneider. Compression techniques for boundary integral equations. Optimal complexity estimates. *SIAM J. Numer. Anal.*, 43:2251–2271, 2006.
- [11] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.*, 63:315–344, 1992.
- [12] K. Giebermann. Multilevel Approximation of Boundary Integral Operators. *Computing*, 67:183–207, 2001.
- [13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.
- [14] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulation. *J. Comput. Phys.*, 73:325–348, 1987.
- [15] W. Hackbusch and S. Börm. Approximation of Boundary Element Operators by Adaptive \mathcal{H}^2 -Matrices. *Applied Numerical Mathematics* 43:129–143, 2002.
- [16] W. Hackbusch. A Aparse Matrix Arithmetic Based on \mathcal{H} -Matrices. Part I: Introduction to \mathcal{H} -Matrices. *Computing*, 64:89–108, 1999.
- [17] S. Heubach and T. Mansour. *Combinatorics of Compositions and Words*. CRC Press, 2009.

- [18] W. Hackbusch. *Integral Equations: Theory and Numerical Treatment*. vol. 120 of International series of numerical mathematics. Birkhäuser, Basel, 1995.
- [19] W. Hackbusch and Z.P. Nowak. On the Fast Matrix Multiplication in the Boundary Element Method by Panel Clustering. *Numer. Math.*, 54:463–491, 1989.
- [20] H. Harbrecht and M. Randrianarivony. Wavelet BEM on Molecular Surfaces. Parametrization and implementation. *Computing*, 86(1):1–22, 2009.
- [21] H. Harbrecht and M. Randrianarivony. From Computer Aided Design to Wavelet BEM. *Comput. Vis. Sci.*, 13(2):69–82, 2010.
- [22] H. Harbrecht and M. Randrianarivony. Wavelet BEM on Molecular Surfaces. Solvent Excluded Surfaces. *Computing*, 92(4):335–364, 2011.
- [23] H. Harbrecht and R. Schneider. Biorthogonal Wavelet Bases for the Boundary Element Method. *Math. Nachr.*, 269–270:167–188, 2004.
- [24] H. Harbrecht and R. Schneider. Wavelet Galerkin Schemes for Boundary Integral Equations. Implementation and Quadrature. *SIAM J. Sci. Comput.*, 27(4):1347–1370, 2006.
- [25] H. Harbrecht and R. Stevenson. Wavelets with Patchwise Cancellation Properties. *Math. Comput.*, 75(256):1871–1889, 2006.
- [26] J. Hoschek and D. Lasser. *Grundlagen der Geometrischen Datenverarbeitung*. Teubner, Stuttgart, 1989.
- [27] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement *Comput. Methods Appl. Mech. Engrg.* 194:4135–4195, 2005.
- [28] S. Jaffard. Wavelet Methods for Fast Resolution of Elliptic Equations. *SIAM J. Numer. Anal.*, 29:965–986, 1992.
- [29] G. Schmidlin, C. Lage and C. Schwab. Rapid Solution of First Kind Boundary Integral Equations in \mathbf{R}^3 . *Eng. Anal. Bound. Elem.*, 27(5):469–490, 2003.
- [30] T. von Petersdorff and C. Schwab. Wavelet Approximation for First Kind Integral Equations on Polygons. *Numer. Math.*, 74:479–519, 1996.

- [31] T. Sauer and Y. Xu. On Multivariate Lagrange Interpolation. *Math. Comp.*, 64:1147–1170, 1995.
- [32] S. Sauter and C. Schwab. *Boundary Element Methods*. Springer, Berlin-Heidelberg, 2010.
- [33] R. Schneider. *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur Lösung Großer Vollbesetzter Gleichungssysteme*. Teubner, Stuttgart, 1998.
- [34] O. Steinbach. *Numerical Approximation Methods for Elliptic Boundary Value Problems. Finite and Boundary Elements*. Springer, New York, 2008.