

Hierarchical Matrix Techniques for Partial Differential Equations with Random Input Data

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie

vorgelegt der

Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Jürgen Dölz

aus

Riehen, Basel-Stadt

Basel, 2017

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Helmut Harbrecht
Prof. Dr. Steffen Börm

Basel, den 19.9.2017

Prof. Dr. Martin Spiess
Dekan

Technical knowledge is not enough. One must transcend techniques so that the art becomes an artless art, growing out of the unconscious.

Daisetsu Teitaro Suzuki
Japanese Author

Acknowledgements

I would like to use this opportunity to acknowledge the contributions of several people to this thesis. My sincerest gratitude goes to Prof. Dr. Helmut Harbrecht. I appreciate his supportive and careful guidance throughout the last years and the time he put into our extensive discussions about my mathematical and non-mathematical problems. I recognize him not only as a mentor, but also as a big-hearted person of trust. Many thanks are also due to Prof. Dr. Steffen Börm for being the co-referee of this thesis.

I further would like to thank my collaborators. I have to thank Dr. Michael Peters, who contributed to three of the publications which this thesis is based on. Especially at the beginning of my thesis, he took the time to share his mathematical knowledge with me. I also want to thank Prof. Dr. Christoph Schwab for the collaboration on the publication concerning the regularity of the solution of correlation equations. Although it is not directly part of this thesis, it provides the theoretical background for the \mathcal{H} -matrix approximation of the solutions.

Financially, thanks are due to the Swiss National Science Foundation (SNSF), which has supported my PhD through the project “ \mathcal{H} -matrix based first and second moment analysis”.

My parents Reinhard and Ute raised me to believe in myself and reminded me to trust in my capabilities whenever it was necessary. Thank you for everything you have done for me.

The familial and supportive atmosphere of the research group of Prof. Dr. Helmut Harbrecht has made my time as a PhD student very enjoyable. I would like to thank Monica Bugeanu, Ilja Kalmykov, Dr. Gianna Mitrou, Dr. Michael Peters, Marc Schmidlin, Dr. Markus Siebenmorgen, Dennis Tröndle, Dr. Manuela Utzinger, and Dr. Peter Zaspel for many cheerful lunches and dinners. Special thanks go to Marc Schmidlin, who proofread this thesis.

Finally, I want to thank my girlfriend Rahel Brügger. Her emotional support and patience throughout my time as a PhD student were endless, and her presence is more than mathematics can ever give me.

Contents

I	Introduction	1
II	Preliminaries	7
II.1	Tensor Products	7
II.1.1	Tensor Products of Hilbert Spaces	7
II.1.2	Tensor Products of Linear Operators	9
II.2	Function Spaces	9
II.2.1	Smooth Functions	10
II.2.2	L^p -spaces	11
II.2.3	Sobolev Spaces	12
II.2.4	Sobolev Spaces on Manifolds	14
II.2.5	Sobolev Spaces on Product Domains	15
II.3	Statistical Quantities of Interest	16
II.4	Correlation Equations for Linear Operators	18
II.5	Hierarchical Matrices	20
II.5.1	Asymptotical Smoothness	20
II.5.2	Cluster Tree	21
II.5.3	Block-Cluster Tree	23
II.5.4	Asymptotical Smoothness of Correlations	26
II.6	\mathcal{H} -matrix Arithmetic	29
II.6.1	Addition of \mathcal{H} -matrices	29
II.6.2	Multiplication of \mathcal{H} -matrices	31
II.6.3	\mathcal{H} -matrix Factorizations	34
III	PDEs with Random Load	39
III.1	Problem Formulation	39
III.2	\mathcal{H} -matrices in the Context of Finite Elements	41
III.2.1	Weak Admissibility	41
III.2.2	Nested Dissection	42
III.3	Iterative Solution	45
III.4	Numerical Examples	46
III.4.1	Experimental Setup	46
III.4.2	Tests for the Iterative Solver	47
III.4.3	Small Correlation Lengths	49
III.5	Conclusion	53
IV	The Fast \mathcal{H}^2-multipole Method on Parametric Surfaces	57
IV.1	Boundary Integral Equations	57
IV.2	Surface Representation	59
IV.3	Problem Formulation	62
IV.4	Galerkin Discretization	64
IV.5	Fast Multipole Method	65
IV.5.1	Kernel Interpolation	66

IV.5.2	Computational Complexity	67
IV.5.3	Nested Cluster Bases	68
IV.5.4	Error Estimates	71
IV.6	Higher Order Continuous Ansatz Functions	71
IV.7	Numerical Examples	73
IV.7.1	Regularization of the Hypersingular Operator	74
IV.7.2	Convergence	75
IV.7.3	Convergence on a More Complex Geometry	76
IV.7.4	Computational Cost and Accuracy	79
IV.8	Conclusion	81
V	PDEs with Random Dirichlet Data	83
V.1	Problem Formulation	83
V.2	\mathcal{H} -matrix Arithmetic for Parametric Surfaces	84
V.3	Numerical Examples	86
V.3.1	Tests for the \mathcal{H} -matrix Arithmetic	87
V.3.2	Tests for the Iterative Solver	88
V.3.3	Stochastic Application	91
V.4	Conclusion	93
VI	PDEs on Random Domains	95
VI.1	Random Domains	95
VI.1.1	Basic Definitions	95
VI.1.2	Shape Calculus for Parametrized Domains	96
VI.1.3	Statistical Moments on Random Domains	97
VI.2	Boundary Integral Equations	98
VI.3	Galerkin Discretization	99
VI.3.1	Dirichlet-to-Neumann Map	99
VI.3.2	Computation of $\text{Cor}[\delta u]$	100
VI.3.3	Computation of $\mathbb{E}[\delta^2 u]$	101
VI.4	Error Estimates	101
VI.4.1	Preliminaries	102
VI.4.2	Approximation Error of $\text{Cor}[\delta u]$	103
VI.4.3	Approximation Error of $\mathbb{E}[\delta^2 u]$	104
VI.5	Hierarchical Matrix Compression	107
VI.6	Numerical Examples	108
VI.6.1	Convergence in h	108
VI.6.2	Convergence in ε	110
VI.6.3	Non-smooth Boundaries	110
VI.7	Conclusion	112
VII	Final Remarks	115

Chapter I

Introduction

The numerical solution of strongly elliptic *linear partial differential equations* (PDEs) is an important task in science and engineering. Many physical phenomena can be modelled in this framework, which we may formulate as

$$\mathcal{L}u = f \quad \text{in } D \tag{I.0.1}$$

for some domain or manifold D , some differential operator \mathcal{L} , some load f , the unknown u , and, if D has a boundary, some boundary conditions.

Depending on the properties of \mathcal{L} , f , D , and the boundary conditions, the numerical solution of such equations is nowadays well understood, provided that all properties of these input data are known. However, this might be an assumption which is too strong in practical applications. One may think of measurement errors in physical constants or tolerances in production processes. Even worse, the uncertainty in the input data propagates into the solution u such that a highly accurate numerical solution of (I.0.1) is of limited use, as the influence of the uncertainty on the solution is unknown. Including the uncertainty into the PDE, one may rewrite (I.0.1) as

$$\mathcal{L}(\omega)u(\omega) = f(\omega) \quad \text{in } D(\omega), \tag{I.0.2}$$

and in some cases boundary conditions depending on ω , for some random parameter ω in some probability space $(\Omega, \Sigma, \mathbb{P})$. In particular, the solution u can then be modelled as a random field.

Quantifying the behavior of u depending on the probability distribution of the input data is thus an important task in science and engineering. In recent years, the need for efficient methods has led to the emerging field of *uncertainty quantification*. Current methods in uncertainty quantification can be categorized into sampling based methods and fully deterministic methods.

Sampling based methods in uncertainty quantification originate from the Monte-Carlo method, cf., e.g., [Caf98, Pro95] and the references therein, which provides the most simple solution to the problem. The Monte-Carlo method relies on the probability distribution of the input data and evaluates the PDE for some randomly drawn samples according to this distribution. The solutions of these samples can then be used to compute statistical output functionals of the solution u . However, the convergence of the Monte-Carlo method with respect to the number of samples can only be guaranteed in the root mean square error, with a convergence rate proportional to the root of the number of drawn samples. To improve computational efficiency, multi-level Monte-Carlo schemes, cf., e.g., [BSZ11, Gil15, HPS12a, HPS16a] and the references therein, can be used as variance reduction methods. Under some smoothness assumptions, one can replace the randomly drawn sample points by deterministically chosen points to obtain quasi-Monte-Carlo schemes, which provide deterministic error bounds and a higher convergence rate, see, e.g., the survey [DKS13] and the references therein. Nevertheless, the problem remains the same: a large number of PDE solutions must be computed to be able to quantify the uncertainty.

Deterministic methods in uncertainty quantification aim at a purely deterministic quantification of the uncertainty based on the stochastic description of the input data. For instance, random loads have been considered in [ST03, vPS06], random coefficients in [BNT07, BTZ04, DBO01, FST05, GS03, KS11, MK05, NTW08], and random domains in [HSS08b, XT06]. Deterministic methods can be divided into stochastic discretization methods and perturbation methods.

The idea of *stochastic discretization methods*, cf. [BNT07, BTZ04, GS03], is to cast the stochastic problem into a deterministic high-dimensional problem and to employ independent discretization methods for the spatial variable and the stochastic variable. Since the dimensionality of the stochastic variable is usually higher than the spatial variable, it is common to employ standard discretization methods in the spatial variable and more advanced discretization methods in the stochastic variable. Suitable methods for the stochastic variable are, for example, polynomial chaos, cf. [DNP+04, FST05, MK05, XK02], and sparse grid methods, cf. [BG04b, HPS12b, HPS16b, ST03, vPS06].

The idea of *perturbation methods*, cf. [BP02, Har10b, HSS08b], is to expand the solution of the PDE into a Taylor expansion with respect to the stochastic variable around its mean. The necessary derivatives for this approach can be obtained by solving additional subproblems, which can be derived mathematically using the Fréchet derivative. As the expansion is generally truncated after a small number of terms, perturbation methods are usually only suitable for small perturbations.

Beside the mean

$$\mathbb{E}[u](\mathbf{x}) := \int_{\Omega} u(\omega, \mathbf{x}) \, d\mathbb{P}(\omega)$$

and the variance

$$\mathbb{V}[u](\mathbf{x}) := \mathbb{E}[u^2](\mathbf{x}) - \mathbb{E}[u](\mathbf{x})^2$$

of the solution u , the correlation

$$\text{Cor}[u](\mathbf{x}, \mathbf{y}) := \int_{\Omega} u(\omega, \mathbf{x})u(\omega, \mathbf{y}) \, d\mathbb{P}(\omega)$$

and the covariance

$$\text{Cov}[u](\mathbf{x}, \mathbf{y}) := \text{Cor}[u](\mathbf{x}, \mathbf{y}) - \mathbb{E}[u](\mathbf{x})\mathbb{E}[u](\mathbf{y})$$

are frequent quantities of interest. In particular, the covariance and the correlation give direct access to the variance due to

$$\mathbb{E}[u^2](\mathbf{x}) = \text{Cor}[u](\mathbf{x}, \mathbf{x})$$

and

$$\mathbb{V}[u](\mathbf{x}) = \text{Cov}[u](\mathbf{x}, \mathbf{x}).$$

Often, and we will discuss several such examples in this thesis, the PDE (I.0.2) can be reformulated as a linear operator equation

$$\mathcal{L}u(\omega) = f(\omega) \quad \text{in } D.$$

Given the mean $\mathbb{E}[f]$ and the correlation $\text{Cor}[f]$ of f , the mean and the correlation of u can be expressed as the solutions of fully deterministic problems

$$\mathcal{L}\mathbb{E}[u] = \mathbb{E}[f] \quad \text{in } D$$

and

$$(\mathcal{L} \otimes \mathcal{L}) \text{Cor}[u] = \text{Cor}[f] \quad \text{in } D \times D. \tag{I.0.3}$$

$\mathbb{E}[u]$ can thus be computed easily whenever the deterministic equation $\mathcal{L}u = f$ can be solved. We also remark that $\text{Cor}[u]$ is the solution to a *linear* problem. Thus, instead of computing the non-linear term $\mathbb{E}[u^2]$ of u to compute $\mathbb{V}[u]$, one may rather choose to compute $\text{Cor}[u]$.

However, since (I.0.3) is posed on the product domain $D \times D$, the dimensionality of the problem squares and the solution is more involved. Therefore, correlation equations have been the topic of several articles.

In [HL13, HPS12a], a low-rank factorization of $\text{Cor}[f]$ was employed to efficiently compute the correlation. The existence of an accurate low-rank approximation is directly related to the spectral decomposition of the associated integral operator

$$(\mathcal{K}_f \psi)(\mathbf{x}) := \int_D \text{Cor}[f](\mathbf{x}, \mathbf{y}) \psi(\mathbf{y}) \, d\mu(\mathbf{y}).$$

Let $\text{Cor}[f] \in H^p(D) \otimes H^p(D)$ and n the dimension of D , then, according to [GH17, DHS17], the eigenvalues of this operator decay like

$$\lambda_m \lesssim m^{-2p/n-1} \text{ as } m \rightarrow \infty. \quad (\text{I.0.4})$$

Thus, if more Sobolev smoothness of $\text{Cor}[f]$, i.e., a higher value of p , is available, we can aim for a better low-rank approximation. On the other hand, no good low-rank approximation may exist for small values of p and the solution of (I.0.3) with low-rank approximations becomes computationally inefficient.

Additionally, the constant in the decay estimate (I.0.4) behaves like the $H^p(D) \otimes H^p(D)$ -norm of $\text{Cor}[f]$. The following consideration shows that this can lead to large constants in the decay estimate if the correlation length ℓ is small. Let the correlation kernel $k(r)$ depend only on the distance $r = \|\mathbf{x} - \mathbf{y}\|$. Then, the derivatives $\partial_{\mathbf{x}}^{\alpha} \text{Cor}[f](\mathbf{x}, \mathbf{y})$ and $\partial_{\mathbf{y}}^{\alpha} \text{Cor}[f](\mathbf{x}, \mathbf{y})$ of the correlation

$$\text{Cor}[f](\mathbf{x}, \mathbf{y}) = k\left(\frac{\|\mathbf{x} - \mathbf{y}\|}{\ell}\right)$$

involve the factor $\ell^{-|\alpha|}$, leading to a constant ℓ^{-2p} in the decay estimate of the eigenvalues. Thus, for a small correlation length, a low-rank approximation of $\text{Cor}[f]$ becomes prohibitively expensive to compute.

Other approaches to tackle the solution of correlation equations have been considered in several articles and are mostly based on a sparse tensor product, i.e., a sparse grid, discretization of the solution. For example, the computation of the second moment, i.e., $\text{Cor}[u]$, has been considered for elliptic diffusion problems with random loads in [ST03] by means of a sparse tensor product finite element method. A sparse tensor product wavelet boundary element method has been used in [HSS08b] to compute the solution's second moment of elliptic potential problems on random domains. In [Har10a, HSS08a], the computation of the second moment was done by sparse multilevel finite element frames. Recently, this concept has been simplified by using the combination technique, cf. [HPS13]. Unfortunately, the sparse tensor product discretization needs to resolve the concentrated measure for short correlation lengths. This means that the number of hierarchies of the involved finite element spaces has to be doubled if the correlation length is halved to get the same accuracy, which might be computationally infeasible.

Summarizing, common methods for the solution of correlation equations fail when the prescribed data correlation is “rough”. That is, if the data have low Sobolev smoothness or are shortly correlated.

These kinds of prescribed correlations shall be addressed with this thesis. We shall therefore develop a different approach, which exploits the fact that common correlation kernels behave similarly to fundamental solutions of elliptic PDEs. While this similarity was already used in [HPS15, KLM09, ST06] for the data sparse representation of correlation matrices, we will exploit this property further. In particular, we shall employ the technique of hierarchical matrices [Hac99, HK00a, HK00b] to represent densely populated correlation matrices by *block-wise* low-rank matrices. A special feature of these matrices is that an efficient arithmetic is available, which will enable us to solve correlation equations in almost linear, i.e., linear up to (poly-)logarithmic terms, time with respect to the dimension of the used finite element space.

The remainder of this thesis is structured as follows.

The following Chapter II is dedicated to the theoretical background for this thesis. We recall important facts about the tensor product of general Hilbert spaces and summarize the notion of Sobolev spaces on domains, manifolds, product domains, and product manifolds. A rather general discussion on statistical quantities and correlation equations will allow us to quantify the regularity of solutions throughout the following chapters. The chapter will conclude with the Galerkin discretization of correlation equations and an introduction to hierarchical matrices and their arithmetic. Especially, we recall the main result of [DHS17], which states that the solutions of correlation equations can, under certain conditions, also be approximated by hierarchical matrices.

Chapter III will provide a first example with PDEs on domains with stochastic load and is based on [DHP17]. We discuss how a black-box finite element solver connects to the framework of hierarchical matrices and how one can exploit this connection to solve correlation equations in almost linear time. In particular, we recall that the LU factorization of a finite element matrix can efficiently be represented using hierarchical matrices, see [Beb07, FMP15]. Several techniques to exploit the sparsity of finite element matrices exist in the hierarchical matrix literature, see [HKK04, GKL09]. We recall the main concepts and show that they can directly be included into the solution algorithm. Extensive numerical experiments verify the convergence of the presented solver. Different combinations of the discussed concepts will be employed to compare the computational time and solution accuracy. The second part of the numerical experiments is concerned with different kinds of Sobolev smoothness and correlation lengths to demonstrate the robustness of the hierarchical matrix approach under these parameters.

Chapter IV is based on [DHP16] and dedicated to the solution of homogeneous PDEs with non-trivial boundary conditions using boundary integral equations. Therefore, we assume that the underlying differential operator provides a fundamental solution, which is, for example, the case for the Laplace equation, the Helmholtz equation, and the heat equation. The major advantage of considering boundary integral equations is the reduction of the problem's dimensionality, with the disadvantage of generally leading to non-local boundary integral operators. Beside collocation and Nyström methods, the boundary element method is commonly used for the numerical discretization of such operators, see [Hac95, SS11, Ste08] and the references therein. Due to the non-locality, one usually ends up with large and densely populated system matrices and, thus, the numerical solution of such problems is rather challenging.

Motivated by the concept of isogeometric analysis [HCB05], we extend the fast multipole method on parametric surfaces from [HP13] to higher order boundary elements, which especially enables the discretization of the hypersingular operator. Therefore, in contrast to the usual practice in the hierarchical matrix literature, we propose an element-wise clustering strategy. The special structure of parametric surfaces allows for an algorithmically easy treatment of an improved version of hierarchical matrices, i.e., \mathcal{H}^2 -matrices, cf. [Bör10]. We especially remark that the introduced fast multipole method is not restricted to the discretization of integral operators of PDEs, but can also be applied to discretize correlation kernels or more general integral operators, as long as the kernel function satisfies certain properties. This will be a crucial property in the following two chapters.

The numerical experiments validate the higher convergence rates from the boundary element theory and provide a computational time to accuracy study. Numerical experiments for a complex non-smooth geometry are also included.

Chapter V is concerned with correlation equations of PDEs with random Dirichlet data. Using an approach with boundary integral equations, this chapter is historically the beginning of the hierarchical matrix approach to correlation equations and was investigated in [DHP15]. It has led to the subsequent development of the regularity theory of [DHS17], which gives a theoretical justification for the approach not only for boundary integral equations, but for general pseudodifferential operators.

We put the original article [DHP15] into the context of the regularity theory of [DHS17] and the subsequent developments in [DHP17]. In particular, compared to the original solution

algorithm we also use an LU factorization instead of the inverse. Although one could use a black-box boundary element solver for the solution of the correlation equation, we stay in the framework of parametric surfaces from the previous chapter and develop an arithmetic of hierarchical matrices tailored to these surfaces. In the numerical experiments, we repeat the convergence studies from [DHP15] also with higher order boundary elements and non-tensor product data.

Chapter VI is concerned with the treatment of PDEs on random domains and based on [DH17]. Since the dependence of the PDE's solution on the domain is nonlinear, we cannot directly apply the hierarchical matrix approach here. While domain mapping methods, as considered in [CCNT16, HPS16a, XT06], aim at modelling large deformations and come in connection with stochastic discretization or Monte-Carlo methods, they struggle with high computational cost if the domain variations are assumed to be small. In this case, perturbation methods are an attractive alternative, cf. [CPT15, HL13, HP15, HSS08b]. The derivatives necessary for the Taylor expansion can be computed by means of *shape calculus* as used in shape optimization, cf. [DZ01, SZ92]. Both approaches were combined recently in [CCNT17].

Additionally to the existing third order accurate expansion in the domain perturbation amplitude for the solution's correlation from [HSS08b], we derive third order accurate expansions for the mean, which are based on an additional correlation equation. We discuss that these expansions become even fourth order accurate when the law of the boundary variations is of a specific kind. A full convergence analysis of the corresponding Galerkin discretization is given. Finally, in the numerical experiments, we compute the solution's mean and correlation with the hierarchical matrix approach and verify the derived convergence rates for different orders of ansatz functions. We also discuss how non-smooth domains can be treated within the smoothness assumptions needed for the expansions.

Finally, in Chapter VII, we briefly review the findings of this thesis and discuss possible future work.

Throughout this thesis, in order to avoid the repeated use of generic but not further specified constants, it is implied by $C \lesssim D$ that C can be bounded by a multiple of D , independently of other parameters which C and D may depend on. Obviously, $C \gtrsim D$ is defined as $D \lesssim C$ and we write $C \sim D$ if $C \lesssim D$ and $C \gtrsim D$.

Chapter II

Preliminaries

II.1 Tensor Products

The correlation equations we are going to consider in this thesis are given in the product of domains with a tensor product operator. Since the natural spaces for the variational formulation are tensor products of Hilbert spaces, we will first recapitulate how the unique tensor product of two Hilbert spaces can be constructed, and then proceed to the action of tensor product operators on these spaces. The construction of the tensor product of Banach and Hilbert spaces was described at the latest by [Sch50].

II.1.1 Tensor Products of Hilbert Spaces

We shall give the precise construction of the tensor product of separable Hilbert spaces as presented in [KR83], that is, we uniquely extend, up to isomorphism, the notion of bilinear forms on Hilbert spaces. Let therefore H_1 and H_2 be two Hilbert spaces with inner products $(\cdot, \cdot)_{H_1}$ and $(\cdot, \cdot)_{H_2}$ and orthonormal bases \mathcal{B}_1 and \mathcal{B}_2 and recall that the existence of these orthonormal bases is equivalent to the Hilbert spaces being separable, see [Alt02, Theorem 7.8].

Definition II.1.1. *A bounded bilinear functional $b: H_1 \times H_2 \rightarrow \mathbb{R}$ is a Hilbert-Schmidt functional on $H_1 \times H_2$ if*

$$\sum_{\substack{\psi_1 \in \mathcal{B}_1 \\ \psi_2 \in \mathcal{B}_2}} |b(\psi_1, \psi_2)|^2 < \infty. \quad (\text{II.1.1})$$

The value of the sum (II.1.1) does not depend on the choice of the orthonormal bases \mathcal{B}_1 and \mathcal{B}_2 , cf. [KR83, Proposition 2.6.1]. The following lemma is [KR83, Proposition 2.6.2].

Lemma II.1.2. *The set of all Hilbert-Schmidt functionals \mathcal{HSF} on $H_1 \times H_2$ is itself a Hilbert space when the linearity and the inner product for elements $b, \tilde{b} \in \mathcal{HSF}$ are defined as follows:*

$$\begin{aligned} (\alpha b + \beta \tilde{b})(x_1, x_2) &= \alpha b(x_1, x_2) + \beta \tilde{b}(x_1, x_2), \\ (b, \tilde{b})_{\mathcal{HSF}} &= \sum_{\substack{\psi_1 \in \mathcal{B}_1 \\ \psi_2 \in \mathcal{B}_2}} b(\psi_1, \psi_2) \tilde{b}(\psi_1, \psi_2). \end{aligned} \quad (\text{II.1.2})$$

The sum in (II.1.2) is absolutely convergent, and the inner product does not depend on the choice of the orthonormal bases.

For $v_1, w_1 \in H_1$ and $v_2, w_2 \in H_2$, the expression

$$b_{v_1, v_2}(x_1, x_2) := (x_1, v_1)_{H_1} (x_2, v_2)_{H_2}$$

defines an element b_{v_1, v_2} of $\mathcal{H}\mathcal{H}\mathcal{F}$ with inner product

$$(b_{v_1, v_2}, b_{w_1, w_2})_{\mathcal{H}\mathcal{H}\mathcal{F}} = (v_1, w_1)_{H_1} (v_2, w_2)_{H_2}$$

and induced norm $\|\cdot\|_{\mathcal{H}\mathcal{H}\mathcal{F}} = \sqrt{(\cdot, \cdot)_{\mathcal{H}\mathcal{H}\mathcal{F}}}$. Moreover, the set $\{b_{\psi_1, \psi_2} : \psi_1 \in \mathcal{B}_1, \psi_2 \in \mathcal{B}_2\}$ is an orthonormal basis of $\mathcal{H}\mathcal{H}\mathcal{F}$.

Definition II.1.3. Let K be a Hilbert space with inner product $(\cdot, \cdot)_K$ and $b: H_1 \times H_2 \rightarrow K$ a continuous bilinear form. We call b a weak Hilbert-Schmidt mapping if

1. for each $u \in K$, the mapping

$$b_u(x_1, x_2) := (b(x_1, x_2), u)_K$$

is a Hilbert-Schmidt functional on $H_1 \times H_2$,

2. there is a real number C_b such that $\|b_u\|_{\mathcal{H}\mathcal{H}\mathcal{F}} \leq C_b \|u\|_K$ for all $u \in K$. We write $\|b\|_{\mathcal{H}\mathcal{H}\mathcal{F}} = C_b$ for the smallest possible C_b .

Having all ingredients at hand, the tensor product of two Hilbert spaces may be characterized by the following theorem, cf. [KR83, Theorem 2.6.4].

Theorem II.1.4. 1. There is a Hilbert space H and a weak Hilbert-Schmidt mapping $p: H_1 \times H_2 \rightarrow H$ with the following property: given any weak Hilbert-Schmidt mapping L from $H_1 \times H_2$ into a Hilbert space K , then there is a unique bounded linear mapping $T: H \rightarrow K$, such that $L = Tp$, i.e., the following diagram is commutative:

$$\begin{array}{ccc} H_1 \times H_2 & & \\ \downarrow p & \searrow L & \\ H & \xrightarrow{T} & K \end{array}$$

Moreover, it holds $\|T\|_{H \rightarrow K} = \|L\|_{\mathcal{H}\mathcal{H}\mathcal{F}}$.

2. If \tilde{H} and \tilde{p} have also the properties attributed in the previous statement to H and p , then there is a unitary transformation $U: H \rightarrow \tilde{H}$ such that $\tilde{p} = Up$.
3. If $v_1, w_1 \in H_1$ and $v_2, w_2 \in H_2$, then it holds

$$(p(v_1, v_2), p(w_1, w_2))_H = (v_1, w_1)_{H_1} (v_2, w_2)_{H_2},$$

$\|p\|_{\mathcal{H}\mathcal{H}\mathcal{F}} = 1$, and the set $\{p(\psi_1, \psi_2) : \psi_1 \in \mathcal{B}_1, \psi_2 \in \mathcal{B}_2\}$ is an orthonormal basis of H .

The second statement of the theorem guarantees the uniqueness of the Hilbert space H and the mapping p up to isomorphism.

Definition II.1.5. With the notation from the previous theorem, we define the tensor product of two separable Hilbert spaces as

$$H_1 \otimes H_2 := H,$$

and refer to p as the canonical mapping $H_1 \times H_2 \rightarrow H_1 \otimes H_2$. We write

$$x_1 \otimes x_2 := p(x_1, x_2)$$

and call this expression a simple tensor.

The third statement of Theorem II.1.4 provides several important properties of the tensor product space. For simple tensors, the tensor product is bilinear, the inner product of $H_1 \otimes H_2$ acts as

$$(v_1 \otimes v_2, w_1 \otimes w_2)_{H_1 \otimes H_2} = (v_1, w_1)_{H_1} (v_2, w_2)_{H_2},$$

and the norm is

$$\|v_1 \otimes v_2\|_{H_1 \otimes H_2} = \|v_1\|_{H_1} \|v_2\|_{H_2}. \quad (\text{II.1.3})$$

Simple tensors do not have to be linearly independent, see [KR83, Proposition 2.6.6], but the third statement of Theorem II.1.4 shows that the space of finite linear combinations of simple tensors, i.e.,

$$\mathring{H} := \left\{ \sum_{i,j=1}^m c_{ij} \psi_{1,i} \otimes \psi_{2,j} : \psi_{1,i} \in \mathcal{B}_1, \psi_{2,j} \in \mathcal{B}_2, m \in \mathbb{N} \right\}, \quad (\text{II.1.4})$$

is dense in $H_1 \otimes H_2$. In particular, \mathring{H} coincides with the *algebraic tensor product* and we can see the tensor product of separable Hilbert spaces $H_1 \otimes H_2$ as the completion of \mathring{H} with respect to its unique inner product

$$(v_1 \otimes v_2, w_1 \otimes w_2)_{\mathring{H}} = (v_1, w_1)_{H_1} (v_2, w_2)_{H_2},$$

cf. [KR83, Remark 2.6.7]. This is a useful property for density arguments. Especially, if H_1 and H_2 are two function spaces of functions with values in \mathbb{R} ,

$$p(v_1, v_2) := v_1 v_2$$

directly satisfies the conditions of the first statement of Theorem II.1.4. Thus, by the second statement of the theorem, we may assume

$$v_1 \otimes v_2 = v_1 v_2 \quad (\text{II.1.5})$$

in this case. This relation can also be seen by considering that p is an isomorphism between two orthonormal bases.

II.1.2 Tensor Products of Linear Operators

We shall define the action of bounded linear operators acting on tensor product spaces as presented in [LC85, Definition 1.29].

Definition II.1.6. *Let H_A, H_B, K_A, K_B be separable Hilbert spaces and*

$$A: H_A \rightarrow K_A, \quad B: H_B \rightarrow K_B$$

two bounded linear operators. Then, $A \otimes B: \mathring{H} \rightarrow K_A \otimes K_B$, with $\mathring{H} \subset H_A \otimes H_B$ defined as in (II.1.4), is defined as

$$(A \otimes B) \sum_{i,j=1}^m c_{ij} (\psi_{A,i} \otimes \psi_{B,j}) = \sum_{i,j=1}^m c_{ij} (A\psi_{A,i} \otimes B\psi_{B,j}).$$

Lemma II.1.7. *Using the notation from the previous definition, there exists a unique linear bounded extension of $A \otimes B$ to $H_A \otimes H_B$ such that*

$$\|A \otimes B\|_{H_A \otimes H_B \rightarrow K_A \otimes K_B} = \|A\|_{H_A \rightarrow K_A} \|B\|_{H_B \rightarrow K_B}.$$

Proof. The proof is a density argument and can be found in [LC85, Lemma 1.30 and 1.36]. \square

II.2 Function Spaces

In the following, we shall introduce the function spaces required for stating variational formulations and convergence rates. Therefore, we denote the euclidean inner product on \mathbb{R}^n by $\langle \cdot, \cdot \rangle$ and the associated norm by $\|\cdot\|$.

II.2.1 Smooth Functions

The definitions of this subsection are along the lines of [Alt02, Chapter 1], unless stated otherwise.

Definition II.2.1. Let $D \subset \mathbb{R}^n$ be open and bounded, Y a Banach space with norm $\|\cdot\|_Y$, and $k \in \mathbb{N}_0$. We define the space of k -times differentiable functions as

$$C^k(D; Y) := \{f: D \rightarrow Y: f \text{ is } k\text{-times continuous differentiable in } D \text{ and } \partial^\beta f \text{ can be continuously extended to } \bar{D} \text{ for all } |\beta| \leq k\}.$$

The C^k -spaces are Banach spaces, if they are equipped with the norm

$$\|f\|_{C^k(D; Y)} := \sum_{|\beta| \leq k} \sup_{\mathbf{x} \in D} \|\partial^\beta f(\mathbf{x})\|_Y.$$

Definition II.2.2. Let $D \subset \mathbb{R}^n$ and Y a Banach space with norm $\|\cdot\|_Y$. For $0 < \alpha \leq 1$, we say that a continuous function $f: D \rightarrow Y$ is Hölder continuous with exponent α , if there is some constant $C > 0$, such that

$$\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\|_Y \leq C \|\mathbf{x}_1 - \mathbf{x}_2\|^\alpha \quad \text{for all } \mathbf{x}_1, \mathbf{x}_2 \in D,$$

and denote by $\text{Höl}^\alpha(D; Y)$ the space of all Hölder continuous functions with exponent α . If a function is Hölder continuous with exponent $\alpha = 1$, we say that it is a Lipschitz continuous function.

Hölder spaces allow for a finer scale in comparison to the C^k -spaces.

Definition II.2.3. Let $D \subset \mathbb{R}^n$ be open and bounded and Y a Banach space with norm $\|\cdot\|_Y$. For $k \in \mathbb{N}_0$ and $0 < \alpha \leq 1$, we define

$$C^{k, \alpha}(D; Y) := \{f \in C^k(D; Y): \partial^\beta f \in \text{Höl}^\alpha(D; Y) \text{ for } |\beta| = k\}.$$

The $C^{k, \alpha}$ -spaces are Banach spaces, if they are equipped with the norm

$$\|f\|_{C^{k, \alpha}(D; Y)} := \|f\|_{C^k(D; Y)} + \sum_{|\beta|=k} \sup_{\substack{\mathbf{x}_1, \mathbf{x}_2 \in D \\ \mathbf{x}_1 \neq \mathbf{x}_2}} \frac{\|\partial^\beta f(\mathbf{x}_1) - \partial^\beta f(\mathbf{x}_2)\|_Y}{\|\mathbf{x}_1 - \mathbf{x}_2\|^\alpha}.$$

Definition II.2.4. For a set $D \subset \mathbb{R}^n$, a Banach space Y , and a function $f: D \rightarrow Y$, we define the support of f as

$$\text{supp } f := \overline{\{\mathbf{x} \in D: f(\mathbf{x}) \neq 0\}}^{\|\cdot\|},$$

i.e., as the closure in \mathbb{R}^n of the set where f attains non-zero values in D .

Definition II.2.5. Let Y a Banach space with norm $\|\cdot\|_Y$. On an open and bounded domain $D \subset \mathbb{R}^n$, we define the space of smooth functions as

$$C^\infty(D; Y) := \bigcap_{k \in \mathbb{N}} C^k(D; Y).$$

On an open and unbounded domain $D \subset \mathbb{R}^n$, we define $C^\infty(D; Y)$ as the space of functions which are in $C^\infty(\tilde{D}; Y)$ for every open and bounded subdomain \tilde{D} of D .

On an open domain $D \subset \mathbb{R}^n$, the space of smooth functions with compact support is defined as

$$C_0^\infty(D; Y) := \{f \in C^\infty(D; Y): \text{supp } f \Subset D\}.$$

If it holds $(Y, \|\cdot\|_Y) = (\mathbb{R}, |\cdot|)$, we simplify notation and write $C^k(D) = C^k(D; Y)$, $C^{k,\alpha}(D) = C^{k,\alpha}(D; Y)$, $C^\infty(D) = C^\infty(D; Y)$, and $C_0^\infty(D) = C_0^\infty(D; Y)$.

The following definition is along the lines of [McL01, Chapter 3].

Definition II.2.6. A partition of unity for an open set $D \subset \mathbb{R}^n$ is a finite or infinite sequence of functions $\{\psi_i\}_i$ in $C^\infty(\mathbb{R}^n)$ with the following properties:

1. It holds $\psi_i \geq 0$ on \mathbb{R}^n for all i .
2. Each point of D has a neighbourhood that intersects $\text{supp } \psi_i$ for only finitely many i .
3. It holds $\sum_i \psi_i(\mathbf{x}) = 1$ for all $\mathbf{x} \in D$.

Given any countable open cover $\{W_i\}_i$ of $D \subset \mathbb{R}^n$, there exists a partition of unity $\{\psi_i\}_i$ for D with $\text{supp } \psi_i \subset W_i$ for all i , see [McL01, Corollary 3.22]. In this case, we call $\{\psi_i\}_i$ subordinate to $\{W_i\}_i$.

We conclude this subsection by introducing the following class of functions, which quantifies the behaviour of functions in $C^\infty(D; Y)$ further, and was introduced in [Gev18].

Definition II.2.7. For an open domain $D \subset \mathbb{R}^n$ and a Banach space Y with norm $\|\cdot\|_Y$, we say that a function $f \in C^\infty(D; Y)$ is of Gevrey class $s \geq 1$, if for every $K \Subset D$ there exist constants $c, \mathcal{A} > 0$ such that for all $\alpha \in \mathbb{N}_0^n$

$$\|\partial^\alpha f(\mathbf{x})\|_Y \leq c \mathcal{A}^{|\alpha|} (\alpha!)^s \quad \text{for all } \mathbf{x} \in K.$$

We say that a function is analytic, if it is of Gevrey class $s = 1$.

II.2.2 L^p -spaces

Throughout this subsection, we shall briefly introduce L^p -spaces. The rather general fashion in terms of the Bochner integral, introduced in [Boc33], will later be helpful to specify different properties for the spatial and the stochastic variable. We proceed along the lines of [Alt02, Chapter 1].

Definition II.2.8. Let (X, \mathcal{B}, μ) be a measure space and Y a Banach space over \mathbb{R} with norm $\|\cdot\|_Y$. For a μ -measurable function $f: X \rightarrow Y$, we define the norms

$$\|f\|_{L_\mu^p(X; Y)} := \left(\int_X \|f(x)\|_Y^p d\mu(x) \right)^{1/p}, \quad 1 \leq p < \infty,$$

and

$$\|f\|_{L_\mu^\infty(X; Y)} := \text{ess sup}_{x \in X} \|f(x)\|_Y.$$

We define the L^p -spaces for $1 \leq p \leq \infty$ as

$$L_\mu^p(X; Y) := \{f: X \rightarrow Y: f \text{ } \mu\text{-measurable and } \|f\|_{L_\mu^p(X; Y)} < \infty\},$$

with the equivalence relation

$$f = g \text{ in } L_\mu^p(X; Y) \iff f = g \text{ } \mu\text{-almost everywhere.}$$

Moreover, we say that f is in $L_{\text{loc}, \mu}^p(X; Y)$, if $f \in L_\mu^p(U; Y)$ for all $U \Subset X$.

The L^p -spaces are Banach spaces under the given norm. If X is a subset of \mathbb{R}^n and μ is the Lebesgue measure, we write $L_\mu^p(X; Y) = L^p(X; Y)$. If additionally $(Y, \|\cdot\|_Y) = (\mathbb{R}, |\cdot|)$, we write $L_\mu^p(X; Y) = L^p(X)$.

If Y is a Hilbert space, the space $L_\mu^2(X; Y)$ is also a Hilbert space and its inner product is given by

$$(f, g)_{L_\mu^2(X; Y)} = \int_X (f(x), g(x))_Y d\mu(x).$$

In this case, we may also characterize the L_μ^2 -spaces as tensor product spaces, see [RS80, Theorem II.10b].

Theorem II.2.9. *Let $L^2_\mu(X)$ be a separable measure space and Y be a separable Hilbert space. It then holds*

$$L^2_\mu(X) \otimes Y \simeq L^2_\mu(X; Y).$$

The isomorphism $L^2_\mu(X) \otimes Y \rightarrow L^2_\mu(X; Y)$ is unique and given by the action

$$f(x) \otimes y \mapsto f(x)y$$

on simple tensors.

II.2.3 Sobolev Spaces

We shall now recall the definition of Sobolev spaces, which were originally introduced in [Co638]. After giving the definition of a weak derivative as in [AF03, Definition 1.62], we follow the presentation in [AF03, Chapter 3] to define the Sobolev spaces as a subset of the L^p -spaces $L^p(D)$ on $D \subset \mathbb{R}^n$.

Definition II.2.10. *Let $u \in L^1_{\text{loc}}(D)$ and $\alpha \in \mathbb{N}_0^n$. A function $v_\alpha \in L^1_{\text{loc}}(D)$ which satisfies*

$$\int_D u(\mathbf{x}) \partial^\alpha \varphi(\mathbf{x}) \, d\mathbf{x} = (-1)^{|\alpha|} \int_D v_\alpha(\mathbf{x}) \varphi(\mathbf{x}) \, d\mathbf{x} \quad \text{for all } \varphi \in C_0^\infty(D)$$

is called a weak derivative of u and is denoted by $\partial^\alpha u = v_\alpha$.

The weak derivative is unique (up to a set of measure zero) and coincides with the common classical derivative, if both of the two exist. Additionally, differentiation rules like the product rule and the chain rule also apply to the weak derivative, cf., e.g., [Alt02, Theorems 2.24 and 2.25].

Definition II.2.11. *For $k \in \mathbb{N}_0$ and $1 \leq p \leq \infty$, we define*

$$\begin{aligned} \|u\|_{W^{k,p}(D)} &= \left(\sum_{|\alpha| \leq k} \|\partial^\alpha u\|_{L^p(D)}^p \right)^{1/p} & \text{if } 1 \leq p < \infty, \\ \|u\|_{W^{k,\infty}(D)} &= \max_{|\alpha| \leq k} \|\partial^\alpha u\|_{L^\infty(D)} & \text{if } p = \infty, \end{aligned}$$

for any function for which the expression on the right-hand side is well defined.

Obviously, $\|\cdot\|_{W^{k,p}(D)}$ defines a norm for $k \in \mathbb{N}_0$ and $1 \leq p \leq \infty$ on any vector space of functions on which it is defined and has finite value.

Definition II.2.12. *For $k \in \mathbb{N}_0$ and $1 \leq p \leq \infty$, we define the Sobolev spaces*

$$W^{k,p}(D) = \{u \in L^p(D) : \partial^\alpha u \in L^p(D) \text{ for } |\alpha| \leq k\}$$

and

$$W_0^{k,p}(D) = \overline{C_0^\infty(D)}^{\|\cdot\|_{W^{k,p}(D)}},$$

i.e., $W_0^{k,p}(D)$ is the closure of $C_0^\infty(D)$ under the $W^{k,p}(D)$ -norm.

Again, both spaces are Banach spaces with respect to the $W^{k,p}(D)$ -norm and separable Hilbert spaces for $p = 2$ with respect to the inner product

$$(u, v)_{W^{k,2}(D)} = \sum_{|\alpha| \leq k} (\partial^\alpha u, \partial^\alpha v)_{L^2(D)},$$

cf. [AF03, Theorem 3.3 and 3.6].

It obviously holds $W^{0,p}(D) = L^p(D)$ and, since $C_0^\infty(D)$ is dense in $L^p(D)$ for $1 \leq p < \infty$, cf. [AF03, Corollary 2.30], it holds $W_0^{0,p}(D) = L^p(D)$. Moreover, we have the chain of embeddings

$$W_0^{k,p}(D) \subset W^{k,p}(D) \subset L^p(D).$$

Due to [MS64], it holds

$$W^{k,p}(D) = \overline{C^\infty(D) \cap W^{k,p}(D)}^{\|\cdot\|_{W^{k,p}(D)}}$$

for $1 \leq p < \infty$, i.e., $C^\infty(D) \cap W^{k,p}(D)$ is dense in $W^{k,p}(D)$.

The notion of Sobolev spaces can be extended to fractional orders of derivatives, which was investigated in several articles [Aro55, Gag58, Clo58]. We use the presentation from the survey [DNPV12].

Definition II.2.13. For non-integer $0 < s \in \mathbb{R}$ and $1 \leq p < \infty$, we define the norm

$$\|u\|_{W^{s,p}(D)} = \left(\|u\|_{W^{\lfloor s \rfloor, p}(D)}^p + |u|_{W^{s,p}(D)}^p \right)^{1/p},$$

where the semi-norm is given by

$$|u|_{W^{s,p}(D)} := \left(\sum_{|\alpha|=\lfloor s \rfloor} \int_D \int_D \frac{|\partial^\alpha u(\mathbf{x}) - \partial^\alpha u(\mathbf{y})|^p}{\|\mathbf{x} - \mathbf{y}\|^{n+p(s-\lfloor s \rfloor)}} d\mathbf{x} d\mathbf{y} \right)^{1/p}.$$

We then define the Sobolev spaces of fractional order $s > 0$ by

$$W^{s,p}(D) = \left\{ u \in W^{\lfloor s \rfloor, p}(D) : |u|_{W^{s,p}(D)} < \infty \right\}$$

and

$$W_0^{s,p}(D) = \overline{C_0^\infty(D)}^{\|\cdot\|_{W^{s,p}(D)}}.$$

As for the integer case, the fractional order Sobolev spaces are Banach spaces with respect to their norm. They are separable Hilbert spaces for $p = 2$ with respect to the inner product

$$\begin{aligned} (u, v)_{W^{s,2}(D)} &= (u, v)_{W^{\lfloor s \rfloor, 2}(D)} \\ &+ \sum_{|\alpha|=\lfloor s \rfloor} \int_D \int_D \frac{(\partial^\alpha u(\mathbf{x}) - \partial^\alpha u(\mathbf{y}))(\partial^\alpha v(\mathbf{x}) - \partial^\alpha v(\mathbf{y}))}{\|\mathbf{x} - \mathbf{y}\|^{n+2(s-\lfloor s \rfloor)}} d\mathbf{x} d\mathbf{y}, \end{aligned}$$

see [McL01, Chapter 3] and [Wlo87, Theorem 3.1]. Following [DNPV12], the notion of Sobolev spaces can also be extended to negative exponents.

Definition II.2.14. For $s < 0$ and $1 < p < \infty$, we define $W^{s,p}(D)$ as the dual of the space $W_0^{-s,q}(D)$ with respect to the pivot space $L^2(D)$, i.e., we define

$$W^{s,p}(D) = (W_0^{s,q}(D))'$$

with $1/p + 1/q = 1$.

Sobolev spaces with negative exponents are strictly speaking spaces of distributions, since they are the dual of spaces having $C_0^\infty(D)$ as a dense subset.

Definition II.2.15. For $0 \leq s \in \mathbb{R}$, we write

$$H^s(D) = W^{s,2}(D) \quad \text{and} \quad H_0^s(D) = W_0^{s,2}(D).$$

For $k \in \mathbb{N}_0$, we write

$$H^{-k}(D) = (H_0^k(D))'$$

Remark II.2.16. In the literature, $H^s(D)$ is sometimes also used to denote the Bessel potential space with exponent $s \in \mathbb{R}$, cf., e.g., [McL01, Chapter 3]. This notation coincides with the definition above, if $D = \mathbb{R}^n$ or its boundary is of Lipschitz class, see [McL01, Theorem 3.16, 3.30 and 3.33], which will be fulfilled for all domains in this thesis.

II.2.4 Sobolev Spaces on Manifolds

We shall extend the domain of definition of Sobolev spaces to the boundary of domains. The range of possible Sobolev exponents will crucially depend on the smoothness of the boundary. The following definition is inspired by [McL01, Chapter 3].

Definition II.2.17. *The open set $D \subset \mathbb{R}^n$, $n \geq 2$, is a $C^{k,\alpha}$ -domain if its boundary ∂D is compact and if there exist finite families $\{W_i\}_i$ and $\{D_i\}_i$, having the following properties:*

1. *The family $\{W_i\}_i$ is an open cover of ∂D , i.e., $W_i \subset \mathbb{R}^n$ is an open subset and $\partial D \subset \bigcup_i W_i$.*
2. *Each D_i can be transformed to a $C^{k,\alpha}$ -hypograph by a rigid motion, i.e., there exist $\tilde{D}_i \subset \mathbb{R}^{n-1}$ and $\phi_i \in C^{k,\alpha}(\tilde{D}_i)$, such that each D_i can be mapped into*

$$\{\mathbf{x} \in \tilde{D}_i \times \mathbb{R} : x_n < \phi_i(x_1, \dots, x_{n-1})\}$$

using rotations and translations.

3. *The equality $W_i \cap D = W_i \cap D_i$ is satisfied for all i .*

Epecially, we say that D is a Lipschitz domain, if it is a $C^{0,1}$ -domain. We say that it is a smooth domain, if it is a C^∞ -domain, and we call it a domain of Gevrey class $s \geq 1$, if the parametrizations ϕ_i are of Gevrey class s .

It follows directly from the definition that for all $C^{k,\alpha}$ -domains there exist mappings $\Phi_i: \mathbb{R}^{n-1} \rightarrow \mathbb{R}^n$ of class $C^{k,\alpha}$ such that $\partial D_i \subset \Phi_i(\mathbb{R}^{n-1})$. These mappings can be used to lift the definition of Sobolev spaces to the boundary of domains.

Definition II.2.18. *By denoting surface measure on ∂D by σ , we write, in accordance with Definition II.2.8,*

$$L^2(\partial D) = L^2_\sigma(\partial D)$$

for any Lipschitz domain $D \subset \mathbb{R}^n$.

The following definition is along the lines of [McL01, Chapter 3].

Definition II.2.19. *Let $k \in \mathbb{N}$ and D be a $C^{k-1,1}$ -domain. Using the notation from this subsection, choose a partition of unity $\{\psi_i\}_i$ subordinate to the open cover $\{W_i\}_i$ of ∂D . We then define the space*

$$H^s(\partial D) = \{u \in L^2(\partial D) : (\psi_i u) \circ \Phi_i \in H^s(\mathbb{R}^{n-1}) \text{ for all } i\}$$

for $0 < s \leq k$, and equip it with the inner product

$$(u, v)_{H^s(\partial D)} = \sum_i ((\psi_i u) \circ \Phi_i, (\psi_i v) \circ \Phi_i)_{H^s(\mathbb{R}^{n-1})}.$$

For $-k \leq s < 0$, we define the space

$$H^s(\partial D) = \left(H^{-s}(\partial D) \right)',$$

which we equip with the dual norm

$$\|u\|_{H^s(\partial D)} = \sup_{\substack{v \in H^{-s}(\partial D) \\ v \neq 0}} \frac{(u, v)_{L^2(\partial D)}}{\|v\|_{H^{-s}(\partial D)}}.$$

We especially note that the defined Sobolev spaces are, up to equivalence of norms, independent of the used parametrization, see [McL01, Chapter 3].

Since the boundary of a domain in \mathbb{R}^n has Lebesgue measure zero, the restrictions of functions in Sobolev spaces to domain boundaries are a priori not well defined. The following theorem, cf. [Wlo87, Theorem 8.7a], provides a meaningful extension of the restriction operator.

Theorem II.2.20. Let $D \subset \mathbb{R}^n$ be a $C^{k,\alpha}$ -domain and $1/2 < l \leq k + \alpha$. For $l \in \mathbb{N}$, we may allow $k = l - 1$ and $\alpha = 1$. Then, there exists a unique continuous linear trace operator

$$\gamma_0: H^l(D) \rightarrow H^{l-1/2}(\partial D),$$

with the property

$$\gamma_0 \varphi = \varphi|_{\partial D},$$

for all $\varphi \in C^l(\overline{D})$, if $l \in \mathbb{N}$, and for all $\varphi \in C^{[l]+1}(\overline{D})$, if $l \notin \mathbb{N}$, respectively.

An extension can also be made for the normal-derivative of functions, cf. [Wlo87, Theorem 8.7b].

Theorem II.2.21. Let $D \subset \mathbb{R}^n$ be a $C^{k,\alpha}$ -domain, $l - m > 1/2$, $m \in \mathbb{N}$, and $l + 1 \leq k + \alpha$. For $l \in \mathbb{N}$, we may allow again $k = l - 1$ and $\alpha = 1$. Then, there exists a unique continuous linear trace operator

$$\gamma_m: H^l(D) \rightarrow \bigtimes_{i=0}^m H^{l-i-1/2}(\partial D),$$

with the property

$$\gamma_m \varphi = \left(\varphi|_{\partial D}, \frac{\partial \varphi}{\partial \mathbf{n}} \Big|_{\partial D}, \frac{\partial^2 \varphi}{\partial \mathbf{n}^2} \Big|_{\partial D}, \dots, \frac{\partial^m \varphi}{\partial \mathbf{n}^m} \Big|_{\partial D} \right),$$

for all $\varphi \in C^{l+m}(\overline{D})$, if $l \in \mathbb{N}$, and for all $\varphi \in C^{[l]+m+1}(\overline{D})$, if $l \notin \mathbb{N}$, respectively. Here, we denote by $\partial/\partial \mathbf{n}$ the derivative in the direction of the outward pointing normal.

II.2.5 Sobolev Spaces on Product Domains

To simplify notation, we shall introduce the *Sobolev spaces of dominant mixed derivatives*.

Definition II.2.22. For a domain $D \subset \mathbb{R}^n$, we define for $0 \leq s, t \in \mathbb{R}$ the spaces

$$H_{\text{mix}}^{s,t}(D \times D) = H^s(D) \otimes H^t(D),$$

and for $s, t \in \mathbb{R}$ we define the spaces

$$H_{\text{mix}}^{s,t}(\partial D \times \partial D) = H^s(\partial D) \otimes H^t(\partial D),$$

provided that ∂D is sufficiently regular.

We readily remark that the dual space of $H_{\text{mix}}^{s,t}(\partial D \times \partial D)$ with respect to the pivot space $L^2(\partial D \times \partial D)$ is given by

$$(H_{\text{mix}}^{s,t}(\partial D \times \partial D))' = H_{\text{mix}}^{-s,-t}(\partial D \times \partial D),$$

see [Wlo87, Chapter 17.1] and [RS80, Theorem II.10a].

Let us further elaborate on the relations between $H_{\text{mix}}^{s,s}(M \times M)$ and $H^s(M \times M)$ with M being either a domain or its boundary. For $s = 0$, we have, cf. [RS80, Theorem II.10a],

$$H_{\text{mix}}^{0,0}(M \times M) = L^2(M) \otimes L^2(M) = L^2(M \times M) = H^0(M \times M). \quad (\text{II.2.1})$$

The situation is different for $s > 0$. For example, for $s \in \mathbb{N}$, we have due to (II.2.1), the alternate characterizations

$$H_{\text{mix}}^{s,s}(M \times M) = \left\{ f \in L^2(M \times M) : \partial_{\mathbf{x}}^{\alpha} \partial_{\mathbf{y}}^{\beta} f \in L^2(M \times M) \text{ for all } |\alpha|, |\beta| \leq s \right\}$$

and

$$H^s(M \times M) = \left\{ f \in L^2(M \times M) : \partial_{\mathbf{x}}^{\alpha} \partial_{\mathbf{y}}^{\beta} f \in L^2(M \times M) \text{ for all } |\alpha| + |\beta| \leq s \right\}.$$

For $s \in \mathbb{N}$, we thus have the important relation

$$H^s(M \times M) \subsetneq H_{\text{mix}}^{s,s}(M \times M) \subsetneq H^{2s}(M \times M),$$

see also Figure II.1 for an illustration.

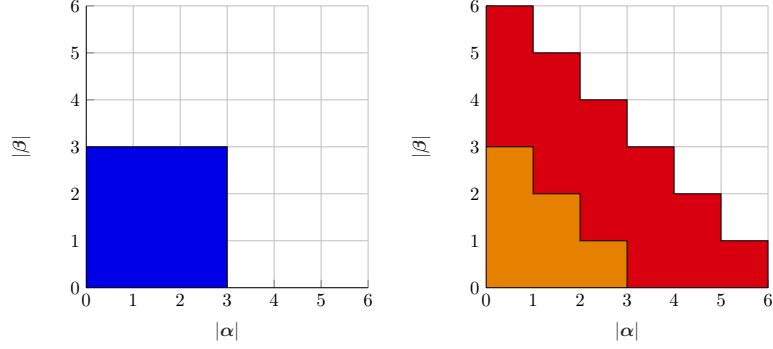


Figure II.1: Available derivatives $\partial_{\mathbf{x}}^{\alpha}$ and $\partial_{\mathbf{y}}^{\beta}$ for $f \in H_{\text{mix}}^{3,3}(M \times M)$ (left) and for $f \in H^3(M \times M)$ and $f \in H^6(M \times M)$ (right).

II.3 Statistical Quantities of Interest

For a given, separable and complete probability space $(\Omega, \Sigma, \mathbb{P})$ and separable Hilbert spaces H, H_1 , and H_2 , this section shall be concerned with statistical quantities of random fields.

Definition II.3.1. *An element of $L_{\mathbb{P}}^2(\Omega; H)$ is called a random field on H .*

We are mainly interested in the mean, the correlation, and the covariance of random fields. A direct application is that, in practical applications, random fields are often assumed to be Gaussian random fields, see [RW05], which are uniquely specified by their mean and covariance, cf. [RW05, Chapter 2.2].

Definition II.3.2. *Let H be a function space on X . We say that a random field $h \in L_{\mathbb{P}}^2(\Omega; H)$ is a Gaussian random field, if for all $k \in \mathbb{N}$, $x_1, \dots, x_k \in X$, the random variable $(h(x_1), \dots, h(x_k))$ has a multivariate Gaussian distribution.*

Recall that the *mean* of a real valued random variable $\psi \in L_{\mathbb{P}}^2(\Omega)$ is given as

$$\mathbb{E}[\psi] = \int_{\Omega} \psi(\omega) \, d\mathbb{P}(\omega)$$

and the *correlation* of two real valued random variables $\psi, \phi \in L_{\mathbb{P}}^2(\Omega)$ is given as

$$\text{Cor}[\psi, \phi] = \int_{\Omega} \psi(\omega)\phi(\omega) \, d\mathbb{P}(\omega).$$

We shall next turn our attention to random fields on more general spaces and define the mean and the correlation in terms of tensor notation.

Definition II.3.3. *We define the mean of a simple tensor $\psi \otimes f \in L_{\mathbb{P}}^2(\Omega) \otimes H$ as*

$$\mathbb{E}[\psi \otimes f] = \mathbb{E}[\psi]f \in H.$$

Moreover, we define the correlation of two simple tensors $\psi \otimes f \in L_{\mathbb{P}}^2(\Omega) \otimes H_1$ and $\phi \otimes g \in L_{\mathbb{P}}^2(\Omega) \otimes H_2$ as

$$\text{Cor}[\psi \otimes f, \phi \otimes g] = \text{Cor}[\psi, \phi](f \otimes g) \in H_1 \otimes H_2.$$

Due to the Cauchy-Schwarz inequality, the representation (II.1.3) for the tensor product norm for simple tensors, and

$$\begin{aligned} \|\mathbb{E}[\psi \otimes f]\|_H &= \|(1, \psi)_{L_{\mathbb{P}}^2(\Omega)}\| \|f\|_H \\ &\leq \|1\|_{L_{\mathbb{P}}^2(\Omega)} \|\psi\|_{L_{\mathbb{P}}^2(\Omega)} \|f\|_H \\ &= \|\psi \otimes f\|_{L_{\mathbb{P}}^2(\Omega) \otimes H}, \end{aligned}$$

the mean is a bounded linear operator on simple tensors, whereas an analogous argument shows that the correlation is a bounded bilinear operator on simple tensors.

Given an orthonormal basis $\{\psi_i\}_i$ of $L_{\mathbb{P}}^2(\Omega)$, Theorem II.1.4 guarantees the existence of expansions

$$f = \sum_{i=1}^{\infty} \psi_i \otimes f_i \quad (\text{II.3.1})$$

for all $f \in L_{\mathbb{P}}^2(\Omega) \otimes H \simeq L_{\mathbb{P}}^2(\Omega; H)$. A density argument yields the following lemma, see also [ST06].

Lemma II.3.4. *The mean and the correlation can be uniquely extended to a bounded linear operator*

$$\mathbb{E}: L_{\mathbb{P}}^2(\Omega; H) \rightarrow H$$

and a bounded bilinear operator

$$\text{Cor}: L_{\mathbb{P}}^2(\Omega; H_1) \otimes L_{\mathbb{P}}^2(\Omega; H_2) \rightarrow H_1 \otimes H_2.$$

Moreover, for expansions of the kind (II.3.1), it holds

$$\mathbb{E}[f] = \sum_{i=1}^{\infty} \mathbb{E}[\psi_i] f_i \quad \text{and} \quad \text{Cor}[f, g] = \sum_{i=1}^{\infty} f_i \otimes g_i.$$

Definition II.3.5. *The covariance of two functions $f, g \in L_{\mathbb{P}}^2(\Omega; H)$ is defined as*

$$\text{Cov}[f, g] = \text{Cor}[f, g] - \mathbb{E}[f] \otimes \mathbb{E}[g] \in H \otimes H.$$

For ease of notation, we write $\text{Cor}[f] := \text{Cor}[f, f]$ and $\text{Cov}[f] := \text{Cov}[f, f]$.

We note in particular that the mean, the correlation, and the covariance are well defined.

For the remainder of this section, we shall assume that H is some function space, i.e., the elements in $L_{\mathbb{P}}^2(\Omega; H)$ depend on some random variable and some additional parameter. Due to (II.1.5), one can see that the definitions of the mean, the correlation, and the covariance coincide with the common definitions for real valued functions, i.e.,

$$\mathbb{E}[f](\mathbf{x}) = \int_{\Omega} f(\omega, \mathbf{x}) \, d\mathbb{P}(\omega)$$

in case of the mean,

$$\text{Cor}[f, g](\mathbf{x}, \mathbf{y}) = \int_{\Omega} f(\omega, \mathbf{x}) g(\omega, \mathbf{y}) \, d\mathbb{P}(\omega)$$

in case of the correlation, and

$$\text{Cov}[f, g](\mathbf{x}, \mathbf{y}) = \text{Cor}[f, g](\mathbf{x}, \mathbf{y}) - \mathbb{E}[f](\mathbf{x}) \mathbb{E}[g](\mathbf{y})$$

in case of the covariance.

Having these quantities available gives direct access to the diagonal of the correlation and the variance.

Definition II.3.6. *The diagonal of the correlation of $f \in L_{\mathbb{P}}^2(\Omega; H)$, with H being a function space, is defined as*

$$\mathbb{E}[f^2](\mathbf{x}) = \text{Cor}[f](\mathbf{x}, \mathbf{x}).$$

The variance is defined as

$$\mathbb{V}[f](\mathbf{x}) = \text{Cov}[f](\mathbf{x}, \mathbf{x}) = \mathbb{E}[f^2](\mathbf{x}) - \mathbb{E}[f](\mathbf{x})^2.$$

In contrast to the mean, the correlation, and the covariance, the diagonal of the correlation and the variance of a random field in $L_{\mathbb{P}}^2(\Omega; H)$ do not necessarily belong to H .

II.4 Correlation Equations for Linear Operators

One of the main objects of investigation of this thesis are linear operator equations of the kind

$$Au(\omega) = f(\omega) \quad \text{on } M \quad \text{for } \mathbb{P}\text{-a.e. } \omega \in \Omega. \quad (\text{II.4.1})$$

We assume M to be either an open and bounded Lipschitz domain in \mathbb{R}^n or its boundary, and $f \in L^2_{\mathbb{P}}(\Omega; V')$ for some separable and complete probability space $(\Omega, \Sigma, \mathbb{P})$ and some Sobolev space V . To that end, assume that the linear operator $A: V \rightarrow V'$ gives rise to a bounded and strongly elliptic bilinear form $a: V \times V \rightarrow \mathbb{R}$ given by

$$a(u, v) := (Au, v)_{L^2(M)},$$

such that the solution of (II.4.1) is equivalent to solving the variational problem

$$\text{Find } u(\omega) \in V, \text{ such that } a(u(\omega), v) = (f(\omega), v)_{L^2(M)} \text{ for all } v \in V \text{ and } \mathbb{P}\text{-a.e. } \omega \in \Omega. \quad (\text{II.4.2})$$

Due to the Lax-Milgram Theorem, see [Alt02, Theorem 4.2], A is then invertible, and the solution $u(\omega)$ is uniquely determined for \mathbb{P} -almost every $\omega \in \Omega$.

Although the action of A in (II.4.1) is only defined for elements in V , we can obviously extend its mapping properties to $L^2_{\mathbb{P}}(\Omega; V) \simeq L^2_{\mathbb{P}}(\Omega) \otimes V$ such that the diagram

$$\begin{array}{ccc} L^2_{\mathbb{P}}(\Omega; V) & \xrightarrow{A} & L^2_{\mathbb{P}}(\Omega; V') \\ \downarrow \simeq & & \simeq \uparrow \\ L^2_{\mathbb{P}}(\Omega) \otimes V & \xrightarrow{\text{Id} \otimes A} & L^2_{\mathbb{P}}(\Omega) \otimes V' \end{array}$$

commutes. Thus, $u = A^{-1}f \in L^2_{\mathbb{P}}(\Omega; V)$ is again a random field. In particular, due to the fact that linear transformations of Gaussian random fields are again Gaussian, u is a Gaussian random field whenever f is a Gaussian random field. It is thus uniquely specified by its mean and covariance in this case.

By taking the mean on both sides of (II.4.1), the linearity of A yields

$$A\mathbb{E}[u] = \mathbb{E}[Au] = \mathbb{E}[f] \quad \text{on } M. \quad (\text{II.4.3})$$

Taking the correlation on both sides yields

$$(A \otimes A) \text{Cor}[u] = \text{Cor}[f] \quad \text{on } M \times M. \quad (\text{II.4.4})$$

Thus, if the mean and the correlation of the load are given, the mean and the correlation of the solution are determined according to purely deterministic problems.

Frequently used correlation kernels for the load are the *Matérn kernels*, cf. [RW05], which go back to [Mat60]. They are given by $\text{Cor}[f](\mathbf{x}, \mathbf{y}) = k_{\nu}(\|\mathbf{x} - \mathbf{y}\|)$ with

$$k_{\nu}(r) := \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{\ell} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}r}{\ell} \right), \quad (\text{II.4.5})$$

where $\ell, \nu \in (0, \infty)$ are some parameters. Here, K_{ν} denotes the modified Bessel function of the second kind, see [AS64]. For half integer values of ν , i.e., $\nu = p + 1/2$ for $p \in \mathbb{N}_0$, the expression simplifies to

$$k_{p+1/2}(r) = \exp \left(\frac{-\sqrt{2\nu}r}{\ell} \right) \frac{p!}{(2p)!} \sum_{i=0}^p \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu}r}{\ell} \right)^{p-i}.$$

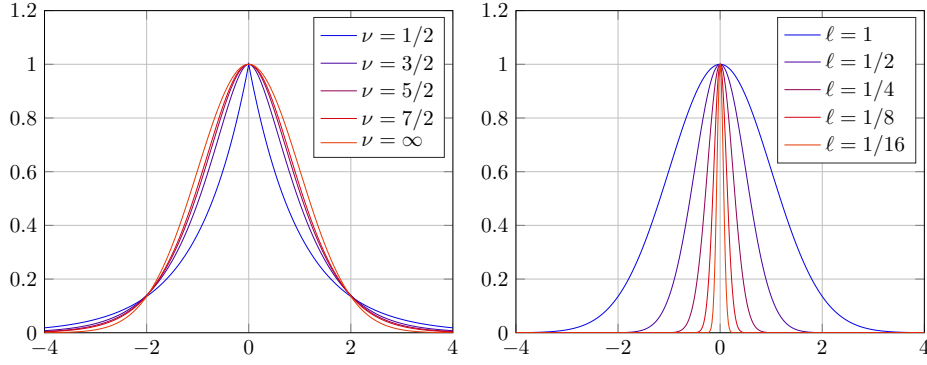


Figure II.2: Matérn kernels for $\ell = 1$ and different values of the smoothness parameter ν (left) and for $\nu = \infty$ and different correlation lengths ℓ (right).

In accordance with [RW05], we obtain in the limit case $\nu \rightarrow \infty$ the Gaussian kernel

$$k_{\infty}(r) = \exp\left(\frac{-r^2}{2\ell^2}\right).$$

The smoothness of the Matérn kernels is controlled by the *smoothness parameter* ν and the *correlation length* ℓ . A visualization of these kernels for varying values of ν is given in the left plot of Figure II.2 and for varying values of ℓ in the right plot of Figure II.2.

We shall now introduce a finite element space $V_h \subset V$ of dimension N with basis $\{\varphi_i\}_i$ and assume that the underlying mesh is quasi-uniform. For later reference, we shall also assume that the supports of the basis functions $\Upsilon_i = \text{supp } \varphi_i$ are local, that is, we require the existence of constants C_{sep} and n_{min} such that

$$\#\{j \in \{1, \dots, N\} : \text{dist}(\Upsilon_i, \Upsilon_j) \leq \text{diam}(\Upsilon_i)/C_{\text{sep}}\} \leq n_{\text{min}} \quad \text{for all } i = 1, \dots, N. \quad (\text{II.4.6})$$

We refer to [Hac15, Chapter 6.4.3.2] for a detailed discussion on this condition and remark that it is fulfilled for the usual local finite element basis functions of continuous piecewise polynomial spaces.

The variational problem of the equation for the mean (II.4.3) is analogous to (II.4.2), i.e.,

$$\text{Find } \mathbb{E}[u] \in V, \text{ such that } a(\mathbb{E}[u], v) = (\mathbb{E}[f], v)_{L^2(M)} \text{ for all } v \in V.$$

Employing a Galerkin scheme, i.e., replacing V by V_h in the variational problem, then yields the system of linear equations

$$\mathbf{A} \mathbf{e}_u = \mathbf{e}_f,$$

where

$$\mathbf{A} = [a(\varphi_j, \varphi_i)]_{i,j=1}^N, \quad \mathbf{e}_u = [e_i]_{i=1}^N, \quad \mathbf{e}_f = [(\mathbb{E}[f], \varphi_i)_{L^2(M)}]_{i=1}^N.$$

The Galerkin approximation $\mathbb{E}[u]_h$ of $\mathbb{E}[u]$ is then given by

$$\mathbb{E}[u]_h = \sum_{i=1}^N e_i \varphi_i.$$

The properties of this system of linear equations are well understood and we refer to [Gal15, Bra13, GW12] for more details on the Galerkin scheme.

The variational formulation of the correlation equation (II.4.4) is given by

Find $\text{Cor}[u] \in V \otimes V$ such that

$$\left((A \otimes A) \text{Cor}[u], v \right)_{L^2(M \times M)} = \left(\text{Cor}[f], v \right)_{L^2(M \times M)} \quad (\text{II.4.7})$$

for all $v \in V \otimes V$.

Obviously, since simple tensors are dense in $V \otimes V$ and since

$$\left((A \otimes A)(u_1 \otimes u_2), v_1 \otimes v_2 \right)_{L^2(M \times M)} = a(u_1, v_1)a(u_2, v_2) \quad \text{for all } u_1, u_2, v_1, v_2 \in V, \quad (\text{II.4.8})$$

the tensor product operator $A \otimes A$ implies a strongly elliptic bilinear form. Therefore, the correlation of the solution $\text{Cor}[u]$ is the unique solution to the variational problem (II.4.7). Moreover, by replacing again V with the finite element space V_h , the variational formulation (II.4.7) yields the system of linear equations

$$(\mathbf{A} \otimes \mathbf{A}) \text{vec}(\mathbf{C}_u) = \text{vec}(\mathbf{C}_f), \quad (\text{II.4.9})$$

where

$$\mathbf{C}_u = [c_{ij}]_{i,j=1}^N, \quad \mathbf{C}_f = [(\text{Cor}[f], \varphi_i \otimes \varphi_j)_{L^2(M \times M)}]_{i,j=1}^N.$$

As usual in connection with matrices, the tensor product is understood as the Kronecker product. For a matrix $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$, the operation $\text{vec}(\mathbf{B})$ is defined as

$$\text{vec}([\mathbf{b}_1, \dots, \mathbf{b}_n]) := \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} \in \mathbb{R}^{mn}.$$

The Galerkin approximation $\text{Cor}[u]_h$ of $\text{Cor}[u]$ is then given by

$$\text{Cor}[u]_h = \sum_{i,j=1}^N c_{ij}(\varphi_i \otimes \varphi_j).$$

Since for matrices $\mathbf{B} \in \mathbb{R}^{k \times n}$, $\mathbf{C} \in \mathbb{R}^{\ell \times m}$, and $\mathbf{X} \in \mathbb{R}^{m \times n}$, there holds the relation

$$(\mathbf{B} \otimes \mathbf{C}) \text{vec}(\mathbf{X}) = \text{vec}(\mathbf{C}\mathbf{X}\mathbf{B}^\top),$$

we may rewrite (II.4.9) as a matrix equation

$$\mathbf{A}\mathbf{C}_u\mathbf{A}^\top = \mathbf{C}_f. \quad (\text{II.4.10})$$

This alternate representation can also be seen when directly substituting (II.4.8) into the variational formulation (II.4.7).

Obviously, when using a non-local kernel function $\text{Cor}[f]$, e.g., one of the Matérn kernels, the matrix \mathbf{C}_f is densely populated. Thus, and since the matrix equation (II.4.10) has N^2 unknowns, the matrix equation is not directly solvable if N is large due to memory and time consumption. For further reference, the uncompressed version of this equation is referred to as the *full tensor product approach*. This reflects the fact that we take all N^2 ansatz functions of $V_h \otimes V_h$ into account.

II.5 Hierarchical Matrices

The main objective of this thesis is to employ an alternate compression technique for the full tensor product approach, other than sparse tensor product or low-rank approaches. Therefore, we will exploit different analytic properties of correlation kernels.

II.5.1 Asymptotical Smoothness

In the following, we will restrict ourselves to *asymptotically smooth* correlation kernels $\text{Cor}[f]$, i.e., correlation kernels satisfying the following definition, see also [Hac15, Definition 4.14].

Definition II.5.1. Let $M_{\mathbf{x}}, M_{\mathbf{y}} \subset \mathbb{R}^n$ such that $k: M_{\mathbf{x}} \times M_{\mathbf{y}} \rightarrow \mathbb{R}$ is defined and arbitrarily often differentiable for all $\mathbf{x} \neq \mathbf{y}$ with $\mathbf{x} \in M_{\mathbf{x}}$ and $\mathbf{y} \in M_{\mathbf{y}}$. The function k is called asymptotically smooth if it holds

$$|\partial_{\mathbf{x}}^{\alpha} \partial_{\mathbf{y}}^{\beta} k(\mathbf{x}, \mathbf{y})| \leq c_1 \frac{(|\alpha| + |\beta|)!}{c_2^{|\alpha| + |\beta|}} \|\mathbf{x} - \mathbf{y}\|^{-n-2q-|\alpha|-|\beta|}, \quad \mathbf{x} \in M_{\mathbf{x}}, \mathbf{y} \in M_{\mathbf{y}}, \mathbf{x} \neq \mathbf{y}, \quad (\text{II.5.1})$$

independently of α and β for some constants $c_1, c_2 > 0$ and $q \in \mathbb{R}$.

The term $2q$ usually reflects the order of the pseudodifferential operator corresponding to k , on which we will comment later in this section. Examples for asymptotically smooth correlation kernels are the Matérn kernels, discussed in Chapter II.4, and kernels of boundary integral operators, which will be discussed in Chapter IV.1. A main feature of asymptotically smooth functions is that they exhibit a data-sparse representation by means of *hierarchical matrices* (\mathcal{H} -matrices for short), which were introduced in a series of articles [Hac99, HK00a, HK00b]. They are a generalization of cluster techniques for the rapid solution of boundary integral equations such as the panel clustering method [HN89], the fast multipole method [GR87], the mosaic skeleton approximation [Tyr96], and the adaptive cross approximation [Beb00], and rely on local low-rank approximations of a given matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$. That is, for suitable non-empty index sets $\tau, \sigma \subset \{1, \dots, N\}$, the matrix blocks $\mathbf{H}|_{\tau \times \sigma}$ are approximated by matrices of rank k . This approximation can be represented in factorized form $\mathbf{H}|_{\tau \times \sigma} \approx \mathbf{L}\mathbf{R}^{\top}$ with factors $\mathbf{L} \in \mathbb{R}^{\tau \times k}$ and $\mathbf{R} \in \mathbb{R}^{\sigma \times k}$, which can be visualized as

$$\begin{array}{ccc} \boxed{\mathbf{H}|_{\tau \times \sigma}} & \approx & \boxed{\mathbf{L}} \quad \boxed{\mathbf{R}^{\top}} \end{array}$$

Hence, if $k \ll \min\{\#\tau, \#\sigma\}$, the complexity for storing the block is considerably reduced.

For the rest of this section, we recapitulate the main ingredients for the construction of hierarchical matrices. Their corresponding arithmetic will be discussed in the next section.

II.5.2 Cluster Tree

We follow the presentation of the monograph [Hac15, Chapter 5.3 and A.2].

Definition II.5.2. Let V be a non-empty finite set, call it vertex set, and let sons be a mapping from V into the power set $\mathcal{P}(V)$, i.e., sons: $V \rightarrow \mathcal{P}(V)$. For any $v \in V$, an element v' in sons(v) is called son, whereas we call v the father of v' .

We call the structure $T(V, \text{sons})$ a tree, if the following properties hold.

1. There is exactly one element $r \in V$ which is not a son of a vertex, i.e.,

$$\bigcup_{v \in V} \text{sons}(v) = V \setminus \{r\}.$$

We call this vertex the root of the tree.

2. All $v \in V$ are successors of r , i.e., there is a $k \in \mathbb{N}_0$, such that $v \in \text{sons}^k(r)$. We define $\text{sons}^k(v)$ recursively as

$$\text{sons}^0(v) = \{v\} \quad \text{and} \quad \text{sons}^k(v) = \bigcup_{v' \in \text{sons}^{k-1}(v)} \text{sons}(v').$$

3. Any $v \in V \setminus \{r\}$ has exactly one father.

Moreover, we say that the number k is the level of v . The depth of a tree is the maximum of the levels of its vertices. We define the set of leaves of T as

$$\mathcal{L}(T) = \{v \in V : \text{sons}(v) = \emptyset\}.$$

We remark that for any $v \in T$, there is exactly one path from r to v , see [Hac15, Remark A.6].

Next, we introduce the cluster tree, which provides a tree structured partitioning of finite index sets.

Definition II.5.3. Let \mathcal{I} be a finite index set. The cluster tree $\mathcal{T}_{\mathcal{I}}$ is a tree with the following properties.

1. $\mathcal{I} \in \mathcal{T}_{\mathcal{I}}$ is the root of the tree $\mathcal{T}_{\mathcal{I}}$,
2. for all $\tau \in \mathcal{T}_{\mathcal{I}} \setminus \mathcal{L}(\mathcal{T}_{\mathcal{I}})$ it holds

$$\dot{\bigcup}_{\sigma \in \text{sons}(\tau)} \sigma = \tau,$$

i.e., all non-leaf vertices are the disjoint union of their sons,

3. all $\tau \in \mathcal{T}_{\mathcal{I}}$ are non-empty.

The vertices of the cluster tree are referred to as clusters.

Remark II.5.4. The second requirement of the above definition together with the definition of a tree directly implies that a non-leaf cluster must have at least two sons. However, we will see in Chapter III.2.2 that in some cases it is reasonable to violate this requirement. Several ways to rigorously deal with this obstacle were suggested in the literature, see also [Hac15, Remark 5.11]. However, to keep notation simple and since it will be rather straightforward how to deal with this situation in our case, we will not further elaborate on this subject.

Obviously, by applying the second requirement of the definition recursively, it holds $\tau \subset \mathcal{I}$ for all $\tau \in \mathcal{T}_{\mathcal{I}}$. We thus have

$$\mathcal{I} = \dot{\bigcup}_{\sigma \in \mathcal{L}(\mathcal{T}_{\mathcal{I}})} \sigma. \quad (\text{II.5.2})$$

By using an index set corresponding to the basis functions of a finite dimensional ansatz or test space from a Galerkin discretization for the cluster tree, we directly introduce a hierarchy in the basis functions. Associating the basis functions with their support yields, together with the following definition, a hierarchical covering of the underlying domain.

Definition II.5.5. Let \mathcal{I} be the index set of an ansatz or test space with basis $\{\varphi_i\}_{i \in \mathcal{I}}$. The support Υ_{τ} of a cluster τ is defined as the union of the supports of its associated basis functions, i.e.,

$$\Upsilon_{\tau} := \bigcup_{i \in \tau} \Upsilon_i.$$

To achieve almost linear complexity for \mathcal{H} -matrix operations, we shall assume that the depth of the cluster tree is bounded by $\mathcal{O}(\log \#\mathcal{I})$. There are various ways to construct a cluster tree fulfilling this requirement along with different kinds of other properties, see [Hac15] and the references therein. We discuss a geometry-based construction via bounding boxes, following the presentation of [Hac15, Chapter 5.4.2]. For that purpose, we assign to each Υ_i , $i \in \mathcal{I}$, a characteristic point ξ_i , e.g., the barycenter of Υ_i or the Lagrangian interpolation point of the corresponding basis function φ_i if nodal finite element shape functions are considered. An *axis-parallel bounding-box* to a cluster $\tau \subset \mathcal{I}$ is constructed as listed in Algorithm II.1. A binary cluster tree, that is, each vertex has either two or no sons, can then be constructed with initial value $\tau = \mathcal{I}$ as shown in Algorithm II.2. The relabelling in each step guarantees that

Algorithm II.1 Construct an axis-parallel bounding-box, see [Hac15, Equation (5.19)]

function BUILDBOUNDINGBOX(cluster $\tau \neq \emptyset$) $\triangleright n$ is the space dimension of \mathbb{R}^n
for $i \in \{1, \dots, n\}$ **do**
 Set $a_i = \min_{j \in \tau} (\xi_j)_i$
 Set $b_i = \max_{j \in \tau} (\xi_j)_i$
end for
 Bounding-box is $\times_{i=1}^n [a_i, b_i]$
end function

Algorithm II.2 Construct a cluster tree, see [Hac15, Equations (5.24) and (5.27)]

function BUILDCLUSTERTREE(cluster $\tau \neq \emptyset$)
 $Q_\tau = \text{BUILDBOUNDINGBOX}(\tau)$
if $\#\tau \geq n_{\min}$ and $|Q_\tau| > 0$ **then**
 Cut Q_τ into two pieces Q_1 and Q_2 by halving the longest edge.
 Set $\tau_1 = \{i \in \tau : \xi_i \in Q_1\}$
 Set $\tau_2 = \tau \setminus \tau_1$
 Relabel indices of clusters in subsequent order: first τ_1 , then τ_2 .
 BUILDCLUSTERTREE(τ_1)
 BUILDCLUSTERTREE(τ_2)
 Set $\text{sons}(\tau) = \{\tau_1, \tau_2\}$
end if
end function

the indices in each cluster are numbered in a subsequent order. Following [Hac15, Remark 5.20], the depth of the constructed cluster tree $T_{\mathcal{I}}$ is bounded by

$$\text{depth}(T_{\mathcal{I}}) \leq n \lceil \log_2(\delta_{\max}/\delta_{\min}) \rceil,$$

where

$$\begin{aligned} \delta_{\min} &:= \min \{ \|\xi_i - \xi_j\|_\infty : i, j \in \mathcal{I} \text{ with } \xi_i \neq \xi_j \}, \\ \delta_{\max} &:= \max \{ \|\xi_i - \xi_j\|_\infty : i, j \in \mathcal{I} \}. \end{aligned}$$

For reasonably chosen characteristic points ξ_i , it follows from the quasi-uniformity of the mesh and (II.4.6) that the depth scales like $\mathcal{O}(\log \#\mathcal{I})$.

II.5.3 Block-Cluster Tree

Given a basis $\{\varphi_i\}_{i \in \mathcal{I}}$ of the ansatz and test space, the corresponding system matrices from the previous section can obviously be identified with the index set $\mathcal{I} \times \mathcal{I}$. Given a cluster tree $\mathcal{T}_{\mathcal{I}}$ on \mathcal{I} , a tree $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$ is given on $\mathcal{I} \times \mathcal{I}$ by setting

$$\text{root}(\mathcal{T}_{\mathcal{I} \times \mathcal{I}}) = \text{root}(\mathcal{T}_{\mathcal{I}}) \times \text{root}(\mathcal{T}_{\mathcal{I}}),$$

and defining the vertices $b = \tau \times \sigma \in \mathcal{T}_{\mathcal{I} \times \mathcal{I}}$, $\tau, \sigma \in \mathcal{T}_{\mathcal{I}}$, starting from the root, recursively as

$$\text{sons}(b) = \begin{cases} \{\tau' \times \sigma' : \tau' \in \text{sons}(\tau), \sigma' \in \text{sons}(\sigma)\}, & \text{sons}(\tau), \text{sons}(\sigma) \neq \emptyset, \\ \emptyset, & \text{otherwise.} \end{cases}$$

By the correspondence between clusters and subsets of the domain M , given in Definition II.5.5, a tree structure is defined on the product domain $M \times M$.

The following admissibility condition aims at using this structure to identify smooth and non-smooth parts of asymptotically smooth functions in the product domain $M \times M$.

Definition II.5.6. Two clusters τ and σ are called η -admissible if

$$\max\{\text{diam}(\Upsilon_\tau), \text{diam}(\Upsilon_\sigma)\} \leq \eta \text{ dist}(\Upsilon_\tau, \Upsilon_\sigma) \quad (\text{II.5.3})$$

holds for some fixed $\eta > 0$.

Different kinds of admissibility exist, see [Hac15] for a throughout discussion and further examples, and we will see two other examples in Chapter III. If not noted otherwise, we are referring to the above condition, whenever we mention admissibility.

The importance of the admissibility condition comes from the fact that asymptotically smooth correlations are known to have exponentially converging low-rank approximations in admissible cluster pairs and the corresponding matrix blocks, see, e.g., [Beb00, BG04a]. Naturally, in order to keep computational effort and storage requirements low, one aims at representing as large and as many matrix blocks as possible by low-rank approximations without sacrificing too much accuracy. One possible and efficient approach to achieve this goal is to apply the admissibility condition recursively as in Algorithm II.3 to identify the admissible cluster pairs. This yields the set \mathcal{F} , to which we refer to as the *farfield*, and the set \mathcal{N} , to which we refer to as the *nearfield*. The algorithm is called with the *block-cluster* $b = \mathcal{I} \times \mathcal{I}$.

Algorithm II.3 Construction of the block-cluster tree \mathcal{B} , cf. [Hac15, Definition 5.26]

```

function BUILD_BLOCK_CLUSTER_TREE(block-cluster  $b = \tau \times \sigma$ )
  if  $\tau$  and  $\sigma$  are admissible then
    sons( $b$ ) :=  $\emptyset$ 
    Add  $b$  to  $\mathcal{F}$ 
  else
    if  $\tau$  and  $\sigma$  have sons then
      sons( $b$ ) :=  $\{\sigma' \times \tau' : \tau' \in \text{sons}(\tau), \sigma' \in \text{sons}(\sigma)\}$ 
      for  $b' \in \text{sons}(b)$  do
        BUILD_BLOCK_CLUSTER_TREE( $b'$ )
      end for
    else
      Add  $b$  to  $\mathcal{N}$ 
    end if
  end if
end function
    
```

Definition II.5.7. The tree structure \mathcal{B} with root $\mathcal{I} \times \mathcal{I}$ constructed by Algorithm II.3 is called the block-cluster tree.

We shall mention at this point that, depending of the shape of the support of the involved clusters, the exact verification of (II.5.3) might be computationally expensive. Instead, one usually relies on axis-parallel bounding-boxes or similar techniques to achieve computational efficiency, see [Hac15, Chapter 5.4.6]. Moreover, in practical applications, one might stop the recursion when either $\#\tau < n_{\min}$ or $\#\sigma < n_{\min}$.

We collect some of the properties of the block-cluster tree, see also [Hac15, Theorem 5.27], and remark that our definition of the block-cluster tree coincides with the notion of a *level-conserving block-cluster tree* from the \mathcal{H} -matrix literature.

Theorem II.5.8. 1. For all $b \in \mathcal{B} \setminus \mathcal{L}(\mathcal{B})$, there holds

$$b = \bigcup_{b' \in \text{sons}(b)} b'.$$

2. All $b \in \mathcal{B}$ have the form $b = \tau \times \sigma$, with $\tau, \sigma \in \mathcal{T}_{\mathcal{I}}$, and it holds

$$\text{level}(b) = \text{level}(\sigma) = \text{level}(\tau).$$

3. The depth of the block-cluster tree is the depth of the cluster tree from which it is constructed.
4. The block-cluster tree has a symmetric pattern, that is, it holds $\tau \times \sigma \in \mathcal{B}$ if and only if $\sigma \times \tau \in \mathcal{B}$.

Obviously, since $\mathcal{I} \times \mathcal{I}$ is the root of \mathcal{B} , the first statement of the theorem, applied recursively, directly implies

$$\mathcal{I} \times \mathcal{I} = \dot{\bigcup}_{b \in \mathcal{L}(\mathcal{B})} b,$$

and, if we consider $\mathcal{L}(\mathcal{B}) = \mathcal{F} \cup \mathcal{N}$ as a union of sets, it also holds

$$\mathcal{I} \times \mathcal{I} = \dot{\bigcup}_{b \in \mathcal{F}} b \dot{\cup} \dot{\bigcup}_{b \in \mathcal{N}} b.$$

We have thus partitioned the index set $\mathcal{I} \times \mathcal{I}$ into disjoint sets which are either admissible or inadmissible and where the matrix blocks corresponding to admissible index sets can be represented as low-rank matrices.

Definition II.5.9. For a block-cluster $b = \tau \times \sigma$ and $k \leq \min\{\#\tau, \#\sigma\}$, we define the set of low-rank matrices as

$$\mathcal{R}(\tau \times \sigma, k) = \{\mathbf{M} \in \mathbb{R}^{\tau \times \sigma} : \text{rank}(\mathbf{M}) \leq k\},$$

where all elements $\mathbf{M} \in \mathcal{R}(\tau \times \sigma, k)$ are stored in low-rank representation, i.e.,

$$\mathbf{M} = \mathbf{L}_\mathbf{M} \mathbf{R}_\mathbf{M}^\top$$

for matrices $\mathbf{L}_\mathbf{M} \in \mathbb{R}^{\tau \times k}$ and $\mathbf{R}_\mathbf{M} \in \mathbb{R}^{\sigma \times k}$.

Obviously, a matrix in $\mathcal{R}(\tau \times \sigma, k)$ requires $k(\#\tau + \#\sigma)$ units of storage instead of $\#\tau \cdot \#\sigma$, which results in a significant storage improvement if $k \ll \min\{\#\tau, \#\sigma\}$. The same consideration holds true for the matrix-vector multiplication.

With the definition of the block-cluster tree at hand, we are in the position to introduce \mathcal{H} -matrices.

Definition II.5.10. Given a block-cluster tree \mathcal{B} , the set of hierarchical matrices, in short \mathcal{H} -matrices, of maximal block-rank k is given by

$$\mathcal{H}(\mathcal{B}, k) := \{\mathbf{H} \in \mathbb{R}^{N \times N} : \mathbf{H}|_{\tau \times \sigma} \in \mathcal{R}(\tau \times \sigma, k) \text{ for all } \tau \times \sigma \in \mathcal{F}\}.$$

A tree structure is induced on this set by the tree structure of the block cluster tree. Note that all nearfield blocks $\mathbf{H}|_{\tau \times \sigma}$, $\tau \times \sigma \in \mathcal{N}$, are allowed to be dense matrices.

The tree structure of the block-cluster tree provides the following useful recursive block matrix structure on \mathcal{H} -matrices. Every matrix block $\mathbf{H}|_{\tau \times \sigma}$, corresponding to a non-leaf block-cluster $\tau \times \sigma$, has the structure

$$\mathbf{H}|_{\tau \times \sigma} = \begin{bmatrix} \mathbf{H}|_{\text{sons}(\tau)_1 \times \text{sons}(\sigma)_1} & \cdots & \mathbf{H}|_{\text{sons}(\tau)_1 \times \text{sons}(\sigma)_{\#\text{sons}(\sigma)}} \\ \vdots & & \vdots \\ \mathbf{H}|_{\text{sons}(\tau)_{\#\text{sons}(\tau)} \times \text{sons}(\sigma)_1} & \cdots & \mathbf{H}|_{\text{sons}(\tau)_{\#\text{sons}(\tau)} \times \text{sons}(\sigma)_{\#\text{sons}(\sigma)}} \end{bmatrix}. \quad (\text{II.5.4})$$

If the matrix block $\mathbf{H}|_{\tau' \times \sigma'}$, $\tau' \in \text{sons}(\tau)$, $\sigma' \in \text{sons}(\sigma)$, is a leaf of \mathcal{B} , the matrix block is either a low-rank matrix (if $\tau' \times \sigma' \in \mathcal{F}$) or a dense matrix (if $\tau' \times \sigma' \in \mathcal{N}$). If the matrix block is not a leaf of \mathcal{B} , it has again a similar block structure as $\mathbf{H}|_{\tau \times \sigma}$. The required ordering of the clusters relies on the order of the indices in the clusters.

In accordance with [Hac15, Lemma 6.13], the storage cost of an \mathcal{H} -matrix $\mathbf{H} \in \mathcal{H}(\mathcal{B}, k)$ is $\mathcal{O}(kN \log N)$. The constant depends on the quantity $C_{\text{sp}}(\mathcal{B})$ which is given by

$$C_{\text{sp}}(\mathcal{B}) := \max_{\tau \in \mathcal{T}_{\mathcal{I}}} \#\{\sigma \in \mathcal{T}_{\mathcal{I}} : \tau \times \sigma \in \mathcal{B}\}$$

and was investigated in [GH03, Gra01, Hac15]. Although most situations are too complex for an in-depth investigation of $C_{\text{sp}}(\mathcal{B})$, the analysis shows that, on quasi-uniform meshes and when the clustering technique from Algorithm II.2 is used, it is reasonable to assume that $C_{\text{sp}}(\mathcal{B})$ can be bounded by a constant depending on η from the admissibility condition (II.5.6), C_{sep} from (II.4.6), and the spatial dimension. For asymptotically smooth correlation kernels, the rank k depends poly-logarithmically on the desired approximation accuracy ε , which in turn usually depends on the degrees of freedom N .

Having the block structure (II.5.4) available, an algorithm for the matrix-vector multiplication, as listed in Algorithm II.4, can easily be derived. Note that the matrix-vector

Algorithm II.4 \mathcal{H} -matrix-vector multiplication $\mathbf{y} += \mathbf{H}\mathbf{x}$, see [Hac15, Equation (7.1)]

```

function  $\mathcal{H}\text{TIMESV}(\mathbf{y}|_{\tau}, \mathbf{H}|_{\tau \times \sigma}, \mathbf{x}|_{\sigma})$ 
  if  $\tau \times \sigma \notin \mathcal{L}(\mathcal{B})$  then
    for  $\tau' \times \sigma' \in \text{sons}(\tau \times \sigma)$  do
       $\mathcal{H}\text{TIMESV}(\mathbf{y}|_{\tau'}, \mathbf{H}|_{\tau' \times \sigma'}, \mathbf{x}|_{\sigma'})$ 
    end for
  else
     $\mathbf{y}|_{\tau} += \mathbf{H}|_{\tau \times \sigma} \mathbf{x}|_{\sigma}$ 
  end if
end function
    
```

multiplication for the leaf block-clusters involves either a dense matrix or a low-rank matrix. In accordance with [Hac15, Lemma 7.17], the complexity of the matrix-vector multiplication for elements in $\mathcal{H}(\mathcal{B}, k)$ is $\mathcal{O}(kN \log N)$ where the hidden constant depends on $C_{\text{sp}}(\mathcal{B})$.

A criterion for the \mathcal{H} -matrix approximability of explicitly given correlation kernels, such as $\text{Cor}[f]$ in (II.4.4), is checking for asymptotical smoothness. While this is a calculation if the kernel is given explicitly, it is more involved when the correlation kernel is implicitly given, as is the case for $\text{Cor}[u]$ in (II.4.4), and topic of the next subsection.

II.5.4 Asymptotical Smoothness of Correlations

The compressibility of an implicitly given correlation kernel, such as $\text{Cor}[u]$ in (II.4.4), was studied in [DHS17] for the case of sufficiently smooth domains and boundaries M . We restate the main theorem, which employs the fact that the integral operator

$$(\mathcal{K}_{\text{Cor}[f]}\varphi)(\mathbf{x}) = \int_M \text{Cor}[f](\mathbf{x}, \mathbf{y})\varphi(\mathbf{y}) \, d\mu(\mathbf{y}),$$

related to the correlation kernel $\text{Cor}[f]$, is in general a pseudodifferential operator.

We recall the main concepts required for the statement from the literature, in particular, elements of the calculus of pseudodifferential operators and its analytic extension in [BdMK67, Kr69]. Especially, we refer to [Rod93] for a comprehensive account of this theory, including subsequent developments. We adopt the notation for the statements of results on pseudodifferential operators from the monographs [Hör07, HW08, Tay81] and, in the analytic case, from [Rod93]. The extensions to analytic and Gevrey-class pseudodifferential operators are based on the Gevrey extension of the symbolic calculus for classical pseudodifferential operators, developed in [BdMK67, Kr69, Rod93].

Definition II.5.11. *For an order $r \in \mathbb{R}$ and an open and bounded domain $D \subset \mathbb{R}^n$ with smooth boundary, the symbol class $S^r(D \times \mathbb{R}^n)$ consists of functions $a \in C^\infty(D \times \mathbb{R}^n)$ such*

that, for any $K \Subset D$ and for every $\alpha, \beta \in \mathbb{N}_0^n$, there exist constants $C_{\alpha\beta}(K) > 0$ such that

$$\left| \partial_{\mathbf{x}}^{\alpha} \partial_{\boldsymbol{\xi}}^{\beta} a(\mathbf{x}, \boldsymbol{\xi}) \right| \leq C_{\alpha\beta}(K) \langle \boldsymbol{\xi} \rangle^{r-|\beta|} \quad \text{for all } \mathbf{x} \in K, \boldsymbol{\xi} \in \mathbb{R}^n,$$

with $\langle \boldsymbol{\xi} \rangle = (1 + \|\boldsymbol{\xi}\|^2)^{1/2}$.

The class $S^r(D \times \mathbb{R}^n)$ is contained in the Hörmander class $S_{1,0}^r(D \times \mathbb{R}^n)$. We shall not require the general classes $S_{\rho,\delta}^r(D \times \mathbb{R}^n)$, see [Hör03, Definition 7.8.1], and omit the fine indices.

Definition II.5.12. For a symbol $a \in S^r(D \times \mathbb{R}^n)$, the corresponding pseudodifferential operator A is defined for $u \in C_0^\infty(D)$ via the oscillatory integral

$$A(\mathbf{x}, -i\partial_{\mathbf{x}})u(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}} \int_{\boldsymbol{\xi} \in \mathbb{R}^n} e^{i\langle \mathbf{x}, \boldsymbol{\xi} \rangle} a(\mathbf{x}, \boldsymbol{\xi}) \hat{u}(\boldsymbol{\xi}) \, d\boldsymbol{\xi}, \quad \mathbf{x} \in D, \quad (\text{II.5.5})$$

with \hat{u} being the Fourier transform of u . The set of all pseudodifferential operators A , generated via (II.5.5) from symbols $a \in S^r(D \times \mathbb{R}^n)$, is denoted by $OPS^r(D)$.

The following is [HW08, Definition 6.1.3].

Definition II.5.13. Given a symbol $a \in S^{m_0}(D \times \mathbb{R}^n)$ and a sequence of symbols $a_{m_\ell} \in S^{m_\ell}(D \times \mathbb{R}^n)$ with $m_\ell \in \mathbb{R}$ monotonically decreasing and $m_\ell \rightarrow -\infty$ as $\ell \rightarrow \infty$, we write

$$a \sim \sum_{\ell=0}^{\infty} a_{m_\ell}$$

and call the (formal) sum the asymptotic expansion of a , if

$$a - \sum_{\ell=0}^{k-1} a_{m_\ell} \in S^{m_k}(D \times \mathbb{R}^n) \quad \text{for all } k \in \mathbb{N}.$$

This allows to introduce the following subset of symbols in $S^r(D \times \mathbb{R}^n)$.

Definition II.5.14. A symbol $a \in S^r(D \times \mathbb{R}^n)$ is called classical symbol of order $r \in \mathbb{R}$ if for every $k \in \mathbb{N}_0$ there exist functions $a_{r-k} \in S^{r-k}(D \times \mathbb{R}^n)$ such that $a \sim \sum_k a_{r-k}$, where a_{r-k} is homogeneous of degree $r-k$, i.e., it holds that $a_{r-k}(\mathbf{x}, t\boldsymbol{\xi}) = t^{r-k} a_{r-k}(\mathbf{x}, \boldsymbol{\xi})$ for every $t > 0$ and for every $\boldsymbol{\xi} \in \mathbb{R}^n$ with $\|\boldsymbol{\xi}\| > 1$.

The class of classical symbols of order r is denoted by $S_{cl}^r(D \times \mathbb{R}^n)$. The set of all pseudodifferential operators A , generated from symbols $a \in S_{cl}^r(D \times \mathbb{R}^n)$, is denoted by $OPS_{cl}^r(D)$.

As a consequence of the asymptotic expansion of $a \in S_{cl}^r(D \times \mathbb{R}^n)$, for every $\alpha, \beta \in \mathbb{N}_0^n$ and for every $K \Subset D$ exists a constant $c_{\alpha\beta}(K) > 0$ such that for every $m \in \mathbb{N}_0$ holds

$$\left| \partial_{\mathbf{x}}^{\alpha} \partial_{\boldsymbol{\xi}}^{\beta} \left(a(\mathbf{x}, \boldsymbol{\xi}) - \sum_{k=0}^m a_{r-k}(\mathbf{x}, \boldsymbol{\xi}) \right) \right| \leq c_{\alpha\beta}(K) \langle \boldsymbol{\xi} \rangle^{r-m-|\beta|-1} \quad \text{for all } \mathbf{x} \in K, \boldsymbol{\xi} \in \mathbb{R}^n. \quad (\text{II.5.6})$$

For a connection with asymptotically smooth functions, we consider another subclass of symbols introduced in [BdMK67]. It is based on tighter control of dependence of $c_{\alpha\beta}(K)$ in (II.5.6). The following definition is [BdMK67, Definition 1.1].

Definition II.5.15. We say that $a \sim \sum_k a_{r-k} \in S_{cl}^r(D \times \mathbb{R}^n)$ is a Gevrey symbol of class $s \geq 1$ if for every $K \Subset D$ exist constants $c, \mathcal{A} > 0$ such that for all $\alpha, \beta \in \mathbb{N}_0^n$

$$\left| \partial_{\mathbf{x}}^{\alpha} \partial_{\boldsymbol{\xi}}^{\beta} a_{r-k}(\mathbf{x}, \boldsymbol{\xi}) \right| \leq c \mathcal{A}^{k+|\alpha+\beta|} \|\boldsymbol{\xi}\|^{r-k-|\beta|} (k + |\alpha|)!^s \beta! \quad \text{for all } \mathbf{x} \in K, \boldsymbol{\xi} \in \mathbb{R}^n. \quad (\text{II.5.7})$$

The subclass of $S_{cl}^r(D \times \mathbb{R}^n)$ which satisfies (II.5.7) is denoted by $S_{cl,s}^r(D \times \mathbb{R}^n)$. The set of all pseudodifferential operators A , generated from symbols $a \in S_{cl,s}^r(D \times \mathbb{R}^n)$, is denoted by $OPS_{cl,s}^r(D)$.

We remark that symbols in $S_{cl,s}^r(D \times \mathbb{R}^n)$ with $s = 1$ depend analytically on $\mathbf{x} \in D$ and continue by discussing pseudodifferential operators on smooth, closed and compact Riemannian n -manifolds $\mathcal{M} \subset \mathbb{R}^{n+1}$. As function spaces and operators on \mathcal{M} are defined via local coordinates in a suitable atlas of coordinate charts, the precise regularity of kernels of pseudodifferential operators on \mathcal{M} will depend on whether \mathcal{M} is C^∞ , Gevrey or analytic. The assumed compactness of \mathcal{M} implies that there exists a finite atlas of $(\mathcal{M}_i, \kappa_i)$ with smooth coordinate charts κ_i which parametrize \mathcal{M} . By κ , we denote a generic chart. Pseudodifferential operators on \mathcal{M} are defined locally on \mathcal{M} via charts κ with (generic) common compact parameter domain $D \subset \mathbb{R}^n$. We recall the definition from [Hör07, Definition 18.1.20].

Definition II.5.16. *On the smooth and compact Riemannian n -manifold $\mathcal{M} \subset \mathbb{R}^{n+1}$, a linear operator $A : C^\infty(\mathcal{M}) \rightarrow C^\infty(\mathcal{M})$ is a pseudodifferential operator of order r on \mathcal{M} if for every chart $\kappa : D \rightarrow \mathcal{M}$ the transported operator satisfies $A^\kappa := (\kappa^{-1})^* A \kappa^* \in OPS^r(D)$. We write $A \in OPS^r(\mathcal{M})$.*

We call $A \in OPS^r(\mathcal{M})$ classical if for each coordinate chart κ holds $A^\kappa \in OPS_{cl}^r(D)$. In this case, we write $A \in OPS_{cl}^r(\mathcal{M})$. In complete analogy, we define the set of Gevrey regular pseudodifferential operators of class $s \geq 1$ on \mathcal{M} as $OPS_{cl,s}^r(\mathcal{M})$, provided the charts of \mathcal{M} are of Gevrey class s .

The following is [Hör07, Definition 18.1.21].

Definition II.5.17. *A (pseudodifferential) operator A on a domain or manifold \mathcal{M} is said to be properly supported, if for every compact set $K \subset \mathcal{M}$ there is a compact set $K' \subset \mathcal{M}$ such that*

$$\text{supp } u \subset K \Rightarrow \text{supp } Au \subset K' \quad \text{and} \quad u = 0 \text{ on } K' \Rightarrow Au = 0 \text{ on } K.$$

We are now in the position to recall the main theorem of [DHS17].

Theorem II.5.18. *In either the euclidean domain D with analytic boundary ∂D or on a compact, analytic n -manifold $\mathcal{M} \subset \mathbb{R}^{n+1}$, assume that the covariance kernel $\text{Cor}[f]$ of $f \in L_{\mathbb{P}}^2(\Omega; V')$ in (II.4.1) gives rise to an operator $\mathcal{K}_{\text{Cor}[f]} \in OPS_{cl,s}^\theta$, i.e., to a classical pseudodifferential operator with symbol $a_{\text{Cor}[f]}(\mathbf{x}, \boldsymbol{\xi})$ of order θ and of Gevrey class $s \geq 1$. Assume further that the operator $A \in OPS_{cl,s}^r$ is properly supported. Then, the covariance kernel $\text{Cor}[u]$ of the random field solution u of (II.4.1) is the Schwartz kernel of an operator $\mathcal{K}_{\text{Cor}[u]} \in OPS_{cl,s}^{\theta-2r}$.*

Moreover, the kernel $\text{Cor}[u]$ is smooth outside of the diagonal (in local coordinates in case of a manifold) and there holds the pointwise estimate

$$|\partial_{\mathbf{x}}^\alpha \partial_{\mathbf{y}}^\beta \text{Cor}_u(\mathbf{x}, \mathbf{y})| \leq c \mathcal{A}^{|\alpha+\beta|} (|\alpha|!)^s \beta! \|\mathbf{x} - \mathbf{y}\|^{-n-(\theta-2r)-|\alpha|-|\beta|} \quad (\text{II.5.8})$$

for all $\alpha, \beta \in \mathbb{N}_0^d$, away from the diagonal (again, in local coordinates in case of a manifold) and some constants c and \mathcal{A} which depend only on D, \mathcal{M} , and on $a_{\text{Cor}[f]}$.

The main tool of the proof is the calculus of pseudodifferential operators of Gevrey class. We only recall the following relation from [BdMK67, Proposition 2.11] for further reference later in this thesis.

Theorem II.5.19. *For $s \geq 1$, $A \in OPS_{cl,s}^r$, and $B \in OPS_{cl,s}^t$, with at least one of A, B properly supported, implies $AB \in OPS_{cl,s}^{r+t}$.*

We refer to [DHS17] for a discussion how restrictive the assumption of A being properly supported is.

Obviously, for $s = 1$, estimate (II.5.8) directly implies condition (II.5.1) for the asymptotic smoothness of $\text{Cor}[u]$, allowing us to approximate $\text{Cor}[u]$ by means of \mathcal{H} -matrices. Note especially that $2q = \theta - 2r$ in this case. The case $s > 1$ yields the following weakened version of asymptotic smoothness, see also [Hac15, Definition 4.14].

Definition II.5.20. Let $M_{\mathbf{x}}, M_{\mathbf{y}} \subset \mathbb{R}^n$ such that $k: M_{\mathbf{x}} \times M_{\mathbf{y}} \rightarrow \mathbb{R}$ is defined and arbitrarily often differentiable for all $\mathbf{x} \neq \mathbf{y}$ with $\mathbf{x} \in M_{\mathbf{x}}$ and $\mathbf{y} \in M_{\mathbf{y}}$. The function k is called asymptotically smooth in one variable if it holds

$$|\partial_{\mathbf{x}}^{\alpha} k(\mathbf{x}, \mathbf{y})| \leq c_1 \frac{|\alpha|!}{c_2^{|\alpha|}} \|\mathbf{x} - \mathbf{y}\|^{-n-2q-|\alpha|}, \quad \mathbf{x} \in M_{\mathbf{x}}, \mathbf{y} \in M_{\mathbf{y}}, \mathbf{x} \neq \mathbf{y},$$

or

$$|\partial_{\mathbf{y}}^{\beta} k(\mathbf{x}, \mathbf{y})| \leq c_1 \frac{|\beta|!}{c_2^{|\beta|}} \|\mathbf{x} - \mathbf{y}\|^{-n-2q-|\beta|}, \quad \mathbf{x} \in M_{\mathbf{x}}, \mathbf{y} \in M_{\mathbf{y}}, \mathbf{x} \neq \mathbf{y},$$

independently of α and β for some constants $c_1, c_2 > 0$ and $q \in \mathbb{R}$.

Similar to asymptotically smooth functions, functions which are asymptotically smooth in one variable, are known to have exponentially converging low-rank approximations in admissible clusters, see, e.g., [Beb00, BG04a]. Still, we distinguish between the two definitions, since the approximation technique we are going to discuss in Chapter V will require the stronger version of the two, i.e., Definition II.5.1. We also remark that the Schwartz kernel of an arbitrary pseudodifferential operator of Gevrey class $s \geq 1$ and order r is asymptotically smooth or asymptotically smooth in one variable and satisfies Definition II.5.1 or II.5.20 with $2q = r$. This can either be seen from the proof of Theorem II.5.18 or from its assertion by setting $A = \text{Id}$.

An example of correlation kernels for $\text{Cor}[f]$, satisfying the condition of this theorem for $s \geq 1$, is the Matérn class of kernels from Chapter II.4, whose Fourier transform in \mathbb{R}^n was already computed in [Mat60]. Since the corresponding Gaussian random fields are stationary, it does not depend on the spatial variable \mathbf{x} and is given by

$$a_{\nu}(\boldsymbol{\xi}) = \alpha \left(1 + \frac{\ell^2}{2\nu} \|\boldsymbol{\xi}\|^2 \right)^{-\nu-n/2},$$

where α is a scaling factor which depends on ν, ℓ and n . Expanding $a_{\nu}(\boldsymbol{\xi})$ asymptotically, as $\|\boldsymbol{\xi}\| \rightarrow \infty$, and comparing with (II.5.6), we readily infer that the associated covariance kernel function is in $OPS_{cl}^{-2\nu-n}$ and, being independent of \mathbf{x} , also in $OPS_{cl,1}^{-2\nu-n}$. This also follows from the symbolic calculus of pseudodifferential operators upon noting that the symbol $a_{\nu}(\boldsymbol{\xi})$ coincides with the symbol of the inverse to the differential operator $A_{\nu} = \alpha^{-1}(\text{Id} - \frac{\ell^2}{2\nu} \Delta)^{\nu+n/2}$ which is of order $2\nu + n$ and thus $A_{\nu} \in OPS_{cl,1}^{2\nu+n}$.

We conclude this section by remarking that [DHS17] also provides some numerical evidence that Theorem II.5.18 could likely be extended to Lipschitz domains.

II.6 \mathcal{H} -matrix Arithmetic

The arithmetic of \mathcal{H} -matrices was first implemented in [GH03, Gra01], where the involved constants were also investigated in detail. The key idea of the arithmetic is to use the hierarchy of the block-cluster tree (and thus, the hierarchy on the \mathcal{H} -matrix) and to build a recursive algorithm. We shall recall the addition and the multiplication of \mathcal{H} -matrices as presented in [Hac15, Chapter 7].

II.6.1 Addition of \mathcal{H} -matrices

We start by recalling a fact about the sum of low-rank matrices. For a block-cluster $b = \tau \times \sigma$ and two ranks $k_{\mathbf{A}}, k_{\mathbf{B}}$, consider the low-rank matrices $\mathbf{A} \in \mathcal{R}(\tau \times \sigma, k_{\mathbf{A}})$ and $\mathbf{B} \in \mathcal{R}(\tau \times \sigma, k_{\mathbf{B}})$. It then holds

$$\mathbf{A} + \mathbf{B} = \mathbf{L}_{\mathbf{A}} \mathbf{R}_{\mathbf{A}}^{\top} + \mathbf{L}_{\mathbf{B}} \mathbf{R}_{\mathbf{B}}^{\top} = [\mathbf{L}_{\mathbf{A}} \ \mathbf{L}_{\mathbf{B}}] [\mathbf{R}_{\mathbf{A}} \ \mathbf{R}_{\mathbf{B}}]^{\top} \in \mathcal{R}(\tau \times \sigma, k_{\mathbf{A}} + k_{\mathbf{B}}). \quad (\text{II.6.1})$$

The storage requirement of $\mathbf{A} + \mathbf{B}$ is thus $(k_{\mathbf{A}} + k_{\mathbf{B}})(\#\tau + \#\sigma)$, although the actual rank of $\mathbf{A} + \mathbf{B}$ could be smaller than $k_{\mathbf{A}} + k_{\mathbf{B}}$. To reduce the storage requirements to the actual (numerical) rank of $\mathbf{A} + \mathbf{B}$, the singular value decomposition is needed.

Algorithm II.5 SVD of a low-rank matrix \mathbf{LR}^\top , see [Hac15, Algorithm 2.17]

function $\mathbf{U}\Sigma\mathbf{V}^\top = \text{LOWRANKSVD}(\mathbf{LR}^\top)$
 $\mathbf{Q}_L\mathbf{R}_L = \text{QR-decomposition of } \mathbf{L}, \mathbf{Q}_L \in \mathbb{R}^{\tau \times \tilde{k}}, \mathbf{R}_L \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$
 $\mathbf{Q}_R\mathbf{R}_R = \text{QR-decomposition of } \mathbf{R}, \mathbf{Q}_R \in \mathbb{R}^{\sigma \times \tilde{k}}, \mathbf{R}_R \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$
 $\tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^\top = \text{SVD}(\mathbf{R}_L\mathbf{R}_R^\top)$
 $\mathbf{U} = \mathbf{Q}_L\tilde{\mathbf{U}}$
 $\mathbf{V} = \mathbf{Q}_R\tilde{\mathbf{V}}$
 end function

Definition II.6.1. A singular value decomposition, SVD for short, of a matrix $\mathbf{M} \in \mathbb{R}^{\tau \times \sigma}$ is a decomposition of the form

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top,$$

where $\mathbf{U} \in \mathbb{R}^{\tau \times \tilde{k}}$ and $\mathbf{V} \in \mathbb{R}^{\sigma \times \tilde{k}}$ are orthogonal matrices, i.e., $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I} \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$, and $\Sigma \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$ is a diagonal matrix whose entries

$$\Sigma_{1,1} \geq \dots \geq \Sigma_{\tilde{k},\tilde{k}} > 0$$

are called singular values. Here, $\tilde{k} \leq \min\{\#\tau, \#\sigma\}$ denotes the actual rank of \mathbf{M} .

Remark II.6.2. A (not necessarily unique) singular value decomposition exists for all matrices in $\mathbb{R}^{\tau \times \sigma}$, see [GVL12, Theorem 2.4.1]. However, the above definition of the singular value decomposition is in contrast to the standard definition, where the matrices \mathbf{U} and \mathbf{V} contain a complete basis for the image space and the preimage space of \mathbf{M} .

The importance of the singular value decomposition lies in the fact that, for any given matrix $\mathbb{R}^{\tau \times \sigma}$ and any given rank $0 \leq k \leq \min\{\#\tau, \#\sigma\}$, the truncation of the singular value decomposition provides a best approximation in $\mathcal{R}(\tau \times \sigma, k)$.

Lemma II.6.3. Let $0 \leq k \leq \tilde{k} \leq \min\{\#\tau, \#\sigma\}$. Then, for a matrix $\mathbf{M} \in \mathbb{R}^{\tau \times \sigma}$ with rank \tilde{k} , a (not necessarily unique) best approximation of $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top$ in $\mathcal{R}(\tau \times \sigma, k)$ with respect to the Frobenius norm and the spectral norm is given by $\tilde{\mathbf{M}} = \mathbf{U}\tilde{\Sigma}\mathbf{V}^\top$, where $\tilde{\Sigma} = \text{diag}(\Sigma_{1,1}, \dots, \Sigma_{k,k}, 0, \dots, 0)$. In particular, it holds

$$\|\mathbf{M} - \tilde{\mathbf{M}}\|_F = \sqrt{\sum_{i=k+1}^{\tilde{k}} \Sigma_{ii}^2} \quad \text{and} \quad \|\mathbf{M} - \tilde{\mathbf{M}}\|_2 = \Sigma_{k+1,k+1}.$$

Furthermore, $\tilde{\mathbf{M}}$ is the best approximation to \mathbf{M} in the sense that

$$\|\mathbf{M} - \tilde{\mathbf{M}}\|_{F/2} = \min_{\mathbf{M}' \in \mathcal{R}(\tau \times \sigma, k)} \|\mathbf{M} - \mathbf{M}'\|_{F/2}.$$

Proof. The proof of this lemma is a straightforward consequence of an extension of \mathbf{U} and \mathbf{V} to an orthonormal basis and the orthogonal invariance of the Frobenius norm and the spectral norm. \square

The computation of the singular value decomposition of a dense $\#\tau \times \#\sigma$ -matrix takes $\mathcal{O}(\#\tau\#\sigma \min\{\#\tau, \#\sigma\})$ operations, see, e.g., [GVL12, Chapter 8.6], and is thus prohibitively expensive if $\#\tau$ and $\#\sigma$ are large. Instead, if the SVD of a matrix in $\mathcal{R}(\tau \times \sigma, \tilde{k})$ shall be computed, the algorithm listed in Algorithm II.5 requires only $\mathcal{O}((\#\tau + \#\sigma)\tilde{k}^2)$ operations, cf. [Hac15, Remark 2.18]. If we then set $\tilde{\mathbf{L}} = \mathbf{U}|_{\tau \times \tilde{k}}$ and $\tilde{\mathbf{R}} = \mathbf{V}|_{\sigma \times \tilde{k}}\Sigma|_{\tilde{k} \times \tilde{k}}$, we have an efficient implementation of the following truncation operator.

Definition II.6.4. Let $\mathbf{M} \in \mathcal{R}(\tau \times \sigma, \tilde{k})$ and $0 \leq k \leq \tilde{k}$. We define the truncation operator \mathcal{T}_k to matrices of rank k ,

$$\mathcal{T}_k: \mathcal{R}(\tau \times \sigma, \tilde{k}) \rightarrow \mathcal{R}(\tau \times \sigma, k), \quad \mathbf{M} \mapsto \mathbf{R},$$

to be one of the best approximations of \mathbf{M} in $\mathcal{R}(\tau \times \sigma, k)$ given by a singular value decomposition. We call $\mathbf{R} = \mathcal{T}_k(\mathbf{M})$ the truncation of \mathbf{M} to rank k .

The truncation operator

$$\mathcal{T}_k: \mathcal{H}(\mathcal{B}, \tilde{k}) \rightarrow \mathcal{H}(\mathcal{B}, k)$$

is defined by the block-wise application of the truncation operator to low-rank matrices to all matrix blocks in the farfield of \mathcal{B} .

Remark II.6.5. In order to avoid the loss of information, the truncation rank should not be fixed, but chosen adaptively for each matrix block such that a desired approximation accuracy ε is maintained. The required rank is usually referred to as ε -rank. The ε -rank can efficiently be determined after the third line of Algorithm II.5 by using the error expressions from Lemma II.6.3. However, the complexity bounds of the \mathcal{H} -matrix arithmetic usually assume that the ε -rank is bounded by some prescribed rank k . To achieve computational efficiency, one usually chooses the minimum of the ε -rank and k in implementations.

We are now in the position to compute the sum of two \mathcal{H} -matrices, both of which correspond to the same block-cluster tree. This is motivated by the addition of two asymptotically smooth kernels, being discretized on the same domain with respect to the same basis, cluster tree, and admissibility condition. We might also think of this operation as the sum of two pseudodifferential operators of the same Gevrey class, being again a pseudodifferential operator of this Gevrey class, which is verified by checking (II.5.6) and (II.5.7).

Definition II.6.6. The formatted \mathcal{H} -matrix addition of two \mathcal{H} -matrices $\mathbf{H}_1 \in \mathcal{H}(\mathcal{B}, k_1)$ and $\mathbf{H}_2 \in \mathcal{H}(\mathcal{B}, k_2)$, corresponding to the same block-cluster tree \mathcal{B} , is defined as

$$\mathbf{H}_1 +_{k_3} \mathbf{H}_2 = \mathcal{T}_{k_3}(\mathbf{H}_1 + \mathbf{H}_2)$$

for a given target rank k_3 . It then holds $\mathbf{H}_3 \in \mathcal{H}(\mathcal{B}, k_3)$.

An algorithmic realisation of the \mathcal{H} -matrix addition is given in Algorithm II.6 and results directly from the recursive block matrix structure of \mathcal{H} -matrices, discussed in (II.5.4). The following interactions between the different kinds of matrix blocks hold:

$+_{k_3}$	\mathcal{H} -matrix	low-rank matrix	dense matrix
\mathcal{H} -matrix low-rank matrix dense matrix	recursively	approximately	exactly

In accordance with [Hac15, Lemma 7.20], the complexity of the algorithm, assuming $k_3 < k_1 + k_2$, is $\mathcal{O}((k_1 + k_2)^2 N \log N)$ where the hidden constant depends on $C_{\text{sp}}(\mathcal{B})$. Obviously, the previous considerations can be generalized to formulate a *formatted matrix subtraction*, which we denote by $-_{k_3}$.

II.6.2 Multiplication of \mathcal{H} -matrices

Informally, the multiplication of two \mathcal{H} -matrices can be seen as the discrete version of the composition of the two associated pseudodifferential operators. In particular, if the two pseudodifferential operators are of Gevrey class, their product is again a pseudodifferential operator of Gevrey class, see Theorem II.5.19, which means that it is compressible by means of \mathcal{H} -matrices. We may choose the same bases for the Galerkin discretization with the same cluster tree, and, as we will see without loss of generality, we choose the same block-cluster tree \mathcal{B} for the representation of the \mathcal{H} -matrices. We thus have to implement an algebraic operation, which efficiently computes the product of two matrices $\mathbf{H}_1 \in \mathcal{H}(\mathcal{B}, k_1)$ and $\mathbf{H}_2 \in \mathcal{H}(\mathcal{B}, k_2)$ and represents it in $\mathcal{H}(\mathcal{B}, k_3)$. Although one can prove that the product can exactly be represented in $\mathcal{H}(\mathcal{B}, k_3)$ when k_3 is sufficiently large (i.e., k_3 should be $\mathcal{O}(k_1 k_2 \log N)$), see [GH03], we

Algorithm II.6 Addition of \mathcal{H} -matrices, start with $b = \text{root}(\mathcal{B})$, see [Hac15, Chapter 7.3]

```

function  $\mathbf{H}_3 = \text{ADD}\mathcal{H}(\mathbf{H}_1|_b, \mathbf{H}_2|_b, k_3)$ 
  if  $b$  has sons then
    for  $b' \in \text{sons}(b)$  do
       $\mathbf{H}_3|_{b'} = \text{ADD}\mathcal{H}(\mathbf{H}_1|_{b'}, \mathbf{H}_2|_{b'}, k_3)$ 
    end for
  else
    if  $b \in \mathcal{F}$  then
       $\mathbf{H}_3|_b = \mathcal{T}_{k_3}(\mathbf{H}_1|_b + \mathbf{H}_2|_b)$ 
    else
       $\mathbf{H}_3|_b = \mathbf{H}_1|_b + \mathbf{H}_2|_b$ 
    end if
  end if
end function
    
```

aim at keeping k_3 low and in particular independent of N to obtain computational efficiency. Think for example of $k_1 = k_2 = k_3$.

The algorithm of the *formatted \mathcal{H} -matrix multiplication* is again based on the recursive block structure of \mathcal{H} -matrices from (II.5.4). Looking at the multiplication of \mathcal{H} -matrices as the product of block matrices, we derive the following structures of the products of the different kinds of blocks:

*	\mathcal{H} -matrix	low-rank matrix	dense matrix
\mathcal{H} -matrix	\mathcal{H} -matrix	low-rank	dense
low-rank matrix	low-rank	low-rank	low-rank
dense matrix	dense	low-rank	dense

Products involving low-rank matrices can exactly be computed by the multiple application of matrix-vector products, which results again in low-rank matrices. A similar consideration holds for the interaction between a dense matrix and an \mathcal{H} -matrix. We get thus the following accuracies for the products:

*	\mathcal{H} -matrix	low-rank matrix	dense matrix
\mathcal{H} -matrix	recursively	exactly	exactly
low-rank matrix	exactly	exactly	exactly
dense matrix	exactly	exactly	exactly

Unfortunately, computing the product of the matrix blocks is only a part of the \mathcal{H} -matrix multiplication. Since the products should be stored in the format of the block-cluster tree \mathcal{B} , the results of the single products have to be converted into the format of the corresponding target block. For ease of presentation, we consider the operation $\mathbf{H}_3 +=_{k_3} \mathbf{H}_1 * \mathbf{H}_2$ and then have the following interactions of different kinds of matrix blocks:

$+=_{k_3}$	\mathcal{H} -matrix	low-rank matrix	dense matrix
\mathcal{H} -matrix	recursively	recursively	recursively
low-rank matrix	approximately	approximately	approximately
dense matrix	exactly	exactly	exactly

The operands in the rows of the table coincide with the target format of the respective operation.

If the target format is a dense matrix, $+=_{k_3}$ denotes the $+=$ -operator, which is explained in the usual way. For the remaining operations, we shall now recapitulate suitable truncation operators based on the truncation operator to low-rank matrices from Definition II.6.4.

Algorithm II.7 Convert \mathcal{H} -matrix to low-rank matrix, see [Hac15, Equation (7.8)]

```

function  $\mathbf{R} = \mathcal{H2R}(\mathbf{H}|_b, k_3)$ 
     $\mathbf{R} = \text{ZERO}(\tau, \sigma)$ 
    if  $b$  has sons then
        for  $b' \in \text{sons}(b)$  do
             $\mathbf{R} \mathop{+}_{k_3} \mathcal{H2R}(\mathbf{H}|_{b'}, k_3)|_{b' \rightarrow b}$ 
        end for
    else  $\triangleright \mathbf{H}|_b$  is dense or low-rank
         $\mathbf{R} \mathop{+}_{k_3} \mathbf{H}|_b$ 
    end if
end function
    
```

The situation $\mathbf{R} \mathop{+}_{k_3} \mathbf{F}$. In this case, a low-rank approximation to the dense matrix is computed using the SVD. The rest of the operation is similar to the formatted \mathcal{H} -matrix addition.

The situation $\mathbf{R} \mathop{+}_{k_3} \mathbf{R}$. This case is directly handled by the formatted \mathcal{H} -matrix addition.

The situation $\mathbf{R} \mathop{+}_{k_3} \mathbf{H}$. This case was originally called *hierarchical conversion*, see [GH03]. The basic idea is to convert, starting from the leaves, the \mathcal{H} -matrix level-by-level into a low-rank matrix. The algorithm for this operation is listed in Algorithm II.7, where the operation $\mathbf{M}|_{b' \rightarrow b}$ for $b' \in \text{sons}(b)$ and matrices $\mathbf{M} \in \mathbb{R}^{b'}$ is given by

$$(\mathbf{M}|_{b' \rightarrow b})_{ij} = \begin{cases} (\mathbf{M}_{b'})_{ij}, & (i, j) \in b', \\ 0, & (i, j) \in b \setminus b'. \end{cases}$$

Algorithm II.7 does not lead to the best possible low-rank approximation of a given \mathcal{H} -matrix. However, assuming \mathbf{R}_{best} is such a best approximation and $\mathbf{R}_{\text{approx}}$ is the result given by Algorithm II.7, one can prove that

$$\|\mathbf{H}|_b - \mathbf{R}_{\text{approx}}\|_F \leq \left(1 + \left(\frac{1 + \sqrt{5}}{2}\right)^{1 + \text{depth}(b)}\right) \|\mathbf{H}|_b - \mathbf{R}_{\text{best}}\|_F,$$

see [Hac15, Lemma 7.6]. We refer to [Hac15, Appendix C.3] for a discussion why much better estimates can be observed in practice and remark that we will discuss an alternate approach for this operation in Chapter V.2.

The situation $\mathbf{H} \mathop{+}_{k_3} \mathbf{F}$. Since the recursive block matrix pattern of an \mathcal{H} -matrix can also be applied to a dense matrix, this operation can be reduced to the operations $\mathbf{F} \mathop{+}_{k_3} \mathbf{F}$ and $\mathbf{R} \mathop{+}_{k_3} \mathbf{F}$ by a recursive algorithm similar to the addition of \mathcal{H} -matrices in Algorithm II.6.

The situation $\mathbf{H} \mathop{+}_{k_3} \mathbf{R}$. Similar to the previous operation, we can explain this operation in a recursive manner, since we can exactly represent \mathbf{R} with respect to the recursive block matrix structure of the target \mathcal{H} -matrix by recursively breaking up the low-rank structure. In the first step, we have

$$\mathbf{R} = \begin{array}{|c|} \hline \text{[Colorful Row]} \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline \text{[Colorful Row]} & \text{[Colorful Row]} & \text{[Colorful Row]} & \text{[Colorful Row]} \\ \hline \end{array} = (\mathbf{R}|_{\tau' \times \sigma'})_{\substack{\tau' \in \text{sons}(\tau) \\ \sigma' \in \text{sons}(\sigma)}}$$

The remaining operations are then covered by the cases $\mathbf{R} \mathop{+}_{k_3} \mathbf{F}$ and $\mathbf{R} \mathop{+}_{k_3} \mathbf{R}$.

For numerical issues, the subdivision of a low-rank matrix to its son clusters can be realized by index shifts. Therefore, no additional calculations or storage are necessary here.

The situation $\mathbf{H} \mathop{+}_{k_3} \mathbf{H}$. This final situation can completely be covered by a recursive algorithm and the previously introduced operations.

Algorithm II.8 Multiplication of \mathcal{H} -matrices, see [Hac15, Equation (7.27)]

```

function  $\mathbf{H}^{(3)}|_{\tau \times \sigma += k_3}$  MULT $\mathcal{H}(\mathbf{H}^{(1)}|_{\tau \times \rho}, \mathbf{H}^{(2)}|_{\rho \times \sigma})$ 
  if  $\tau \times \sigma$  has sons then
    for  $\tau' \in \text{sons}(\tau)$  and  $\sigma' \in \text{sons}(\sigma)$  do
      for  $\rho' \in \text{sons}(\rho)$  do
         $\mathbf{H}^{(3)}|_{\tau' \times \sigma' += k_3}$  MULT $\mathcal{H}(\mathbf{H}^{(1)}|_{\tau' \times \rho'}, \mathbf{H}^{(2)}|_{\rho' \times \sigma'})$ 
      end for
    end for
  else
     $\mathbf{H}^{(3)}|_{\tau \times \sigma += k_3} \mathbf{H}^{(1)}|_{\tau \times \rho} * \mathbf{H}^{(2)}|_{\rho \times \sigma}$ 
  end if
end function
    
```

Having explained the necessary operations, the multiplication of \mathcal{H} -matrices is given by Algorithm II.8. In accordance with [GH03, Theorem 2.24], the complexity of the algorithm, assuming for simplicity $\mathbf{H}^{(1)}, \mathbf{H}^{(2)} \in \mathcal{H}(\mathcal{B}, k)$ and $n_{\min} \leq k$, is $\mathcal{O}(kN \log N (\log N + k^2))$. The hidden constant depends on $C_{\text{sp}}(\mathcal{B})$ and the quantity $C_{\text{id}}(\mathcal{B})$ given by

$$C_{\text{id}}(\tau \times \sigma) := \#\{\tau' \times \sigma' : \tau' \in \text{successors}(\tau), \sigma' \in \text{successors}(\sigma) \text{ such that} \\ \text{there exists } \rho' \in \mathcal{T}_{\mathcal{I}} \text{ such that } \tau' \times \rho' \in \mathcal{B}, \rho' \times \sigma' \in \mathcal{B}\},$$

$$C_{\text{id}}(\mathcal{B}) := \max_{\tau \times \sigma \in \mathcal{L}(\mathcal{B})} C_{\text{id}}(\tau \times \sigma).$$

Similar to $C_{\text{sp}}(\mathcal{B})$, on quasi-uniform meshes and when the clustering technique from Algorithm II.2 is used, it is reasonable to assume that $C_{\text{id}}(\mathcal{B})$ can be bounded by a constant depending on η from the admissibility condition (II.5.6), C_{sep} from (II.4.6), and the spatial dimension.

We shall remark at this point that the algorithm itself does not return the best possible approximation to the product of two \mathcal{H} -matrices. However, numerical experiments indicate that the result of the algorithm is close to the best approximation in $\mathcal{H}(\mathcal{B}, k)$. For an algorithm yielding the actual best approximation in complexity $\mathcal{O}(k^3 N \log^3 N)$ with significantly larger constants, we refer to [Hac15, Chapter 7.4.3.3].

We conclude this section by remarking that, by replacing the \mathcal{H} -matrix addition by the \mathcal{H} -subtraction, one can construct the operation $\mathbf{H}_{3 -= k_3} \mathbf{H}_1 * \mathbf{H}_2$ with similar complexity.

II.6.3 \mathcal{H} -matrix Factorizations

Based on the formatted addition, multiplication, and the hierarchical block-matrix structure of \mathcal{H} -matrices, recursive algorithms for the formatted inversion, the LU or Cholesky factorization, and forward and backward substitution can be formulated. Throughout this thesis, we will only require the action of the inverse of an \mathcal{H} -matrix to another \mathcal{H} -matrix. Since the computation of an LU or Cholesky factorization with subsequent forward and backward substitution is more efficient than the computation and the application of an inverse, we will concentrate on factorizations in this section.

Given an (approximation of an) LU factorization of an \mathcal{H} -matrix stored in the \mathcal{H} -matrix format, the forward substitution applied to a vector can be formulated as stated in Algorithm II.9. We recall that the block-cluster tree yields a symmetric partitioning, see Lemma II.5.8, and remark that \mathbf{r} is overwritten during the algorithm. The algorithm becomes an in-place algorithm if the variable \mathbf{y} is replaced by \mathbf{r} . Similar considerations hold for the formulation of the backward substitution algorithm which is listed in Algorithm II.10.

While the forward and backward substitution of \mathcal{H} -matrices, applied to vectors, is exact, the application to \mathcal{H} -matrices relies on the approximate addition and multiplication of \mathcal{H} -matrices as discussed in the previous section. The forward substitution algorithm for the

Algorithm II.9 Forward substitution $\mathbf{y} = \mathbf{L}^{-1}\mathbf{r}$, see [Hac15, Equation (7.32a)]

```

function  $\mathbf{y}|_{\tau} = \mathcal{H}\text{FORVEC}(\mathbf{L}|_{\tau \times \tau}, \mathbf{r}|_{\tau})$ 
  if  $\tau \times \tau \notin \mathcal{L}(\mathcal{B})$  then
    for  $i = 1, \dots, \#\text{sons}(\tau)$  do
       $\mathbf{y}_{\text{sons}(\tau)_i} = \mathcal{H}\text{FORVEC}(\mathbf{L}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i}, \mathbf{r}|_{\text{sons}(\tau)_i})$ 
      for  $j = i + 1, \dots, \#\text{sons}(\tau)$  do
         $\mathbf{r}|_{\text{sons}(\tau)_j} -= \mathbf{L}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i} \mathbf{r}|_{\text{sons}(\tau)_i}$ 
      end for
    end for
  else
     $\mathbf{y}|_{\tau} = \text{DENSEFORVEC}(\mathbf{L}|_{\tau \times \tau}, \mathbf{r}|_{\tau})$ 
  end if
end function

```

Algorithm II.10 Backward substitution $\mathbf{x} = \mathbf{U}^{-1}\mathbf{y}$, see [Hac15, Chapter 7.6.2]

```

function  $\mathbf{x}|_{\tau} = \mathcal{H}\text{BACKVEC}(\mathbf{L}|_{\tau \times \tau}, \mathbf{y}|_{\tau})$ 
  if  $\tau \times \tau \notin \mathcal{L}(\mathcal{B})$  then
    for  $i = \#\text{sons}(\tau), \dots, 1$  do
       $\mathbf{x}_{\text{sons}(\tau)_i} = \mathcal{H}\text{BACKVEC}(\mathbf{U}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i}, \mathbf{y}|_{\text{sons}(\tau)_i})$ 
      for  $j = 1, \dots, i - 1$  do
         $\mathbf{y}|_{\text{sons}(\tau)_j} -= \mathbf{U}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i} \mathbf{y}|_{\text{sons}(\tau)_i}$ 
      end for
    end for
  else
     $\mathbf{x}|_{\tau} = \text{DENSEBACKVEC}(\mathbf{L}|_{\tau \times \tau}, \mathbf{y}|_{\tau})$ 
  end if
end function

```

Algorithm II.11 Forward substitution $\mathbf{Y} = \mathbf{L}^{-1}\mathbf{R}$, see [Hac15, Equation (7.33a)]

```

function  $\mathbf{Y}|_{\tau \times \sigma} = \mathcal{H}\text{FORMAT}(\mathbf{L}|_{\tau \times \tau}, \mathbf{R}|_{\tau \times \sigma}, k_3)$ 
  if  $\tau \times \sigma \notin \mathcal{L}(\mathcal{B})$  then
    for  $i = 1, \dots, \#\text{sons}(\tau)$  and  $\sigma' \in \text{sons}(\sigma)$  do
       $\mathbf{Y}_{\text{sons}(\tau)_i \times \sigma'} = \mathcal{H}\text{FORMAT}(\mathbf{L}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i}, \mathbf{R}|_{\text{sons}(\tau)_i \times \sigma'}, k_3)$ 
      for  $j = i + 1, \dots, \#\text{sons}(\tau)$  do
         $\mathbf{R}|_{\text{sons}(\tau)_j \times \sigma'} =_{k_3} \mathbf{L}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i} * \mathbf{R}|_{\text{sons}(\tau)_i \times \sigma'}$ 
      end for
    end for
  else
    if  $\tau \times \sigma \in \mathcal{F}$  then  $\triangleright \mathbf{R}|_{\tau \times \sigma} = \mathbf{A}\mathbf{B}^\top \in \mathcal{R}(\tau \times \sigma, k)$ 
      Solve  $\mathbf{C} = \mathbf{L}|_{\tau \times \tau}^{-1} \mathbf{A}$ , applying  $\mathcal{H}\text{FORVEC}(\mathbf{L}|_{\tau \times \tau}, \cdot)$  column-wise
      Set  $\mathbf{Y}|_{\tau \times \sigma} = \mathbf{C}\mathbf{B}^\top$ 
    else
      Solve  $\mathbf{Y}|_{\tau \times \sigma} = \mathbf{L}|_{\tau \times \tau}^{-1} \mathbf{R}|_{\tau \times \sigma}$ , applying  $\mathcal{H}\text{FORVEC}(\mathbf{L}|_{\tau \times \tau}, \cdot)$  column-wise
    end if
  end if
end function

```

application to an \mathcal{H} -matrix can be found in Algorithm II.11, whereas the backward substitution algorithm can be found in Algorithm II.12. For the computation of an (approximate) LU factorization of an \mathcal{H} -matrix, to which we will refer as the \mathcal{H} -LU factorization, we shall need the forward substitution and an algorithm $\mathbf{X} = \mathcal{H}\text{BACKMATR}(\mathbf{U}, \mathbf{Z})$, which solves $\mathbf{X}\mathbf{U} = \mathbf{Z}$. It is a straightforward modification of Algorithm II.12, which we leave to the reader. Having these algorithms available, the \mathcal{H} -LU factorization can be computed as formulated in Algorithm II.13. The formulation of an \mathcal{H} -Cholesky factorization is analogous and also left to the reader.

We recall the computational complexity of the algorithms of this section in accordance with [Hac15, Chapter 7.8.5]. Since the factorizations themselves are stored in \mathcal{H} -matrix format, their storage consumption coincides with a generic \mathcal{H} -matrix in $\mathcal{H}(\mathcal{B}, k)$, which is $\mathcal{O}(kN \log N)$. The computational complexity for the forward and backward substitution, applied to a vector, is also $\mathcal{O}(kN \log N)$. In both cases, the hidden constant depends on $C_{\text{sp}}(\mathcal{B})$. The complexity of the application of the substitution algorithms to a matrix and the computation of the factorizations itself has the same complexity as the formatted \mathcal{H} -matrix multiplication, that is, $\mathcal{O}(kN \log N (\log N + k^2))$. In these cases, the hidden constants depend on $C_{\text{sp}}(\mathcal{B})$ and $C_{\text{id}}(\mathcal{B})$.

We conclude this chapter by remarking that the parallelization of the \mathcal{H} -matrix arithmetic and the \mathcal{H} -LU factorization was discussed in [Iza12, Kri05, Kri13].

Algorithm II.12 Backward substitution $\mathbf{X} = \mathbf{U}^{-1}\mathbf{Y}$, see [Hac15, Chapter 7.6.3]

```

function  $\mathbf{X}|_{\tau \times \sigma} = \mathcal{H}\text{BACKMAT}(\mathbf{U}|_{\tau \times \tau}, \mathbf{Y}|_{\tau \times \sigma}, k_3)$ 
  if  $\tau \times \sigma \notin \mathcal{L}(\mathcal{B})$  then
    for  $i = \#\text{sons}(\tau), \dots, 1$  and  $\sigma' \in \text{sons}(\sigma)$  do
       $\mathbf{X}_{\text{sons}(\tau)_i \times \sigma'} = \mathcal{H}\text{BACKMAT}(\mathbf{U}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i}, \mathbf{Y}|_{\text{sons}(\tau)_i \times \sigma'}, k_3)$ 
      for  $j = 1, \dots, i - 1$  do
         $\mathbf{Y}|_{\text{sons}(\tau)_j \times \sigma'} \mathrel{-} \mathrel{=}_{k_3} \mathbf{U}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i} * \mathbf{Y}|_{\text{sons}(\tau)_i \times \sigma'}$ 
      end for
    end for
  else
    if  $\tau \times \sigma \in \mathcal{F}$  then  $\triangleright \mathbf{Y}|_{\tau \times \sigma} = \mathbf{A}\mathbf{B}^\top \in \mathcal{R}(\tau \times \sigma, k)$ 
      Solve  $\mathbf{C} = \mathbf{L}|_{\tau \times \tau}^{-1} \mathbf{A}$ , applying  $\mathcal{H}\text{BACKVEC}(\mathbf{U}|_{\tau \times \tau}, \cdot)$  column-wise
      Set  $\mathbf{X}|_{\tau \times \sigma} = \mathbf{C}\mathbf{B}^\top$ 
    else
      Solve  $\mathbf{X}|_{\tau \times \sigma} = \mathbf{L}|_{\tau \times \tau}^{-1} \mathbf{Y}|_{\tau \times \sigma}$ , applying  $\mathcal{H}\text{BACKVEC}(\mathbf{U}|_{\tau \times \tau}, \cdot)$  column-wise
    end if
  end if
end function

```

Algorithm II.13 In-place \mathcal{H} -LU factorization, see [Hac15, Chapter 7.6.4]

```

function  $\mathcal{H}\text{-LU}(\mathbf{A}|_{\tau \times \tau}, k_3)$ 
  if  $\tau \times \tau \notin \mathcal{L}(\mathcal{B})$  then
    for  $i = 1, \dots, \#\text{sons}(\tau)$  do
      for  $j = 1, \dots, i - 1$  do  $\triangleright$  Update diagonal
         $\mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i} \mathrel{-} \mathrel{=}_{k_3} \mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_j} * \mathbf{A}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i}$ 
      end for
       $\mathcal{H}\text{-LU}(\mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i}, k_3)$ 
      for  $j = i + 1, \dots, \#\text{sons}(\tau)$  do  $\triangleright$  Update strictly lower part
        for  $k = 1, \dots, i - 1$  do
           $\mathbf{A}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i} \mathrel{-} \mathrel{=}_{k_3} \mathbf{A}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_k} * \mathbf{A}|_{\text{sons}(\tau)_k \times \text{sons}(\tau)_i}$ 
        end for
         $\mathbf{A}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i} = \mathcal{H}\text{BACKMATR}(\mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i}, \mathbf{A}|_{\text{sons}(\tau)_j \times \text{sons}(\tau)_i})$ 
      end for
      for  $j = i + 1, \dots, \#\text{sons}(\tau)$  do  $\triangleright$  Update strictly upper part
        for  $k = 1, \dots, i - 1$  do
           $\mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_j} \mathrel{-} \mathrel{=}_{k_3} \mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_k} * \mathbf{A}|_{\text{sons}(\tau)_k \times \text{sons}(\tau)_j}$ 
        end for
         $\mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_j} = \mathcal{H}\text{FORMAT}(\mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_i}, \mathbf{A}|_{\text{sons}(\tau)_i \times \text{sons}(\tau)_j})$ 
      end for
    end for
  else  $\triangleright \mathbf{A}|_{\tau \times \tau}$  is a dense matrix
     $\text{DENSELU}(\mathbf{A}|_{\tau \times \tau})$ 
  end if
end function

```

Chapter III

PDEs with Random Load

III.1 Problem Formulation

The topic of this chapter is the solution of correlation equations which have their origins in the uncertainty quantification of partial differential equations with random load. More specifically, for the remainder of this chapter, let $D \subset \mathbb{R}^n$ be a Lipschitz domain, $(\Omega, \Sigma, \mathbb{P})$ a separable and complete probability space and \mathcal{L} be the linear differential operator of second order given by

$$(\mathcal{L}u)(\mathbf{x}) = -\operatorname{div}(\mathbf{A}(\mathbf{x}) \cdot \nabla u(\mathbf{x})) + \langle \mathbf{b}(\mathbf{x}), \nabla u(\mathbf{x}) \rangle + c(\mathbf{x})u(\mathbf{x}). \quad (\text{III.1.1})$$

The coefficients shall satisfy $\mathbf{A} \in W^{1,\infty}(D; \mathbb{R}^{n \times n})$, $\mathbf{b} \in L^\infty(D; \mathbb{R}^n)$, and $c \in L^\infty(D; \mathbb{R})$, and the corresponding bilinear form

$$a(u, v) = (\mathcal{L}u, v)_{L^2(D)}$$

is assumed to be bounded and strongly elliptic.

Under these assumptions, for a given load $f \in L^2_{\mathbb{P}}(\Omega; H^{-1}(D))$, the Dirichlet problem

$$\begin{aligned} \mathcal{L}u(\omega, \mathbf{x}) &= f(\omega, \mathbf{x}) & \text{for } \mathbf{x} \in D, \\ u(\omega, \mathbf{x}) &= 0 & \text{for } \mathbf{x} \in \partial D, \end{aligned} \quad (\text{III.1.2})$$

is known to have a unique solution $u(\omega, \cdot) \in H_0^1(D)$ for \mathbb{P} -almost every $\omega \in \Omega$, see, e.g., [Bra13, Theorem 2.9]. In particular, the differential operator \mathcal{L} is a mapping $\mathcal{L}: H_0^1(D) \rightarrow H^{-1}(D)$. Following the considerations in Chapter II.4, the mean $\mathbb{E}[u] \in H_0^1(D)$ and the correlation $\operatorname{Cor}[u] \in H_0^1(D) \otimes H_0^1(D)$ are well defined. Moreover, the coefficients of the Galerkin approximations of the mean and the correlation of u are given by

$$\mathbf{A}\mathbf{e}_u = \mathbf{e}_f$$

and

$$\mathbf{A}\mathbf{C}_u\mathbf{A}^\top = \mathbf{C}_f, \quad (\text{III.1.3})$$

respectively. Here, \mathbf{A} denotes the finite element stiffness matrix.

We remark that this approach was extended to boundary value problems with non-homogeneous boundary conditions in [Har10a], but, for simplicity, we restrict ourselves to homogeneous boundary conditions. A specific version of Theorem II.5.18 for (III.1.2) is then given by the following corollary.

Corollary III.1.1. *In the domain D with analytic boundary ∂D , assume that the correlation $\operatorname{Cor}[f]$ gives rise to an operator $\mathcal{K}_{\operatorname{Cor}[f]} \in OPS_{cl,s}^\theta(D)$, i.e., to a classical pseudodifferential*

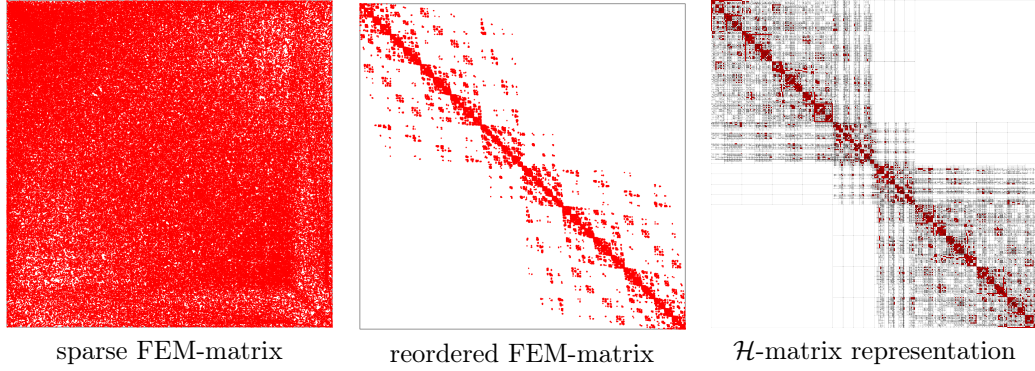


Figure III.1: Sparsity pattern of a three-dimensional finite element matrix, its reordered finite element matrix, and the corresponding \mathcal{H} -matrix. Red blocks in the \mathcal{H} -matrix correspond to the nearfield, white blocks correspond to the empty farfield.

operator with symbol $a_{\text{Cor}[f]}(\mathbf{x}, \boldsymbol{\xi})$ of order θ and of Gevrey class $s \geq 1$. Assume further that the coefficients of the differential operator \mathcal{L} are of Gevrey class s . Then, the correlation $\text{Cor}[u]$ is the Schwartz kernel of an operator $\mathcal{K}_{\text{Cor}[u]} \in OPS_{cl,s}^{\theta-4}(D)$.

Moreover, $\text{Cor}[u]$ is smooth in $D \times D$ outside of the diagonal and there holds the pointwise estimate

$$|\partial_{\mathbf{x}}^{\boldsymbol{\alpha}} \partial_{\mathbf{y}}^{\boldsymbol{\beta}} \text{Cor}[u](\mathbf{x}, \mathbf{y})| \leq c \mathcal{A}^{|\boldsymbol{\alpha}+\boldsymbol{\beta}|} (|\boldsymbol{\alpha}|!)^s \beta! \|\mathbf{x} - \mathbf{y}\|^{-n-\theta+4-|\boldsymbol{\alpha}|-|\boldsymbol{\beta}|}$$

away from the diagonal for all $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}_0^n$, with some constants c and \mathcal{A} which depend only on D and $a_{\text{Cor}[f]}$.

Thus, if we assume $\text{Cor}[f]$ to be asymptotically smooth and to fulfil the above assumptions, we can represent $\text{Cor}[f]$ and, if the domain is smooth enough, $\text{Cor}[u]$ by an \mathcal{H} -matrix. It remains to discuss whether the finite element matrix can be represented by an \mathcal{H} -matrix, which, since \mathbf{A} is a sparse matrix, is not obvious.

An analytical reasoning is that we can write \mathcal{L} as an integral operator with distributional kernel k , i.e.,

$$(\mathcal{L}u)(\mathbf{x}) = \int_D \underbrace{\left(- \sum_{i,j=1}^d a_{ij}(\mathbf{x}) \partial_{y_i} \partial_{y_j} \delta_{\mathbf{x}}(\mathbf{y}) + \sum_{i=1}^d b_i(\mathbf{x}) \partial_{y_i} \delta_{\mathbf{x}}(\mathbf{y}) + c(\mathbf{x}) \delta_{\mathbf{x}}(\mathbf{y}) \right)}_{=:k(\mathbf{x},\mathbf{y})} u(\mathbf{y}) \, d\mathbf{y}.$$

Since the Dirac distribution $\delta_{\mathbf{x}}(\mathbf{y})$ at the point \mathbf{x} is formally zero for $\mathbf{x} \neq \mathbf{y}$, we may argue that k is an asymptotically smooth function and \mathbf{A} can be represented as an \mathcal{H} -matrix. A more algebraic and mathematically sound reasoning is that a necessary condition for an entry \mathbf{A}_{ij} in the finite element matrix to be non-zero is $\Upsilon_i \cap \Upsilon_j \neq \emptyset$, i.e., the intersection of the corresponding supports of the basis functions is non-empty. This yields, together with the η -admissibility condition (II.5.3), that all entries of a finite element matrix have η -inadmissible supports, i.e., they are contained in the nearfield of an \mathcal{H} -matrix. A sparse finite element matrix can therefore be represented as an \mathcal{H} -matrix by reordering the index set and inserting the non-zero entries into the nearfield. An illustration of this procedure can be found in Figure III.1.

It is thus reasonable to represent all matrices occurring in (III.1.3) by \mathcal{H} -matrices. It remains to discuss how to compute the unknown matrix \mathbf{C}_u . In the next section, we recapitulate some specialities of \mathcal{H} -matrices in the context of finite elements and then present a solution algorithm to solve (III.1.3) in almost linear time using the \mathcal{H} -matrix arithmetic from Chapter II.6.

III.2 \mathcal{H} -matrices in the Context of Finite Elements

Although a finite element matrix has a sparse structure, its inverse and both factors of its LU factorization are generally densely populated. Nevertheless, the inverse and the LU factorization exhibit a data-sparse structure in the sense that they are \mathcal{H} -matrix compressible. We recall the main concepts from the literature, see, e.g., [Beb05, Beb07, BH03, Fau15, FMP15].

Other than the calculus of pseudodifferential operators as discussed in [DHS17], a rough argument for the \mathcal{H} -matrix compressibility of the inverse makes use of the Green's function G of \mathcal{L} . Let $\delta_{\mathbf{x}}$ denote the Dirac distribution at the point \mathbf{x} , and let $G: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ satisfy

$$\mathcal{L}_{\mathbf{y}}G(\mathbf{x}, \mathbf{y}) = \delta_{\mathbf{x}}(\mathbf{y}) \text{ and } G(\cdot, \mathbf{y})|_{\partial D} = 0. \quad (\text{III.2.1})$$

Then, the solution of

$$\begin{aligned} \mathcal{L}u(\mathbf{x}) &= f(\mathbf{x}) & \text{for } \mathbf{x} \in D, \\ u(\mathbf{x}) &= 0 & \text{for } \mathbf{x} \in \partial D, \end{aligned}$$

can be represented by

$$u(\mathbf{x}) = (\mathcal{L}^{-1}f)(\mathbf{x}) = \int_D G(\mathbf{x}, \mathbf{y})f(\mathbf{y}) \, d\mathbf{y}, \quad \mathbf{x} \in D.$$

If the Green's function is analytic away from the diagonal, e.g., in the case of constant coefficients of \mathcal{L} , we can approximate the Green's function away from the diagonal by local expansions of the kind

$$G(\mathbf{x}, \mathbf{y}) \approx \sum_{i=1}^k a_i(\mathbf{x})b_i(\mathbf{y}),$$

which is the foundation for an \mathcal{H} -matrix approximation.

One of the advantages of the finite element method is that it can also be applied in case of non-constant coefficients. In [BH03], a proof was presented to guarantee the existence of an \mathcal{H} -matrix approximation to the inverse of the finite element stiffness matrix even in the case of essentially bounded diffusion coefficients and the other coefficients set to zero. This result was then extended in [Beb05] to allow all coefficients to be only essentially bounded, providing the theoretical foundation for an \mathcal{H} -matrix approximation to the inverse of the differential operator from (III.1.1). Having the \mathcal{H} -matrix approximability of the inverse of the finite element matrix available, the approximability of the LU factorization of the finite element matrix was proven in [Beb07]. While these first results hold up to the finite element discretization error, the results were recently improved in [Fau15, FMP15] to hold without additional error.

Having the finite element matrix represented by an \mathcal{H} -matrix, the LU factorization can be computed with Algorithm II.13 in $\mathcal{O}(k^2N \log^2 N)$ operations. As discussed in Chapter II.6, an approximate inverse can also be computed in $\mathcal{O}(k^2N \log^2 N)$ operations, but the computation of the LU factorization together with its forward and backward substitution algorithms, listed in Algorithms II.11 and II.12, has smaller constants than the computation and the application of an approximate inverse.

III.2.1 Weak Admissibility

Approximate \mathcal{H} -matrix representations for the inverse or LU factorizations of finite element matrices were used to construct preconditioners for iterative solvers, see, e.g., [Beb08, GKL09] and the references therein. In [HKK04], it was observed for the one-dimensional case that the computation of an approximate inverse can be considerably sped up by replacing the η -admissibility condition (II.5.3) by the following weak admissibility condition.

Definition III.2.1. *Two clusters σ and σ' are called weakly admissible if $\sigma \neq \sigma'$.*

We observe immediately that an η -admissible block-cluster is also weakly admissible. Thus, by replacing the η -admissibility condition by the weak admissibility, we obtain a much coarser partition of the \mathcal{H} -matrix. This leads to smaller constants in the storage and computational complexity, cf. [HKK04]. We also remark that finite element matrices can exactly be represented as \mathcal{H} -matrices which were constructed using the weak admissibility condition. In particular, the corresponding off-diagonal blocks have low-rank, cf. [HKK04].

By partitioning the matrix according to the weak admissibility condition, we cannot ensure the exponential convergence of fast black box low-rank approximation techniques as used for boundary element matrices. For example, the techniques in [Beb00, BG04a] rely on an admissibility condition similar to (II.5.3) to ensure exponential convergence. Instead, the authors of [HKK04] suggest assembling a weakly admissible matrix block according to the η -admissibility condition and transforming it on-the-fly into a low-rank matrix in order to obtain a good approximation.

The behavior of the ranks of the low-rank matrices in weakly admissible partitions is not yet fully understood compared to η -admissible partitions. Suppose that k_η is an upper bound for the ranks corresponding to an η -admissible partition and suppose that k_w shall be an upper bound for the ranks of a weakly admissible partition. In [HKK04], it is proven for one-dimensional finite element discretizations that one should generally choose

$$k_w = Lk_\eta$$

in order to obtain the same approximation accuracy in the weakly admissible case as in the η -admissible case. Here, L is a constant which depends on the depth of the block-cluster tree, which is assumed to depend logarithmically on N , see Chapter II.5.2. Already in the same article, the authors observed in the numerical examples that this bound on k_w seems to be too pessimistic and one could possibly choose

$$k_w = c_{\eta \rightarrow w} k_\eta, \tag{III.2.2}$$

where $c_{\eta \rightarrow w} \in [2, 3.5]$.

Unfortunately, weak admissibility is not suitable for dimensions greater than one due to the fact that clusters can possibly intersect each other in $\mathcal{O}(N^\alpha)$ points, where $\alpha \geq 0$ depends on the spatial dimension. However, one can try to reduce this negative influence of the weak admissibility condition by mixing it with η -admissibility. In the software package HLib, cf. [BGH03], the authors use the η -admissibility for all block-clusters with block size larger than a given threshold. They apply the weak admissibility condition for block-clusters $\tau \times \sigma$, which are below that threshold, provided that the condition

$$a_i^\tau < \frac{a_i^\sigma + b_i^\sigma}{2} < b_i^\tau \quad \text{or} \quad a_i^\sigma < \frac{a_i^\tau + b_i^\tau}{2} < b_i^\sigma \tag{III.2.3}$$

is satisfied for the corresponding bounding-boxes $\prod_{i=1}^3 [a_i^\mu, b_i^\mu]$, $\mu = \tau, \sigma$, in at most one coordinate direction. This condition restricts the application of weak admissibility to essentially one-dimensional cluster intersections with a length below a certain threshold. The impact of this specific admissibility condition is illustrated in Figure III.2.

III.2.2 Nested Dissection

While weak admissibility only takes the sparsity of the finite element matrix into account during the construction of the \mathcal{H} -matrix, it is also possible to consider the sparsity already during the construction of the cluster tree. One possibility was introduced in [GKLB09] and is based on nested dissection, cf. [BT02, Geo73, HR98, LRT79] and the references therein. We briefly review the idea of nested dissection in the context of \mathcal{H} -matrices as discussed in [GKLB09] and refer the reader to [GKLB09] for more details. The idea is to employ a recursive algorithm as listed in Algorithm III.1.

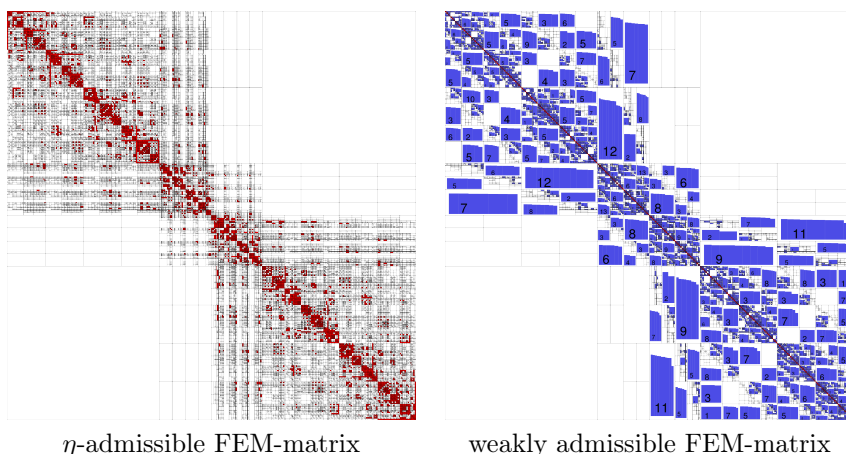


Figure III.2: Comparison of the partition for η -admissibility and for weak admissibility. Red blocks correspond to full matrices, blue blocks correspond to low-rank matrices with inscribed rank, and white blocks are zero.

Algorithm III.1 Nested dissection

function NESTEDDISSECTION(\mathcal{I})

Build partition $\mathcal{I} = \mathcal{I}_1 \dot{\cup} \mathcal{I}_2 \dot{\cup} \mathcal{I}_3$ such that

- \mathcal{I}_1 and \mathcal{I}_2 have comparable sizes,
- all entries \mathbf{A}_{ij} and \mathbf{A}_{ji} , $i \in \mathcal{I}_1$, $j \in \mathcal{I}_2$ of the finite element matrix are zero,
- \mathcal{I}_3 is the boundary layer between \mathcal{I}_1 and \mathcal{I}_2 .

Relabel indices in subsequent order: first \mathcal{I}_1 , then \mathcal{I}_2 , and then \mathcal{I}_3 .

NESTEDDISSECTION(\mathcal{I}_1)

NESTEDDISSECTION(\mathcal{I}_2)

Set $\mathcal{I} = \mathcal{I}_1 \dot{\cup} \mathcal{I}_2 \dot{\cup} \mathcal{I}_3$

▷ Later: extra procedure for \mathcal{I}_3 in \mathcal{H} -matrix framework

end function

Reordering the index sets of the finite element matrix in accordance with this procedure yields a sparsity pattern as illustrated in Figure III.3. Due to the special construction, large parts of the matrix are zero and will remain zero in a subsequent LU factorization.

In the following, we take the approach of [GKLB09] to construct an \mathcal{H} -matrix which reorders the index set such that the pattern of the finite element matrix exposes a nested dissection ordering. We therefore recapitulate the construction of a cluster tree based on domain decomposition as proposed in [GKLB09]. For that purpose, the cluster algorithm distinguishes between *domain clusters* and *interface clusters*. The algorithm listed in Algorithm III.2 is employed for the root $\{1, \dots, N\}$ and all domain clusters.

Due to their special construction, the bounding-boxes of interface clusters are “flat” in one coordinate direction. Therefore, an unmodified cluster algorithm such as Algorithm II.2 leads to a cluster tree with unfavourable asymptotic properties and the cluster algorithm has to be adapted. To achieve more desirable asymptotic properties, we define $\text{level}_{\text{int}}(\tau)$ as the distance of τ to the nearest domain cluster in the cluster tree and employ the cluster algorithm listed in Algorithm III.3 for the interface clusters, see [GKLB09] for a detailed discussion on this topic. We remark that, strictly speaking, the generated cluster tree is not a tree in the sense of Definition II.5.2, since a cluster may have itself as its only son, see also Remark II.5.4. However, due to the recursive structure of the cluster algorithm, it is straightforward how to deal with this situation.

In order to translate the sparsity of the finite element matrix into the block structure of an \mathcal{H} -matrix, we can combine the η -admissibility condition from Definition II.5.6 and the weak admissibility condition from Definition III.2.1 into a nested dissection admissibility condition.

Algorithm III.2 Cluster algorithm for domain clusters, see [GKLB09, Section 3.2]

function \mathcal{H} NESTEDDISSECTIONDOMAIN(τ)
 $Q_\tau = \text{BUILDBOUNDINGBOX}(\tau)$ ▷ See Algorithm II.1
 Cut Q_τ into two pieces Q_1 and Q_2 by halving the longest edge.
 Set $\tau_1 = \{i \in \tau : \xi_i \in Q_1\}$.
 Set $\tau_2 = \{i \in \tau : \mathbf{A}_{ij} = 0 \text{ and } \mathbf{A}_{ji} = 0 \text{ for all } j \in \tau_1\}$.
 Set $\tau_3 = \tau \setminus \{\tau_1 \cup \tau_2\}$.
 Set $\text{sons}(\tau) = \{\tau_1, \tau_2, \tau_3\}$.
 Relabel indices of clusters in subsequent order: first τ_1 , then τ_2 , and then τ_3 .
 \mathcal{H} NESTEDDISSECTIONDOMAIN(τ_1)
 \mathcal{H} NESTEDDISSECTIONDOMAIN(τ_2)
 \mathcal{H} NESTEDDISSECTIONINTERFACE(τ_3) ▷ See Algorithm III.3 below
end function

Algorithm III.3 Cluster algorithm for interface clusters, see [GKLB09, Section 3.2]

function \mathcal{H} NESTEDDISSECTIONINTERFACE(τ)
if $\text{level}_{\text{int}}(\tau) = 0 \pmod{d}$ **then**
 Do not subdivide τ , and set $\tau' = \tau$ as its only son.
else
 $Q_\tau = \text{BUILDBOUNDINGBOX}(\tau)$ ▷ See Algorithm II.1
 Cut Q_τ into two pieces Q_1 and Q_2 by halving one of the “non-flat” edges.
 Set $\tau_1 = \{i \in \tau : \xi_i \in Q_1\}$.
 Set $\tau_2 = \{i \in \tau : \xi_i \in Q_2\}$.
 Set $\text{sons}(\tau) = \{\tau_1, \tau_2\}$.
end if
 Apply \mathcal{H} NESTEDDISSECTIONINTERFACE(τ') to all $\tau' \in \text{sons}(\tau)$.
end function

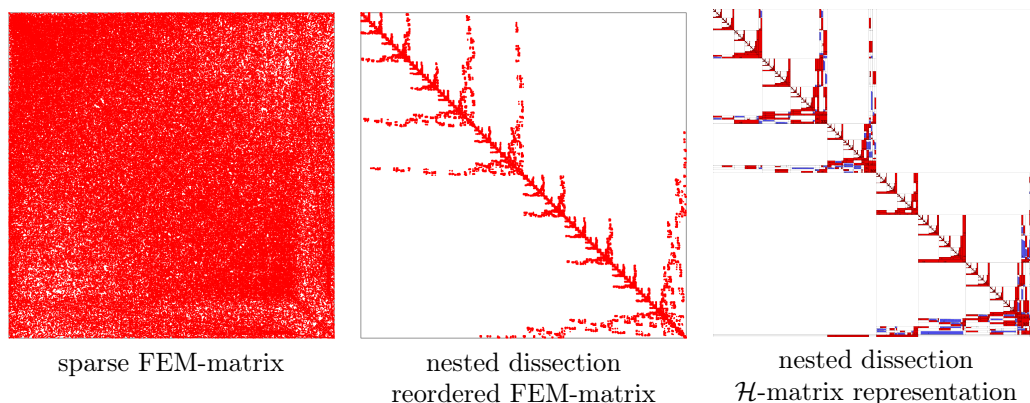


Figure III.3: Sparsity pattern of a three-dimensional finite element matrix, its reordered finite element matrix according to nested dissection, and its corresponding \mathcal{H} -matrix.

Definition III.2.2. Two clusters τ and σ are called nd-admissible if either

- $\tau \neq \sigma$ are both domain clusters or
- τ and σ are η -admissible.

In fact, if $\tau \neq \sigma$ are both domain clusters, we can directly say that the corresponding \mathcal{H} -matrix block has rank zero. Figure III.3 illustrates the sparsity pattern of the finite element matrix after the permutations, determined by the cluster algorithms, and how large parts of the constructed \mathcal{H} -matrix have rank zero. The low-rank blocks in the representation are due to some internal checks of HLib, which aim at replacing inadmissible blocks by low-rank matrices only if very few entries in the corresponding matrix block are non-zero.

The numerical experiments at the end of this chapter show that the sparse structure constructed here leads to smaller constants in the complexity of the solution algorithm.

III.3 Iterative Solution

It remains to discuss how to implement an efficient solver for the \mathcal{H} -matrix equations (II.4.4) and (III.1.3), respectively. For two \mathcal{H} -matrices $\mathbf{H}_1, \mathbf{H}_2 \in \mathcal{H}(\mathcal{B}, k)$, we have discussed the formatted addition $\mathbf{H}_1 + \mathbf{H}_2 \in \mathcal{H}(\mathcal{B}, k)$ and the formatted multiplication $\mathbf{H}_1 * \mathbf{H}_2 \in \mathcal{H}(\mathcal{B}, k)$ as well as the \mathcal{H} -LU factorization in Chapter II.6. Having these operations at hand, we may consider an iterative solver based on the *iterative refinement* method, see [GVL12, Mol67, Wil63]. It was originally introduced in [Wil63] for the improvement of solutions to systems of linear equations computed by using an LU factorization and algebraically coincides with an undamped preconditioned Richardson iteration, see, e.g., [Saa03].

Having all matrices in (III.1.3) represented by \mathcal{H} -matrices, the solution can be approximated as follows. Let $\mathbf{A} \approx \hat{\mathbf{L}}\hat{\mathbf{U}}$, where $\hat{\mathbf{L}}, \hat{\mathbf{U}} \in \mathcal{H}(\mathcal{B}, k)$, be an approximate LU factorization to \mathbf{A} , e.g. computed from \mathbf{A} using Algorithm II.13. Starting with the initial guess $\mathbf{C}_u^{(0)} = \hat{\mathbf{U}}^{-1}\hat{\mathbf{L}}^{-1}\mathbf{C}_f\hat{\mathbf{L}}^{-\top}\hat{\mathbf{U}}^{-\top}$, we iterate

$$\Theta^{(i)} = \mathbf{C}_f - \mathbf{A}\mathbf{C}_u^{(i)}\mathbf{A}^\top, \quad \mathbf{C}_u^{(i+1)} = \mathbf{C}_u^{(i)} + \hat{\mathbf{U}}^{-1}\hat{\mathbf{L}}^{-1}\Theta^{(i)}\hat{\mathbf{L}}^{-\top}\hat{\mathbf{U}}^{-\top}, \quad i = 0, 1, \dots \quad (\text{III.3.1})$$

The idea of the algorithm is that the residual $\Theta^{(i)}$ is computed with a higher precision than the correction $\hat{\mathbf{U}}^{-1}\hat{\mathbf{L}}^{-1}\Theta^{(i)}\hat{\mathbf{L}}^{-\top}\hat{\mathbf{U}}^{-\top}$, which yields an improved approximation to the solution in each step. Note that we use, in contrast to the first algorithm in [DHP15], the LU factorization with forward and backward substitution. If \mathbf{A} is symmetric and positive definite, the LU factorization could also be replaced by a Cholesky factorization. Nonetheless, we will see in the numerical experiments that the computation time of the factorization is negligible compared to the overall computation time, and we prefer to stay in the more general, i.e., non-symmetric, setting.

III.4 Numerical Examples

Before we summarize the settings of the numerical experiments, we briefly recall that the algorithm of the \mathcal{H} -matrix approach to correlation equations from this chapter consists of the following three steps.

1. Compute the sparse finite element matrix \mathbf{A} in linear and the correlation \mathcal{H} -matrix \mathbf{C}_f in almost linear time.
2. Compute the approximate LU factorization of \mathbf{A} in the \mathcal{H} -matrix format in almost linear time.
3. Solve the matrix equation (III.1.3) with iterative refinement (III.3.1) in almost linear time for each iteration.

The numerical experiments in this thesis shall mainly focus on the third step and the overall behaviour of the method. We will see that only one iteration is required in the third step, which yields an almost linear overall complexity of the algorithm. To improve the computation time, we exploit symmetry and store and compute only the lower triangular part of \mathbf{C}_f and \mathbf{C}_u .

All the computations in the following experiments were carried out on a single core of a computing server with two Intel(R) Xeon(R) E5-2670 CPUs with a clock rate of 2.60GHz and a main memory of 256GB. For the \mathcal{H} -matrix computations, we use the software package HLib and, for the finite element discretization, we use the Partial Differential Equation Toolbox of MATLAB¹, which employs piecewise linear finite elements. The two libraries are coupled together in a single C-program, see [KR88], using the MATLAB Engine interface. The meshes are generated by TETGEN, see [Si15], and then imported into MATLAB.

III.4.1 Experimental Setup

To obtain computational efficiency and to keep the ranks of the low-rank matrices under control, HLib imposes, in accordance with Remark II.6.5, an upper threshold in terms of the ε -rank and a prescribed maximal rank. In the case of an η -admissible \mathcal{H} -matrix, the maximal rank is denoted as k_η . A lower threshold n_{\min} is imposed for the minimal block size. For the application of the weak admissibility condition, we rely on the criterion of HLib, which considers the weak admissibility condition only if one of the index sets of the block-cluster $\tau \times \sigma$ has a cardinality below 1,024 and the condition (III.2.3) is satisfied for the corresponding bounding boxes $\prod_{i=1}^3 [a_i^\mu, b_i^\mu]$, $\mu = \tau, \sigma$, in at most one coordinate direction. Otherwise, η -admissibility is used instead. In the case of a weakly admissible matrix block, HLib imposes an upper threshold of $k_w = 3k_\eta$ instead of k_η for the rank, setting, in accordance with [HKK04], $c_{\eta \rightarrow w} = 3$ in (III.2.2). For our experiments, we choose $\eta = 2$, $k_\eta = 20$, $\varepsilon = 10^{-8}$, and $n_{\min} = 50$, and employ either a geometric cluster strategy, i.e., the binary cluster strategy from Algorithm II.2, or the nested dissection cluster strategy from Algorithms III.2 and III.3. The iterative refinement is stopped if the absolute error of the residual in the Frobenius norm is smaller than 10^{-6} .

In the following examples, we want to study, in addition to other aspects, the influence of the weak admissibility condition and the nested dissection clustering for the partitioning of the different \mathcal{H} -matrices. Namely, we successively want to replace η -admissibility by weak admissibility for a binary cluster tree and a nested dissection cluster tree, as described in Table III.1, in order to lower the constants hidden in the complexity of the \mathcal{H} -matrix arithmetic and, thus, to improve the computation time. For the discretization of the correlation kernel $\text{Cor}[f]$, we will always use adaptive cross approximation, see [Beb00].

Whereas the *all- η* case is the canonical case, the *weak-FEM* case is a first relaxation to apply the weak admissibility condition. This is justified, since the stiffness matrix \mathbf{A} can exactly be represented as a weakly admissible \mathcal{H} -matrix and the iterative refinement only

¹Release 2015b, The MathWorks, Inc., Natick, MA.

Case	Operator and admissibility		Cluster tree
	\mathcal{L} and \mathcal{L}^{-1}	Cor_f and Cor_u	
all- η	η -admissibility	η -admissibility	binary
weak-FEM	weak admissibility	η -admissibility	binary
all-weak	weak admissibility	weak admissibility	binary
nd- η	nd-admissibility	η -admissibility	nested dissection
nd-weak	nd-admissibility	weak-admissibility	nested dissection

Table III.1: The five combinations of admissibility conditions used for the partition of \mathcal{H} -matrices.

involves an approximate LU factorization. Hence, we expect at most an influence on the quality of the approximate LU factorization and, thus, on the number of iterations in the iterative refinement. We therefore have to investigate if possible additional iterations are compensated by the faster \mathcal{H} -matrix arithmetic.

The aforementioned cases have in common that they rely on the asymptotic smoothness of $\text{Cor}[f]$ and $\text{Cor}[u]$ and the η -admissibility which leads to exponential convergence of the \mathcal{H} -matrix approximation. In the case *all-weak*, we want to examine if there is some indication that weak admissibility could possibly also be considered for the partition of the \mathcal{H} -matrices for $\text{Cor}[f]$ and $\text{Cor}[u]$. To that end, we approximate $\text{Cor}[f]$ with adaptive cross approximation relative to the η -admissibility partition and convert it on-the-fly to the partition of the weak admissibility, as proposed in [HKK04].

While the three aforementioned cases all rely on a binary cluster tree, the cases *nd- η* and *nd-weak* rely on a cluster tree which is constructed by nested dissection. In both cases, the finite element matrix \mathbf{A} is partitioned by nd-admissibility. For $\text{Cor}[f]$ and $\text{Cor}[u]$, we use η -admissibility in the nd- η case and weak admissibility in the nd-weak case, where we assemble the matrix for $\text{Cor}[f]$ in the same way as in the all-weak case.

The following numerical examples are divided into two parts. In the first part, we demonstrate the convergence of the presented method by comparing it to a low-rank reference solution computed with the pivoted Cholesky factorization, cf. [HPS12a]. In the second part, we will demonstrate that the presented method also works well in the case of correlations with low Sobolev smoothness or small correlation length, where no low-rank approximations exist and sparse tensor product approximations fail to resolve the correlation length. Note that in both examples, since we are also dealing with dense matrices, the computed system matrices are smaller than usual for the finite element method. In particular, the unknown in the system (III.1.3) of linear equations is a matrix with N^2 entries, whereas the corresponding mesh has only N degrees of freedom. The \mathcal{H} -matrix compression reduces the computational complexity for the assembly and the amount of required storage from N^2 to $\mathcal{O}(kN \log N)$, whereas the complexity of the solution algorithm decreases from $\mathcal{O}(N^3)$ to $\mathcal{O}(k^2 N \log^2 N)$.

III.4.2 Tests for the Iterative Solver

Due to the truncation operators involved in the block matrix algorithms of the \mathcal{H} -matrix arithmetic, it is not immediately clear if the presented solver converges. Still, it can be shown that some iterative \mathcal{H} -matrix schemes converge up to a certain accuracy, cf. [HKT08]. In the following, we want to demonstrate for a specific example that our iterative scheme indeed provides convergence.

On the dumbbell geometry pictured in Figure III.4, we consider $\mathcal{L} = -\Delta$ in (III.1.1) and the Matérn-5/2 kernel as input correlation $\text{Cor}[f]$, i.e., for $r = \|\mathbf{x} - \mathbf{y}\|$ and $\nu = 5/2$ in (II.4.5), we set

$$\text{Cor}[f](\mathbf{x}, \mathbf{y}) = \left(1 + \sqrt{5} \frac{r}{\ell} + \frac{5}{3} \frac{r^2}{\ell^2}\right) \exp\left(-\sqrt{5} \frac{r}{\ell}\right),$$

where $\ell \approx \text{diam}(D)$ denotes the correlation length. The conversion of the finite element

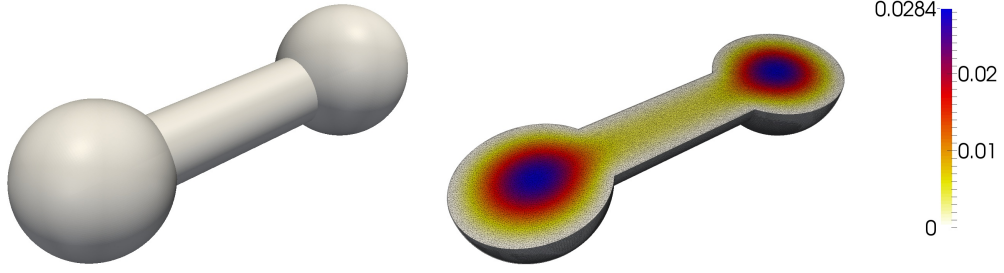


Figure III.4: The dumbbell geometry and its meshed cross section with the diagonal of the solution correlation kernel $\text{Cor}[u]|_{\mathbf{x}=\mathbf{y}}$ for load data prescribed by the Matérn-5/2 kernel.

matrix to an \mathcal{H} -matrix for the dumbbell geometry was already illustrated in Figure III.1. The difference between the η -admissibility and the weak admissibility is illustrated in Figure III.2, and the effect of nested dissection ordering is illustrated in Figure III.3.

For determining a reference solution, we compute a low-rank approximation $\mathbf{C}_f \approx \mathbf{L}_f \mathbf{L}_f^\top$ with the pivoted Cholesky factorization as proposed in [HPS12a]. The numerical solution \mathbf{C}_u of (III.1.3) is then given by

$$\mathbf{C}_u \approx \mathbf{L}_u \mathbf{L}_u^\top,$$

where \mathbf{L}_u solves $\mathbf{A} \mathbf{L}_u = \mathbf{L}_f$. To compute the error of the \mathcal{H} -matrix solution, we compare the correlation and the correlation's diagonal $\text{Cor}[u]|_{\mathbf{x}=\mathbf{y}}$ of the \mathcal{H} -matrix approximation with the correlation and the correlation's diagonal $\text{Cor}[u]|_{\mathbf{x}=\mathbf{y}}$ derived from the pivoted Cholesky factorization on a finer reference mesh. We refer the reader to Table III.2 for more details on the meshes under consideration.

Level	1	2	3	4	5	Reference mesh
Mesh points	238	1,498	6,958	34,112	175,562	1,033,382
N	4	201	1,742	13,341	98,177	756,626
N^2	16	40,401	3,034,564	$1.78 \cdot 10^8$	$9.64 \cdot 10^9$	$5.72 \cdot 10^{11}$

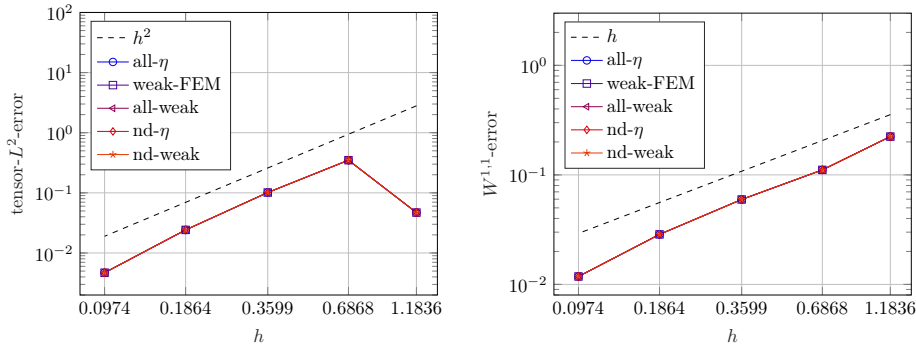
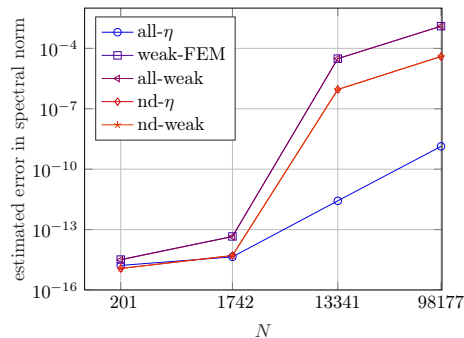
Table III.2: Mesh points and number N of degrees of freedom of the finite element mesh and number N^2 of entries in the correlation matrices for different levels of the dumbbell geometry.

While the error of the correlation itself can be measured in the L^2 -norm on the tensor product domain, the appropriate norm for error measurements of its diagonal is the $W^{1,1}$ -norm, see also [DDH15] for an in-depth discussion on this topic. Due to the Poincaré-Friedrich and the Cauchy-Schwartz inequalities and

$$\begin{aligned} \|u^2 - u_h^2\|_{W^{1,1}(D)} &\lesssim \|\nabla(u^2 - u_h^2)\|_{L^1(D)} \\ &\leq \|(\nabla(u - u_h))(u + u_h)\|_{L^1(D)} + \|(u - u_h)(\nabla(u + u_h))\|_{L^1(D)} \\ &\lesssim |u - u_h|_{H^1(D)} + \|u - u_h\|_{L^2(D)} \\ &\lesssim h, \end{aligned}$$

we can expect a convergence rate in the $W^{1,1}$ -norm which is proportional to the mesh size h . A standard tensor product argument yields a convergence rate of order h^2 in the tensor product L^2 -norm. Figure III.5 shows that we indeed reach these rates for all five cases of admissibility which are considered in Table III.1. In fact, the observed errors coincide in the first few digits.

We are also interested in the quality of the approximate LU factorization $\mathbf{A} \approx \hat{\mathbf{L}} \hat{\mathbf{U}}$. We use a built-in function of HLib to estimate the deviation $\hat{\mathbf{L}} \hat{\mathbf{U}} - \mathbf{A}$ in the spectral norm by ten power iterations, which is a good indicator of the approximation quality of the LU factorization of the finite element matrix. The estimated errors are plotted in Figure III.6.


 Figure III.5: L^2 -error of $\text{Cor}[u]$ (left) and $W^{1,1}$ -error of $\text{Cor}[u]|_{\mathbf{x}=\mathbf{y}}$ (right).

 Figure III.6: The deviation $\hat{\mathbf{L}}\hat{\mathbf{U}} - \mathbf{A}$ in the estimated spectral norm for fixed rank.

Note that the observed behaviour is in contrast to the behaviour typically observed for preconditioning, cf., e.g., [Beb08] and the references therein, since we do not increase the rank with the number of unknowns. We can see that the LU factorization is most accurate in the all- η case. Nonetheless, only one iteration is needed in the iterative refinement in all cases. When it comes to computation times, Figure III.7 and Tables III.3, III.4, and III.5 indicate that all cases of admissibility under consideration might yield essentially linear complexity, although it seems that the asymptotic regime is not reached in the considered levels of refinement. Both, the weak admissibility condition and the nested dissection approach, lead to considerable speed-ups, where the combination of these approaches, the nd-weak case, seems to be the fastest approach. Figure III.8 illustrates the required average and maximal ranks needed for the computations, whereas Figure III.9 illustrates the amount of storage needed per degree of freedom. For reasons of performance, HLib allocates the worst-case scenario for the ranks. Thus, in the latter case, only the different admissibilities for a single \mathcal{H} -matrix, built from a binary cluster tree and a nested dissection cluster tree, have to be considered. In conclusion, nested dissection clustering consumes less computation time and less storage for the LU factorization.

III.4.3 Small Correlation Lengths

Having verified the convergence of our solver, we now want to consider different correlation lengths and different classes of smoothness. Hence, in the second part of the numerical experiments, we employ correlation kernels with smaller correlation lengths and lower regularity such that low-rank approximations would become prohibitively expensive and sparse tensor product approaches would fail to resolve the concentrated measure.

We consider the screw-nut geometry, pictured in Figure III.10, which is discretized by a mesh with 269,950 vertices, 197,480 degrees of freedom, and a maximal element diameter

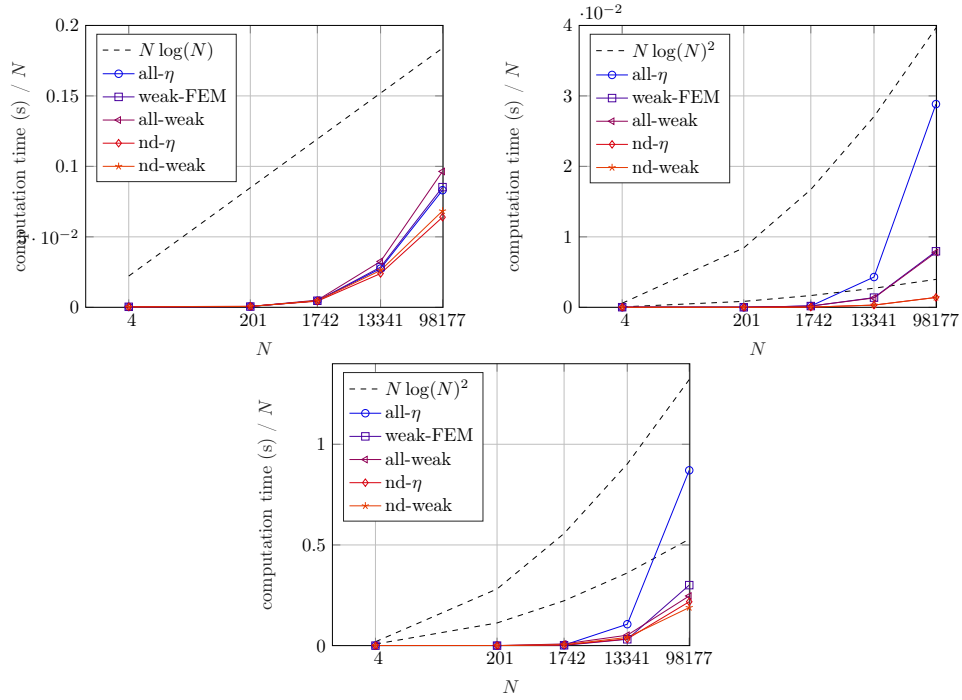


Figure III.7: Computation times in seconds for the computation of the data correlation \mathcal{H} -matrix \mathbf{C}_f (top left), the approximate LU factorization of the system matrix \mathbf{A} (top right), and for the iterative refinement (bottom) on the dumbbell geometry.

Level	1	2	3	4	5
all- η	0.000954	0.123520	8.07820	367.071	8,158.12
weak-FEM	0.001476	0.109641	8.12224	380.533	8,370.79
all-weak	0.001113	0.113122	8.86575	434.289	9,464.19
nd- η	0.001569	0.151018	7.54221	319.773	6,276.75
nd-weak	0.000885	0.123694	8.07407	349.585	6,711.85

Table III.3: Computation times in seconds to compute the data correlation \mathcal{H} -matrix \mathbf{C}_f on the dumbbell geometry.

Level	1	2	3	4	5
all- η	$2.2 \cdot 10^{-5}$	0.001274	0.339432	56.2521	2,806.5
weak-FEM	$2.9 \cdot 10^{-5}$	0.00143	0.315344	17.7348	743.624
all-weak	$2.2 \cdot 10^{-5}$	0.001831	0.316497	18.3358	746.364
nd- η	$2.5 \cdot 10^{-5}$	0.000513	0.048170	4.17870	135.778
nd-weak	$3.6 \cdot 10^{-5}$	0.000588	0.050896	4.05899	132.319

Table III.4: Computation times in seconds to compute the approximate LU factorization of the finite element matrix on the dumbbell geometry.

Level	1	2	3	4	5
all- η	$5.2 \cdot 10^{-5}$	0.011115	4.68355	1,419.19	85,477.9
weak-FEM	0.000104	0.010592	2.64065	420.153	29,492.1
all-weak	$5.4 \cdot 10^{-5}$	0.042098	14.5121	691.129	24,225.6
nd- η	0.000102	0.039209	5.60769	443.310	21,390.0
nd-weak	$4.4 \cdot 10^{-5}$	0.061530	7.67542	544.080	18,570.5

Table III.5: Computation times in seconds for the iterative refinement on the dumbbell geometry.

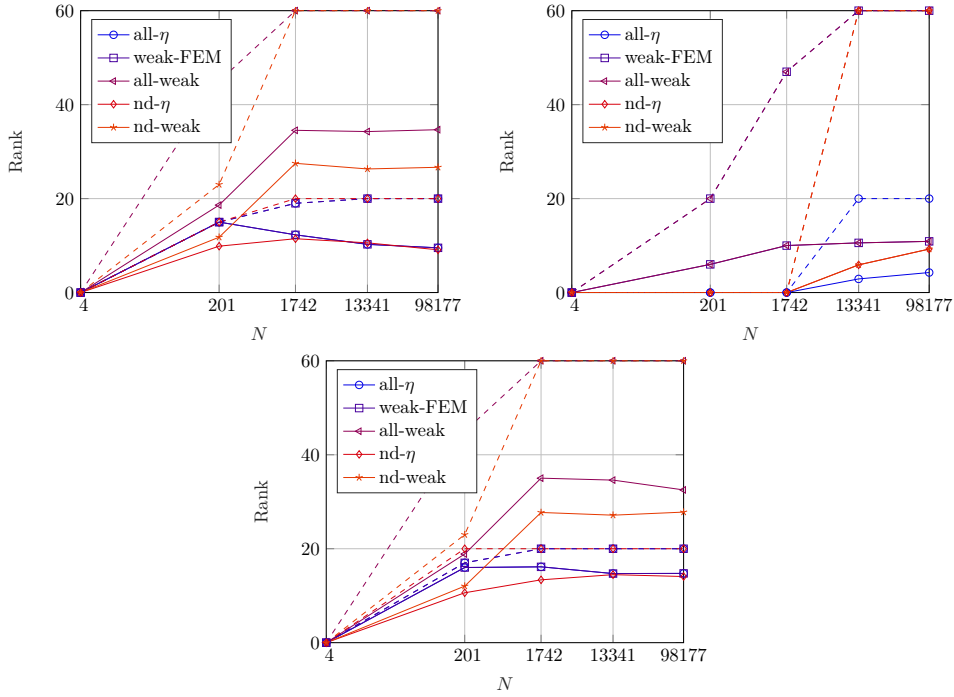


Figure III.8: Required ranks for the prescribed correlation C_f (top left), the LU factorization of \mathbf{A} (top right), and the solution correlation C_u (bottom). The solid lines indicate the average ranks, whereas the dashed lines illustrate the maximal rank attained.

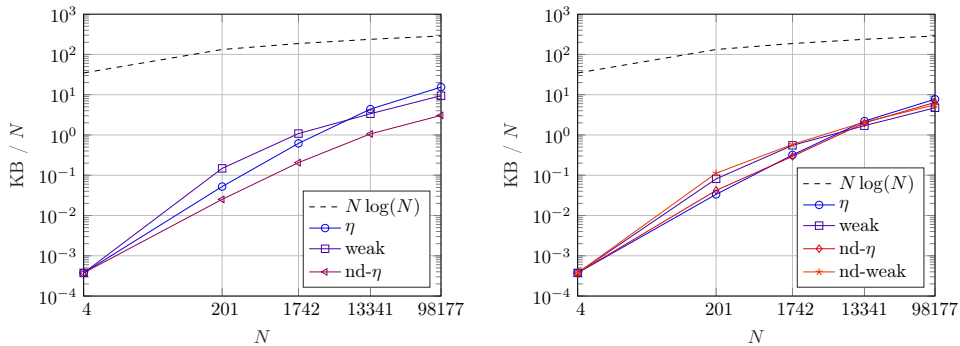


Figure III.9: Allocated storage per degree of freedom for different admissibility conditions for nonsymmetric (left) and symmetric (right) matrices. The allocated storage is independent of the content of the matrix.

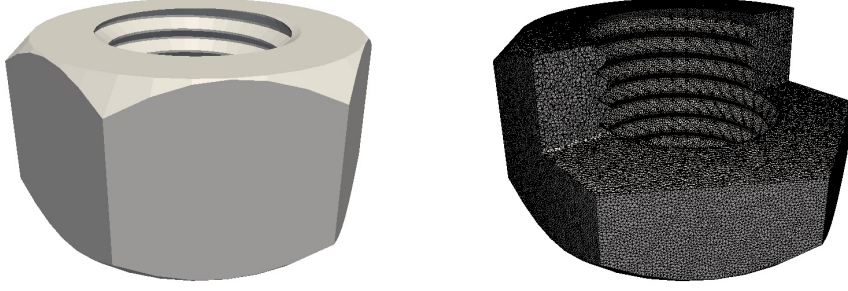


Figure III.10: The screw-nut geometry (left) and its meshed cross section (right).

of $h/\text{diam}(D) \approx 0.0225$, yielding a matrix equation with $3.90 \cdot 10^{10}$ unknowns. We choose $\mathcal{L} = -\Delta$ in (III.1.1) and as input correlation $\text{Cor}[f]$ we choose either the Gaussian kernel, i.e., letting $\nu \rightarrow \infty$ in (II.4.5),

$$\text{Cor}[f](\mathbf{x}, \mathbf{y}) = \frac{1}{\ell} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\ell^2}\right),$$

or the exponential kernel, i.e., setting $\nu = 1/2$ in (II.4.5),

$$\text{Cor}[f](\mathbf{x}, \mathbf{y}) = \frac{1}{\ell} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|}{\ell}\right).$$

In the following, we want to demonstrate that the presented method is well suited for small correlation lengths $\ell > 0$. We therefore choose the correlation lengths

$$\ell \in \left\{ \frac{\text{diam}(D)}{1}, \frac{\text{diam}(D)}{2}, \frac{\text{diam}(D)}{4}, \frac{\text{diam}(D)}{8}, \frac{\text{diam}(D)}{16}, \frac{\text{diam}(D)}{32} \right\}$$

for both the Gaussian kernel and the exponential kernel, and compute the corresponding correlation of the solution $\text{Cor}[u]$. A visualization of the Gaussian kernel for different ℓ without the rescaling was given in Figure II.2.

In our first test, we use the nd-weak case, as the previous section has shown that it is more memory efficient and has superior computation times. The computation time for the assembly of the prescribed correlation is about 20,000 seconds, and the computation time of the approximate LU factorization is about 400 seconds, whereas the computation times for the iterative refinement are contained in Table III.6.

	$\ell/\text{diam}(D)$	1	1/2	1/4	1/8	1/16	1/32
Exponential	nd-weak	51,656.7	53,011.0	52,876.5	51,459.2	49,838.2	51,524.6
	all-weak	77,784.5	79,101.8	79,155.3	79,155.3	76,952.6	72,256.9
Gaussian	nd-weak	47,921.8	50,644.0	50,819.5	51,753.7	—	—
	all-weak	73,405.4	74,877.0	75,165.7	68,222.8	72,259.4	75,070.4

Table III.6: Computation times in seconds for the nd-weak case and the all-weak case for the iterative refinement on the screw-nut geometry in case of the exponential kernel and the Gaussian kernel with different correlation lengths.

We do not tabulate the computation times for the Gaussian kernel in case of the correlation lengths $\ell/\text{diam}(D) = 1/16$ and $\ell/\text{diam}(D) = 1/32$, since the iterative refinement does not converge to the prescribed tolerance. In all other cases, the iterative refinement needs only one iteration.

Repeating the computations in the two problematic cases with increased k_η or in the $\text{nd-}\eta$ instead of the nd-weak case also does not lead to convergence. However, repeating all computations in the all-weak case resolves the issue, as the computation times in Table III.6 show. In the all-weak case, the computation time for the prescribed correlation is again around 20,000 seconds, and the computation time for the approximate LU factorization is around 1,700 seconds. The iterative refinement again needs one iteration in all tabulated cases.

The cross sections found in Figure III.11 illustrate the different behaviour of the correlation's diagonal $\text{Cor}[u]_{\mathbf{x}=\mathbf{y}}$ for the different correlation lengths in case of the exponential kernel. The related results for the Gaussian kernel are presented in Figure III.12. It seems that a mass defect occurs for the correlation lengths $\ell/\text{diam}(D) = 1/16$ and $\ell/\text{diam}(D) = 1/32$, i.e., we roughly find an decrease of the diagonal of the correlation although we should find an increase in those two cases. This could be due to the fact that the mesh size of the finite element method is not able to resolve the correlation length properly. Nevertheless, the computation times are independent of ℓ , although the underlying finite element method cannot resolve the correlation length. Moreover, the nested dissection clustering technique can lead to a speed-up, while the binary clustering technique seems to be more robust.

III.5 Conclusion

We considered the solution of strongly elliptic partial differential equations with random load by means of the finite element method. We employed the \mathcal{H} -matrix technique to efficiently discretize the non-local correlation kernel of the data and to approximate the LU factorization of the finite element stiffness matrix. The resulting \mathcal{H} -matrix equation was then efficiently solved in essentially linear complexity by the \mathcal{H} -matrix arithmetic.

Compared to sparse tensor product or low-rank approximations, the proposed method does not suffer from large constants in the complexity estimates or lack of resolving of the roughness in the case of shortly correlated data. This was shown by numerical experiments on a nontrivial three-dimensional geometry. Indeed, neither the computation times nor the storage requirements increase for correlation kernels with short correlation length. It was moreover demonstrated that the use of the weak admissibility condition for the partition of the \mathcal{H} -matrix improves the constants in the computational complexity without having a significant impact on the solution accuracy. The application of a nested dissection clustering strategy can additionally lead to a speed-up of computation and save storage, whereas the binary clustering strategy seems to be the more robust approach.

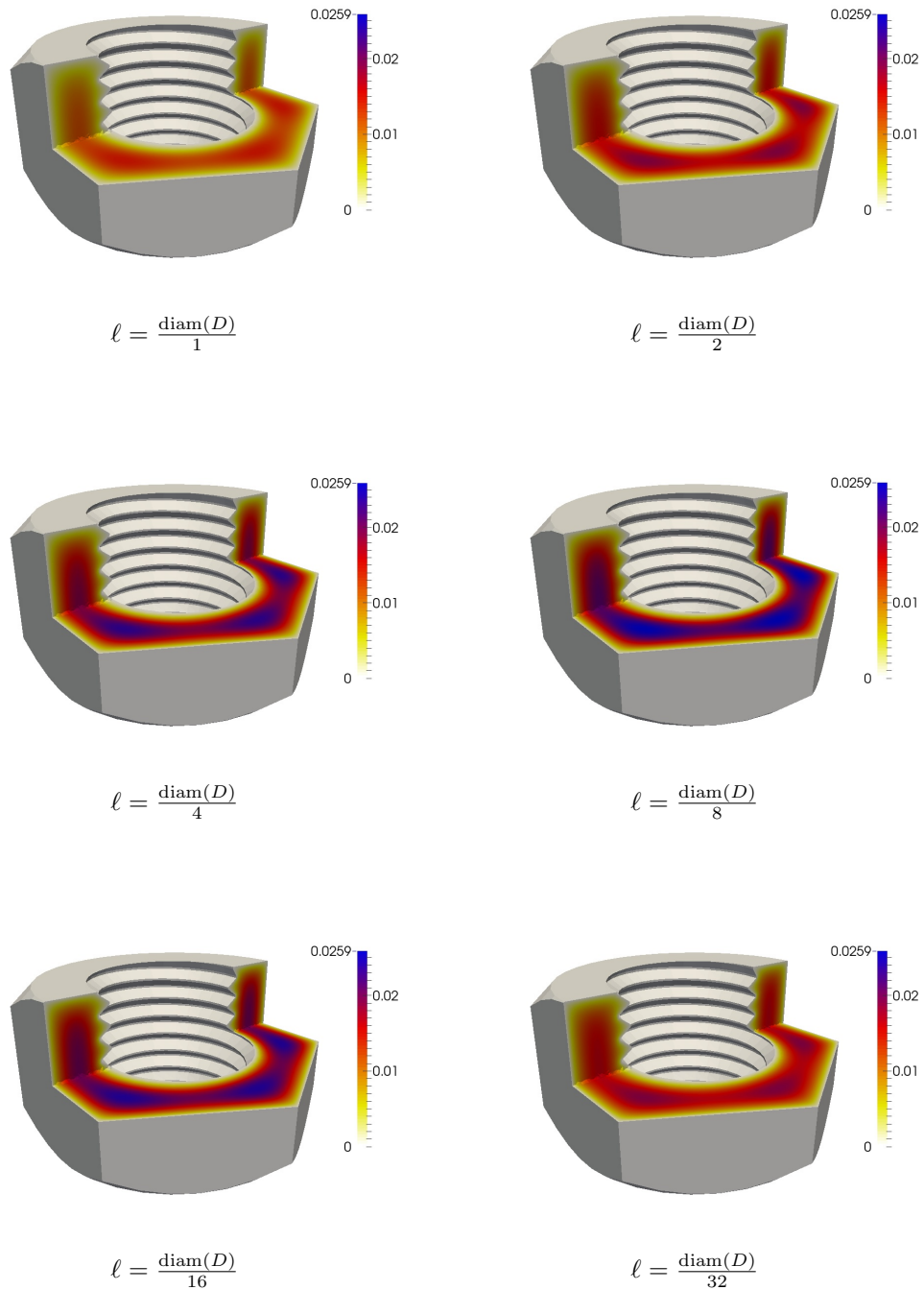


Figure III.11: Cross sections of the diagonal of the correlation of the solution through the screw-nut geometry for the exponential kernel with different correlation lengths ℓ .

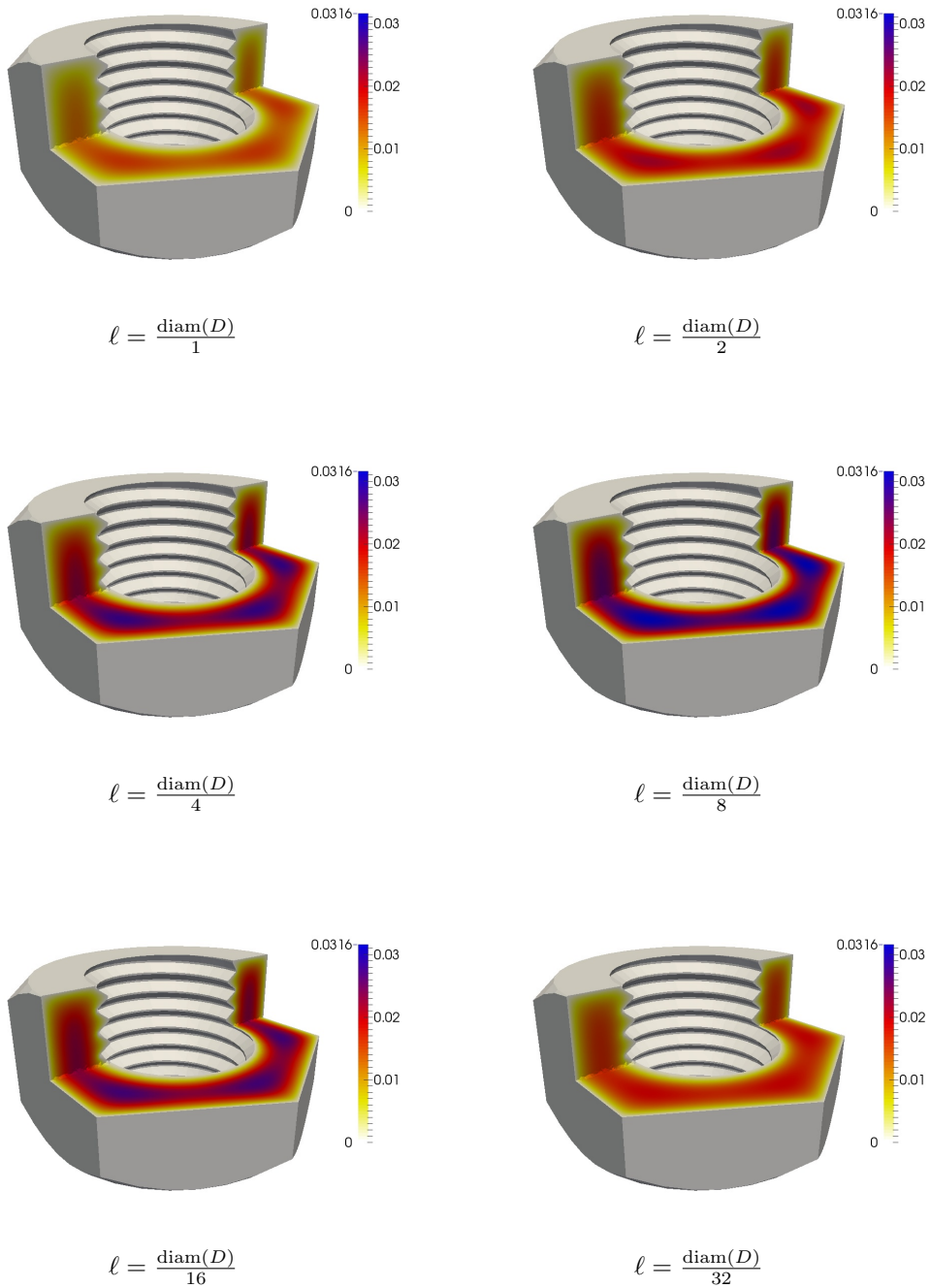


Figure III.12: Cross sections of the diagonal of the correlation of the solution through the screw-nut geometry for the Gaussian kernel with different correlation lengths ℓ .

Chapter IV

The Fast \mathcal{H}^2 -multipole Method on Parametric Surfaces

IV.1 Boundary Integral Equations

The previous chapter was concerned with the uncertainty quantification of the solution of a partial differential equation. As the discretization of the PDE was accomplished by using the finite element method, a mesh had to be generated on the whole computational domain. If a Green's function of the underlying differential operator is known, i.e., a function fulfilling (III.2.1) without the boundary conditions, the boundary value problem may be reformulated by means of boundary integral equations. Before we recall the major concepts of this procedure, we remark that it has the major advantage of reducing the problem's dimensionality.

To keep notation straight, we consider the Laplace equation

$$-\Delta u(\mathbf{x}) = 0 \quad \text{for } \mathbf{x} \in D \quad (\text{IV.1.1})$$

in a bounded three-dimensional Lipschitz domain $D \subset \mathbb{R}^3$ and refer to [SS11, Ste08] for more general differential operators and non-homogeneous right-hand sides. The Green's function to the Laplace operator in three dimensions is given by

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi\|\mathbf{x} - \mathbf{y}\|}.$$

Having the Green's function available, the solution to (IV.1.1) can then be expressed using its Cauchy data by the *representation formula*

$$u(\mathbf{x}) = \int_{\partial D} \frac{1}{4\pi\|\mathbf{x} - \mathbf{y}\|} \frac{\partial u}{\partial \mathbf{n}_{\mathbf{y}}}(\mathbf{y}) \, d\sigma_{\mathbf{y}} - \int_{\partial D} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{n}_{\mathbf{y}} \rangle}{4\pi\|\mathbf{x} - \mathbf{y}\|^3} u(\mathbf{y}) \, d\sigma_{\mathbf{y}} \quad \text{for } \mathbf{x} \in D, \quad (\text{IV.1.2})$$

see, e.g., [Ste08, Chapter 5.1]. The first boundary integral operator of the representation formula has the following properties, see also [Ste08, Lemma 6.6].

Theorem IV.1.1. *For any $\rho \in H^{-1/2}(\partial D)$, the single layer potential*

$$(\tilde{\mathcal{V}}\rho)(\mathbf{x}) := \int_{\partial D} \frac{\rho(\mathbf{y})}{4\pi\|\mathbf{x} - \mathbf{y}\|} \, d\sigma_{\mathbf{y}} \quad \text{for } \mathbf{x} \in \mathbb{R}^3 \setminus \partial D \quad (\text{IV.1.3})$$

satisfies the Laplace equation (IV.1.1) in $\mathbb{R}^3 \setminus \partial D$. Moreover, it is a continuous mapping

$$\tilde{\mathcal{V}}: H^{-1/2}(\partial D) \rightarrow H^1(D).$$

The second boundary integral operator has similar properties, see also [Ste08, Lemma 6.10].

Theorem IV.1.2. *For any $\rho \in H^{1/2}(\partial D)$, the double layer potential*

$$(\tilde{\mathcal{K}}\rho)(\mathbf{x}) := \int_{\partial D} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{n}_{\mathbf{y}} \rangle \rho(\mathbf{y})}{4\pi \|\mathbf{x} - \mathbf{y}\|^3} d\sigma_{\mathbf{y}} \quad \text{for } \mathbf{x} \in \mathbb{R}^3 \setminus \partial D \quad (\text{IV.1.4})$$

satisfies the Laplace equation (IV.1.1) in $\mathbb{R}^3 \setminus \partial D$. Moreover, it is a continuous mapping

$$\tilde{\mathcal{K}}: H^{1/2}(\partial D) \rightarrow H^1(D).$$

We may use the single layer and the double layer potential to solve (IV.1.1) with either Dirichlet boundary conditions

$$\begin{aligned} -\Delta u(\mathbf{x}) &= 0 & \text{for } \mathbf{x} \in D, \\ u(\mathbf{x}) &= f(\mathbf{x}) & \text{for } \mathbf{x} \in \partial D, \end{aligned} \quad (\text{IV.1.5})$$

or Neumann boundary conditions

$$\begin{aligned} -\Delta u(\mathbf{x}) &= 0 & \text{for } \mathbf{x} \in D, \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) &= g(\mathbf{x}) & \text{for } \mathbf{x} \in \partial D. \end{aligned} \quad (\text{IV.1.6})$$

While there exists a unique solution to (IV.1.5), see, e.g., [Ste08, Chapter 4.1.1], the solution of (IV.1.6) is only unique up to a constant and exists only when the compatibility condition

$$\int_{\partial D} g(\mathbf{x}) d\sigma_{\mathbf{x}} = 0$$

is satisfied, see also [Ste08, Chapter 4.1.3].

Since the single layer potential satisfies (IV.1.1), we may make the ansatz

$$u(\mathbf{x}) = (\tilde{\mathcal{V}}\rho)(\mathbf{x}), \quad \mathbf{x} \in D. \quad (\text{IV.1.7})$$

This leads to a Fredholm integral equation of the first kind for the Dirichlet problem,

$$(\mathcal{V}\rho)(\mathbf{x}) = \int_{\partial D} \frac{\rho(\mathbf{y})}{4\pi \|\mathbf{x} - \mathbf{y}\|} d\sigma_{\mathbf{y}} = f(\mathbf{x}), \quad \mathbf{x} \in \partial D, \quad (\text{IV.1.8})$$

and to a Fredholm integral equation of the second kind for the Neumann problem,

$$\frac{1}{2}\rho(\mathbf{x}) + (\mathcal{K}^*\rho)(\mathbf{x}) = \frac{1}{2}\rho(\mathbf{x}) + \int_{\partial D} \frac{\langle \mathbf{n}_{\mathbf{x}}, \mathbf{y} - \mathbf{x} \rangle \rho(\mathbf{y})}{4\pi \|\mathbf{x} - \mathbf{y}\|^3} d\sigma_{\mathbf{y}} = g(\mathbf{x}), \quad \mathbf{x} \in \partial D, \quad (\text{IV.1.9})$$

for the unknown *density* ρ , see [Ste08, Chapter 7.1 and 7.2].

We may also make an ansatz with the double layer potential

$$u(\mathbf{x}) = (\tilde{\mathcal{K}}\rho)(\mathbf{x}), \quad \mathbf{x} \in D. \quad (\text{IV.1.10})$$

This leads to a Fredholm integral equation of the second kind for the Dirichlet problem,

$$\frac{1}{2}\rho(\mathbf{x}) - (\mathcal{K}\rho)(\mathbf{x}) = \frac{1}{2}\rho(\mathbf{x}) - \int_{\partial D} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{n}_{\mathbf{y}} \rangle \rho(\mathbf{y})}{4\pi \|\mathbf{x} - \mathbf{y}\|^3} d\sigma_{\mathbf{y}} = f(\mathbf{x}), \quad \mathbf{x} \in \partial D, \quad (\text{IV.1.11})$$

and to a Fredholm integral equation of the first kind for the Neumann problem,

$$(\mathcal{W}\rho)(\mathbf{x}) = \frac{\partial}{\partial \mathbf{n}_{\mathbf{x}}} \int_{\partial D} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{n}_{\mathbf{y}} \rangle \rho(\mathbf{y})}{4\pi \|\mathbf{x} - \mathbf{y}\|^3} d\sigma_{\mathbf{y}} = g(\mathbf{x}), \quad \mathbf{x} \in \partial D, \quad (\text{IV.1.12})$$

for the unknown density ρ , see [Ste08, Chapter 7.1 and 7.2].

The following theorem is a collection of results from [Ste08, Chapter 6] and recalls important properties of the just introduced boundary integral operators.

Theorem IV.1.3. *The previously introduced boundary integral operators are well defined and have the following mapping properties:*

- $\mathcal{V}: H^{-1/2}(\partial D) \rightarrow H^{1/2}(\partial D)$ is called the single layer operator.
- $\mathcal{K}^*: H^{-1/2}(\partial D) \rightarrow H^{-1/2}(\partial D)$ is called the adjoint double layer operator.
- $\mathcal{K}: H^{1/2}(\partial D) \rightarrow H^{1/2}(\partial D)$ is called the double layer operator.
- $\mathcal{W}: H^{1/2}(\partial D) \rightarrow H^{-1/2}(\partial D)$ is called the hypersingular operator.

It is worth mentioning that the numerical treatment of the hypersingular operator requires special care, which is the topic of Chapter IV.7.1.

Being assured of the well definedness of the integral equations (IV.1.8), (IV.1.9), (IV.1.11), and (IV.1.12), we may discuss the existence of solutions. The following lemma is [Ste08, Theorem 6.22 and 6.24 and (6.38)].

Lemma IV.1.4. *The single layer operator and the hypersingular operator yield elliptic bilinear forms on the following spaces:*

1. The bilinear form $(\mathcal{V}u, v)_{L^2(\partial D)}$ is $H^{-1/2}(\partial D)$ -elliptic.
2. The bilinear form $(\mathcal{W}u, v)_{L^2(\partial D)}$ is $H_\star^{1/2}(\partial D)$ -elliptic, with

$$H_\star^{1/2}(\partial D) := \{v \in H^{1/2}(\partial D) : (v, 1)_{L^2(\partial D)} = 0\}.$$

As a consequence, the integral equations (IV.1.8) and (IV.1.12) are uniquely solvable in the corresponding spaces due to the Lax-Milgram theorem. However, we may remark that the hypersingular operator is only semi-elliptic on $H^{1/2}(\partial D)$, see [Ste08, Corollary 6.25].

In contrast, the integral equations of the second kind are not elliptic. Instead, one ensures the existence of the solutions to (IV.1.9) and (IV.1.11) with an argument based on Neumann series, see [Ste08, Chapter 7.1 and 7.2] for an in-depth discussion.

We conclude this section by remarking that the boundary integral equations (IV.1.8), (IV.1.9), (IV.1.11), and (IV.1.12) are referred to as *indirect* methods to solve (IV.1.5) and (IV.1.6). Especially, the intermediate result ρ does not have a physical meaning. Instead, one may use a *direct* approach, i.e., the representation formula (IV.1.2) to represent the solution. The *Dirichlet-to-Neumann map*

$$\mathcal{S} := \mathcal{V}^{-1} \left(\frac{1}{2} + \mathcal{K} \right) : H^{1/2}(\partial D) \rightarrow H^{-1/2}(\partial D) \quad (\text{IV.1.13})$$

is a bounded operator which establishes a connection between the Dirichlet data and the Neumann data. Especially, it fulfils the same ellipticity estimates as the hypersingular operator. We refer to [Ste08, Chapter 6.6.3] for a more detailed discussion.

IV.2 Surface Representation

For the numerical treatment of the boundary integral equations of the previous section, a representation of the domain boundary is required. For that purpose, let $\square := [0, 1]^2$ denote the unit square, which serves as *reference domain*. We then assume that the surface ∂D can be subdivided into several *patches*

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

where, for each patch, there exists a smooth diffeomorphism

$$\gamma_i : \square \rightarrow \Gamma_i \quad \text{with} \quad \Gamma_i = \gamma_i(\square) \quad \text{for } i = 1, 2, \dots, M, \quad (\text{IV.2.1})$$

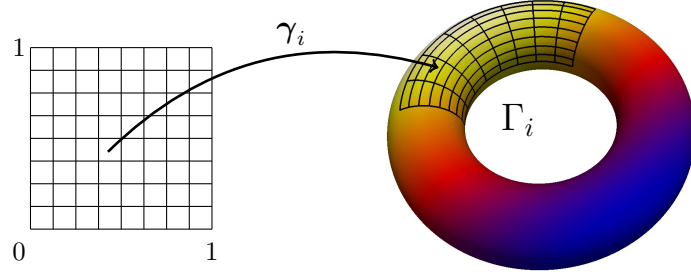


Figure IV.1: Surface representation and mesh generation.

as illustrated in Figure IV.1. Moreover, we assume that the intersection $\Gamma_i \cap \Gamma_j$, $i \neq j$, is either a common edge, a common vertex, or empty.

For constructing regular surface meshes, one has to impose the following *matching condition*: For each $\mathbf{x} = \gamma_i(\mathbf{s})$ on a common edge of Γ_i and Γ_j there has to exist a bijective and affine mapping $\Xi: \square \rightarrow \square$ such that there holds $\gamma_i(\mathbf{s}) = (\gamma_j \circ \Xi)(\mathbf{s})$. This means that the parameterizations γ_i and γ_j coincide on the common edge except for orientation. Figure IV.2 visualizes three parameterizations which satisfy the present requirements.

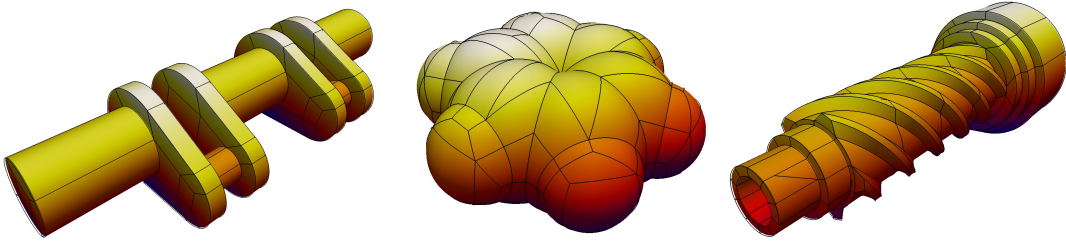


Figure IV.2: Different parametric surfaces with their patch boundaries.

For later reference, in order to pull the kernel functions back to the reference domain, the *surface measure* of the diffeomorphisms γ_i is required. On the patch Γ_i , it is given by

$$\kappa_i(\mathbf{s}) := \|\partial_{s_1} \gamma_i(\mathbf{s}) \times \partial_{s_2} \gamma_i(\mathbf{s})\|. \quad (\text{IV.2.2})$$

One major advantage of parametric surfaces stems from the fact that more geometric information is available, which can therefore be exploited in the discretization. In particular, no difficulties arise if geometric entities occur in the kernel function of the integral operator under consideration, like the normal or tangent vector, as for example in the double layer operator or the adjoint double layer operator. Moreover, parametric surfaces provide an exact representation of the surface, which is in contrast to the common approximation of surfaces by panels. Particularly, there is no further approximation step required if the surface is given in this form. As a consequence, the rate of convergence in a corresponding boundary element method is not limited by the accuracy of the surface approximation.

Many parametric surfaces are nowadays directly accessible as surfaces generated by tools from Computer Aided Design (CAD). Very common surface representations in CAD are defined by the IGES (Initial Graphics Exchange Specification) and the STEP (Standard for the Exchange of Product Model Data) standards, cf. [U.S96, Int14]. In both standards, 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 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, see [HL89]. The representation with NURBS is intensively studied in the context of isogeometric analysis, see, e.g., [CHB09, HCB05, MZBF15, ZMBF14]. Nevertheless, in contrast to the isogeometric analysis framework, the scope of this chapter is not restricted to geometries that can be represented by NURBS, but considers any surface which provides the requirements specified at the beginning of this section.

An algorithm to decompose a technical surface, described in the IGES format, into a collection of parameterized four-sided patches, fulfilling all the above requirements, was proposed in [HR10]. This algorithm was extended in [HR09, HR11] to molecular surfaces.

Starting from this surface representation, it is straightforward to generate a nested sequence of meshes for ∂D . The mesh \mathcal{Q}_j on level j for ∂D is induced by dyadic subdivisions of depth j of the unit square into 4^j congruent squares, each of which is lifted to ∂D by the associated parameterization γ_i , see Figure IV.1 for a visualization. This procedure leads to a nested and especially 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 may use the hierarchy imposed by the nestedness of the meshes as an alternative to Algorithm II.2 to construct a perfectly balanced cluster tree as follows.

The particular elements shall be referred to as $\Gamma_{i,j,k}$, where i is the index of the underlying parameterization γ_i , j denotes the level of the element, and k is the index of the element in hierarchical order as illustrated in Figure IV.3. For notational convenience the triple (i, j, k) shall be referenced by $\lambda := (i, j, k)$ with $|\lambda| := j$. Obviously, for $j < J$, the element $\Gamma_{i,j,k}$ consists of the four elements $\Gamma_{i,j+1,4k+\ell}$, $\ell = 0, \dots, 3$. We can thus define a son relation on the corresponding index set by

$$\text{sons}(\lambda) := \{(i, j+1, 4k+\ell) : \ell = 0, \dots, 3\}.$$

It directly follows that the sons mapping induces a cluster tree in the sense of Definition II.5.3 and we remark that, in accordance with Definition II.5.5, it holds $\Upsilon_\lambda = \Gamma_\lambda$. An illustration for the cluster tree on the patch Γ_i up to level 2 can be found in in Figure IV.3.

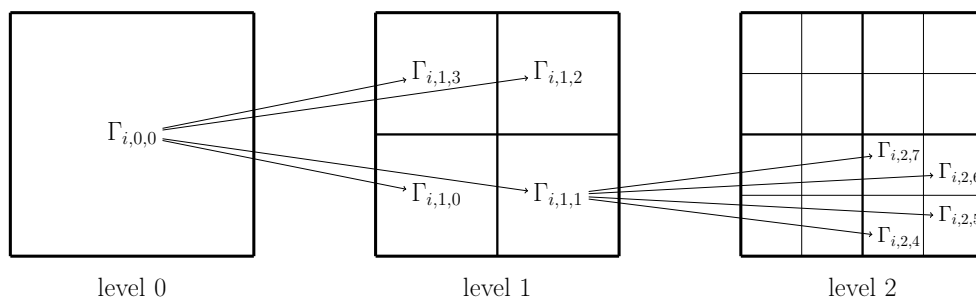


Figure IV.3: Visualization of the cluster tree.

Obviously, the depth of the constructed cluster tree is $J + 1$ on each patch, i.e., it is $\mathcal{O}(\log N_J)$.

Remark IV.2.1. *The proposed method of mesh generation particularly implies that all surface patches are represented by the same number of elements. In fact, this assumption is very similar to the “propagation of refinement” in isogeometric analysis, cf. [CHB09, HCB05]. An approach to cope with locally refined meshes in the isogeometric context was made in [CHR07].*

IV.3 Problem Formulation

For the rest of this chapter, we shall focus on the solution of boundary integral equations similar to the ones in Section IV.1. More generally, we consider equations of the type

$$(\mathcal{A}\rho)(\mathbf{x}) := \int_{\partial D} k(\mathbf{x}, \mathbf{y})\rho(\mathbf{y}) d\sigma_{\mathbf{y}} = f(\mathbf{x}) \quad (\text{IV.3.1})$$

on the closed, parametric surface ∂D , where the boundary integral operator \mathcal{A} is supposed to map $H^q(\partial D)$ continuously and one-to-one onto $H^{-q}(\partial D)$.

Remark IV.3.1. *If \mathcal{A} is a pseudodifferential operator of order $2q$, it directly follows that $\mathcal{A}: H^q(\partial D) \rightarrow H^{-q}(\partial D)$ is continuous, see also [HW08, Theorem 8.1.2].*

Following the considerations in Chapter II.4, the corresponding system matrix is densely populated. Assuming the asymptotical smoothness or similar properties of the kernel function, several ideas for the efficient approximation of the discrete system of linear equations were developed in the last decades to efficiently approximate the system matrix. The most prominent examples of such methods are the fast multipole method [GR87, GR97], the panel clustering method [HN89], the wavelet Galerkin scheme [BCR91, DHS06], and the adaptive cross approximation [Beb00]. Except for the wavelet Galerkin scheme, all these methods can be cast into an \mathcal{H} -matrices structure. This compression might even be improved by means of \mathcal{H}^2 -matrices, cf. [Bör10]. Then, the underlying tree structure of the \mathcal{H} -matrix is exploited to construct nested cluster bases for the compressible matrix blocks. In the end, all these discretization methods turn out to be of linear or almost linear complexity, i.e., linear up to a poly-logarithmic factor, with respect to the number of boundary elements.

We will provide a simple black-box version of the fast multipole method (FMM) for higher order boundary elements in order to make use of the features of parametric surfaces. In particular, the presented fast multipole method interpolates the kernel function directly on the reference domain. This is in contrast to the interpolation of the kernel in space, as in, e.g., [Gie01, HB02], and yields a remarkable speed-up of the FMM, since the dimension reduction due to the boundary integral formulation of the underlying problem can be fully exploited. In three spatial dimensions, the surface is a two-dimensional manifold and so the problem is inherently two-dimensional. This results in a dramatic reduction of the computational effort. Moreover, one can still profit from the \mathcal{H}^2 -matrix techniques presented in, e.g., [Gie01, HB02]. Notably, since the considered realization of parametric surfaces is based on four-sided patches, one can exploit the tensor product structure of the reference domain to simplify the construction of \mathcal{H}^2 -matrices. More precisely, due to the special structure of the reference domain, the construction of \mathcal{H}^2 -matrices only slightly differs from that of usual \mathcal{H} -matrices.

The variational formulation of the boundary integral equation (IV.3.1) now reads as follows:

$$\text{Find } \rho \in H^q(\partial D) \text{ such that } (\mathcal{A}\rho, v)_{L^2(\partial D)} = (f, v)_{L^2(\partial D)} \text{ for all } v \in H^q(\partial D). \quad (\text{IV.3.2})$$

Inserting the parametric representation (IV.2.1) of ∂D , the bilinear form becomes

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

and the linear form becomes

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

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

$$\left. \begin{aligned} k_{i,i'} : \square \times \square &\rightarrow \mathbb{R}, \\ k_{i,i'}(\mathbf{s}, \mathbf{t}) &:= k(\gamma_i(\mathbf{s}), \gamma_{i'}(\mathbf{t})) \kappa_i(\mathbf{s}) \kappa_{i'}(\mathbf{t}) \end{aligned} \right\} \quad i, i' = 1, 2, \dots, M. \quad (\text{IV.3.3})$$

We have thus pulled back the linear and bilinear form to the reference domain. In the following, we shall assume that the underlying kernel function is *analytically standard*, which also allows us to pull back the notion of asymptotic smoothness.

Definition IV.3.2. A kernel function $k(\mathbf{x}, \mathbf{y})$ is called *analytically standard of order $2q$* if constants $c_1 > 0$ and $c_2 > 0$ exist such that the partial derivatives of the transported kernel functions $k_{i,i'}(\mathbf{s}, \mathbf{t})$ are uniformly bounded by

$$|\partial_{\mathbf{s}}^{\boldsymbol{\alpha}} \partial_{\mathbf{t}}^{\boldsymbol{\beta}} k_{i,i'}(\mathbf{s}, \mathbf{t})| \leq c_1 \frac{(|\boldsymbol{\alpha}| + |\boldsymbol{\beta}|)!}{c_2^{|\boldsymbol{\alpha}| + |\boldsymbol{\beta}|}} \|\gamma_i(\mathbf{s}) - \gamma_{i'}(\mathbf{t})\|^{-2-2q-|\boldsymbol{\alpha}|-|\boldsymbol{\beta}|}, \quad (\text{IV.3.4})$$

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

Note that, since the parametric representation is patch-wise smooth, all asymptotically smooth kernels are also analytically standard, see, e.g., [HP13] for a proof of this statement. However, the converse is not true. For example, the kernels of the double layer operator and the adjoint double layer operator are only asymptotically smooth in one variable in space and, on non-smooth domains, also only asymptotically smooth in one variable in local coordinates. Nevertheless, both functions are analytically standard, since the normal vector is a smooth function on each patch.

In the context of the Galerkin approximation, it is convenient to refer also to the *localized kernel functions*. To that end, 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 dilation and translation. Then, the localized kernel functions $k_{\lambda,\lambda'} : \square \times \square \rightarrow \mathbb{R}$ are given by

$$k_{\lambda,\lambda'}(\mathbf{s}, \mathbf{t}) := k(\gamma_{\lambda}(\mathbf{s}), \gamma_{\lambda'}(\mathbf{t})) \kappa_{\lambda}(\mathbf{s}) \kappa_{\lambda'}(\mathbf{t}) \quad (\text{IV.3.5})$$

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 as defined in (IV.2.2). An illustration of the mappings γ_{λ} is given by Figure IV.4.

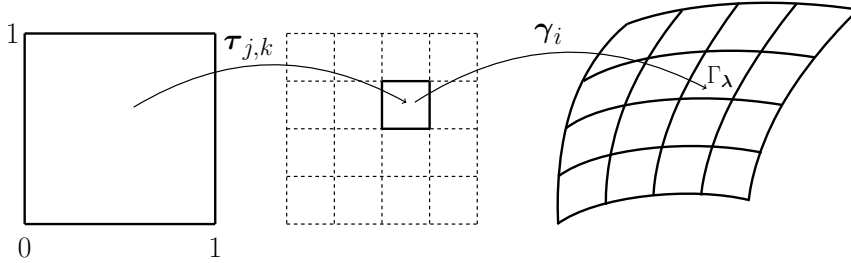


Figure IV.4: Localized parameterization.

In the following, only localized kernel functions shall be considered. The next theorem is an immediate consequence of the definition (IV.3.5) and the fact that $\partial_{\mathbf{s}}^{\boldsymbol{\alpha}} \tau_{j,k}(\mathbf{s}) = 2^{-j}$ if $|\boldsymbol{\alpha}| = 1$ and $\partial_{\mathbf{s}}^{\boldsymbol{\alpha}} \tau_{j,k}(\mathbf{s}) = 0$ if $|\boldsymbol{\alpha}| > 1$.

Theorem IV.3.3. *Let the kernel function $k(\mathbf{x}, \mathbf{y})$ be analytically standard of order $2q$. Then, there exist constants $c_1 > 0$ and $c_2 > 0$ such that*

$$|\partial_{\mathbf{s}}^{\alpha} \partial_{\mathbf{t}}^{\beta} k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t})| \leq c_1 \frac{(|\alpha| + |\beta|)!}{c_2^{|\alpha| + |\beta|}} \frac{2^{-|\lambda|(|\alpha| + 2)} 2^{-|\lambda'|(|\beta| + 2)}}{\|\gamma_{\lambda}(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|^{2 + 2q + |\alpha| + |\beta|}} \quad (\text{IV.3.6})$$

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

IV.4 Galerkin Discretization

In this section, the Galerkin discretization of the variational formulation (IV.3.2) is considered. To this end, fix a polynomial order $d \in \mathbb{N}$, a level of refinement $j \in \mathbb{N}_0$, and define the ansatz space

$$\hat{V}_j := \{\hat{\varphi}: \square \rightarrow \mathbb{R}: \hat{\varphi}|_{\square_{j,k}} \text{ is a polynomial of order } d\} \subset L^2(\square) \quad (\text{IV.4.1})$$

of discontinuous, element-wise polynomial ansatz functions on the reference domain. With the help of this space, one can introduce the ansatz space V_j in accordance with

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

This construction of the ansatz spaces obviously yields a nested sequence

$$V_0 \subset V_1 \subset \dots \subset V_J \subset H^t(\partial D), \quad (\text{IV.4.2})$$

where the Sobolev smoothness t depends on the global smoothness of the functions $\varphi \in V_j$. For arbitrary functions $\varphi \in V_j$, one has $t < 1/2$, and, for the subset of globally continuous functions in V_j , one has $t < 3/2$.

By replacing the energy space $H^q(\partial D)$ in the variational formulation (IV.3.2) by the finite dimensional ansatz space $V_J \subset H^q(\partial D)$, the Galerkin discretization for the boundary integral equation (IV.3.1) is given by:

Find $\rho_J \in V_J$, such that

$$\int_{\partial D} \int_{\partial D} k(\mathbf{x}, \mathbf{y}) \rho_J(\mathbf{y}) v_J(\mathbf{x}) d\sigma_{\mathbf{y}} d\sigma_{\mathbf{x}} = \int_{\partial D} f(\mathbf{x}) v_J(\mathbf{x}) d\sigma_{\mathbf{x}} \quad (\text{IV.4.3})$$

for all $v_J \in V_J$.

Setting $\hat{\rho}_{\lambda} := \rho_J \circ \gamma_{\lambda}$ and $\hat{v}_{\lambda} := v_J \circ \gamma_{\lambda}$, rewriting (IV.4.3) yields

$$\sum_{|\lambda'|=J} \int_{\square} \int_{\square} k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t}) \hat{\rho}_{\lambda'}(\mathbf{t}) \hat{v}_{\lambda}(\mathbf{s}) d\mathbf{t} d\mathbf{s} = \int_{\square} f(\gamma_{\lambda}(\mathbf{s})) \hat{v}_{\lambda}(\mathbf{s}) \kappa_{\lambda}(\mathbf{s}) d\mathbf{s} \quad (\text{IV.4.4})$$

for all λ with $|\lambda| = J$.

A basis for V_J is obtained by tensorizing polynomial shape functions on $[0, 1]$ and applying the localized parameterizations γ_{λ} . For $d = 1, 2, 3$, suitable shape functions are depicted in Table IV.1.

By choosing such a basis, (IV.4.3) immediately yields a system of linear equations:

$$\mathbf{A}_J \boldsymbol{\rho}_J = \mathbf{f}_J. \quad (\text{IV.4.5})$$

To realize globally continuous B-splines as ansatz functions, enabling for example the discretization of the hypersingular integral operator, suitable transformation matrices shall be applied. The construction of these transformation matrices is the topic of Section IV.6.

Having the Galerkin solution $\rho_J \in V_J$ at hand yields the following well known error estimate by use of the standard approximation property for ansatz functions of polynomial order d . Note that the rate of convergence doubles due to the Aubin-Nitsche lemma.

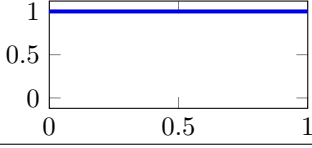
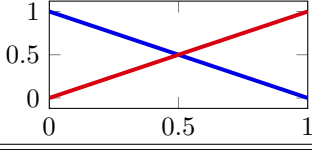
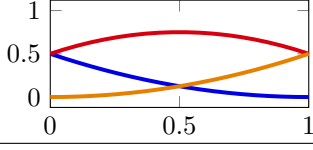
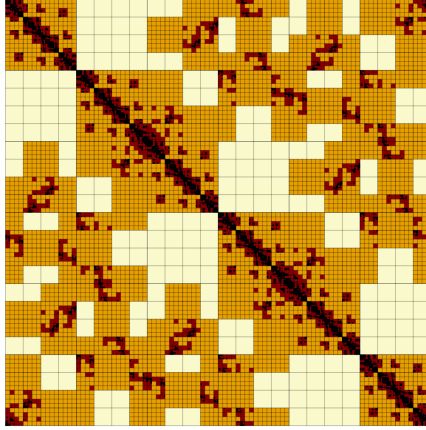
d	Shape Functions	Visualization
1	$\phi^{(i)}(x) = 1$	
2	$\phi^{(i)}(x) = \begin{cases} 1-x \\ x \end{cases}$	
3	$\phi^{(i)}(x) = \begin{cases} (1-x)^2/2 \\ -(x-1/2)^2 + 3/4 \\ x^2/2 \end{cases}$	

Table IV.1: B-spline based shape functions on the interval.


 Figure IV.5: The special block partitioning of the \mathcal{H} -matrix.

Theorem IV.4.1. *Let $\rho \in H^q(\partial D)$ be the solution of the boundary integral equation (IV.3.1) and $\rho_J \in V_J$ the related Galerkin solution of (IV.4.3). Then, there holds the error estimate*

$$\|\rho - \rho_J\|_{H^{2q-d}(\partial D)} \lesssim 2^{2J(q-d)} \|\rho\|_{H^q(\partial D)},$$

provided that ρ and ∂D are sufficiently regular.

IV.5 Fast Multipole Method

Having the cluster tree from the parametric surface representation available, we may use Algorithm II.3 to construct a block-cluster tree which partitions the matrix into nearfield and farfield blocks. Since all clusters on the same level of the cluster tree have the same cardinality, all block-clusters and their corresponding matrix blocks are quadratic, i.e., for all block-clusters $\lambda \times \lambda'$ it holds $\#\lambda = \#\lambda'$. In particular, for a block-cluster on level j , the cardinality of the corresponding clusters λ and λ' is a power of four. See also Figure IV.5 for a visualization of this special block partitioning of an \mathcal{H} -matrix.

The rest of this section shall discuss how this additional structure can be combined with

the tensor product structure of the reference domain to obtain an efficient compression of the farfield.

IV.5.1 Kernel Interpolation

To compress the admissible matrix blocks, we describe a black-box version of the FMM based on the interpolation of the kernel $k(\mathbf{x}, \mathbf{y})$ as firstly proposed in [Gie01]. Note that, later on, this idea was also followed in [HB02] to construct \mathcal{H}^2 -matrices. Nevertheless, in contrast to these works, the approach presented here interpolates the localized kernel (IV.3.5) on the reference domain rather than the original kernel in space.

For a given polynomial degree $p \in \mathbb{N}_0$, let $\{x_0, x_1, \dots, x_p\} \subset [0, 1]$ denote $p + 1$ interpolation points. Furthermore, let $L_m(s)$ for $m = 0, \dots, p$ be the Lagrangian basis polynomials with respect to these interpolation points. By a tensor product construction, one obtains the interpolation points $\mathbf{x}_m := (x_{m_1}, x_{m_2})$ and the corresponding tensor product basis polynomials $\mathbf{L}_m(\mathbf{s}) := L_{m_1}(s_1) \cdot L_{m_2}(s_2)$ for $m_1, m_2 = 0, \dots, p$. In all admissible blocks $\boldsymbol{\lambda} \times \boldsymbol{\lambda}' \in \mathcal{F}$, this gives rise to the approximation

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

Hence, for two basis functions $\hat{\phi}_\ell, \hat{\phi}_{\ell'} \in \hat{V}_{J-|\boldsymbol{\lambda}|}$ of the ansatz space on level $J - |\boldsymbol{\lambda}|$, the corresponding matrix entry has the representation

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

which can be visualized as

$$\boxed{\mathbf{A}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}'}} \approx \boxed{\mathbf{M}_{|\boldsymbol{\lambda}|}^{\square}} \boxed{\mathbf{K}_{\boldsymbol{\lambda}, \boldsymbol{\lambda}'}} \boxed{(\mathbf{M}_{|\boldsymbol{\lambda}'|}^{\square})^\top}$$

As remarked at the beginning of this section, by construction, each cluster on a particular level contains the same number of basis functions, namely $\dim(\hat{V}_{J-|\boldsymbol{\lambda}|})$. Additionally, the *moment matrices* $\mathbf{M}_{|\boldsymbol{\lambda}|}^{\square}$ are independent of the particular parameterization. This yields the following statement.

Theorem IV.5.1. *For $j = 1, 2, \dots, J$ and all $|\boldsymbol{\lambda}| = |\boldsymbol{\lambda}'| = j$, it holds*

$$\mathbf{M}_{|\boldsymbol{\lambda}|}^{\square} = \mathbf{M}_{|\boldsymbol{\lambda}'|}^{\square}. \quad (\text{IV.5.1})$$

Thus, only a single moment matrix

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

for each particular level has to be computed and stored. Because of quadrangular meshes, one may exploit the tensor product structure of the ansatz functions. To that end, let $\hat{\phi}_\ell =$

$\hat{\phi}_\ell^{(1)} \otimes \hat{\phi}_\ell^{(2)}$ and $\hat{\phi}_{\ell'} = \hat{\phi}_{\ell'}^{(1)} \otimes \hat{\phi}_{\ell'}^{(2)}$, respectively. Then, the moment matrices $\mathbf{M}_{|\lambda|}^\square$ can be decomposed even further:

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

Since

$$\mathbf{M}_{|\lambda|} \in \mathbb{R}^{\sqrt{\dim(\hat{V}_{J-|\lambda|}) \times (p+1)}}, \quad (\text{IV.5.2})$$

this yields an improved storage complexity for the farfield.

IV.5.2 Computational Complexity

In the sequel, complexity estimates for the FMM under consideration shall be derived. Similar to the constant n_{\min} in the Chapters II.5 and II.6, we may impose a lower threshold for the number of elements in the leaves. Setting $n_{\min} = p^2$ and taking the perfectly balanced structure of the cluster tree into account, we may only consider the block-cluster tree up to a level $J - j_{\min}$, with, assuming $p \geq d$,

$$j_{\min} := \lfloor 2 \log_4(p/d) \rfloor.$$

Thus, matrix blocks of size $p^2 \times p^2$ are considered as nearfield and will not be compressed by the FMM. The proof of the next theorem implies that this results in $\mathcal{O}(N_J(p/d)^{-2})$ nearfield blocks with a storage cost of $\mathcal{O}(N_J(pd)^2)$, where N_J is the number of elements on level J . Moreover, the following result holds for the cost complexity of the farfield.

Theorem IV.5.2. *The complexity for the computation and the storage of the farfield is $\mathcal{O}(N_J(pd)^2)$.*

Proof. We first remark, that for each cluster, there exist at most $C_{\max\eta}$ and at least $C_{\min\eta}$ neighbouring clusters on the same level which do not satisfy the admissibility condition (II.5.3). Due to the special structure of the mesh, these constants depend only on η and the position of the patches to each other.

We now look at an arbitrary cluster on level j . Since this cluster has at most $C_{\max\eta}$ non-admissible cluster-cluster interactions and there are $M4^j$ of these clusters, we have $M4^j C_{\max\eta}$ non-admissible cluster-cluster interactions on level j . To estimate the number of admissible cluster-cluster interaction for a cluster on level j , we remark that its father cluster has $C_{\max\eta}$ non-admissible cluster-cluster interactions. We have thus $4C_{\max\eta}$ candidates for admissible cluster-cluster interactions for each cluster on level j . Since at least $C_{\min\eta}$ of these interactions are non-admissible, we have for each cluster on level j at most $4C_{\max\eta} - C_{\min\eta}$ admissible cluster-cluster interactions and thus in total $M4^j(4C_{\max\eta} - C_{\min\eta})$ admissible cluster-cluster interactions on level j .

Due to the lower threshold, the maximum level to be computed is $J - j_{\min}$. Since $N_J = 4^J M$, one may estimate

$$\sum_{j=0}^{J-j_{\min}} M4^j(4C_{\max\eta} - C_{\min\eta}) = \mathcal{O}(M4^{J-j_{\min}}) = \mathcal{O}(M4^J(p/d)^{-2}) = \mathcal{O}(N_J(p/d)^{-2}),$$

leading to $\mathcal{O}(N_J(p/d)^{-2})$ farfield blocks and accordingly $\mathcal{O}(N_J(p/d)^{-2})$ nearfield blocks.

For each farfield block, the localized kernel functions have to be evaluated and stored in $\mathcal{O}(p^4)$ points. The complexity for assembly and storage of the moment matrices is $\mathcal{O}(\sqrt{N_J}pd)$ in total, cf. (IV.5.2). Consequently, the farfield complexity is

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

□

Remark IV.5.3. *Due to the parametric surface representation, an improved cost complexity is obtained. The standard interpolation-based FMM proposes to interpolate the kernel in space. Thus, the polynomial degree enters with $\mathcal{O}(p^3)$, cf. [Gie01, HB02]. Since the transported kernel is only interpolated on the reference domain, this cost can be reduced to $\mathcal{O}(p^2)$.*

The improved storage complexity also affects the cost of the \mathcal{H} -matrix-vector multiplication. The complexity of the conventional \mathcal{H} -matrix-vector multiplication is, setting $k = p^3$, $\mathcal{O}(N_J \log N_J p^3 d^2)$, whereas we obtain the following result for the here presented one.

Theorem IV.5.4. *The complexity of the \mathcal{H} -matrix-vector multiplication for the FMM is $\mathcal{O}(N_J \log N_J (pd)^2)$.*

Proof. On level $j \leq J - j_{\min}$, there are $M4^j(4C_{\max\eta} - C_{\min\eta})$ farfield blocks with a block size of $d^2 4^{J-j}$. Since it holds $p^4 = p^2 p^2 \leq p^2 4^{j_{\min}}$, the complexity of the matrix-vector multiplication for the farfield is therefore

$$\begin{aligned} \sum_{j=0}^{J-j_{\min}} M4^j(4C_{\max\eta} - C_{\min\eta})(2 \cdot 4^{J-j}(pd)^2 + p^4) \\ \leq 3(4C_{\max\eta} - C_{\min\eta})N_J(J - j_{\min})(pd)^2 = \mathcal{O}(N_J \log N_J (pd)^2). \end{aligned}$$

Next, look at the nearfield blocks and recall that there are $\mathcal{O}(N_J(p/d)^{-2})$ blocks with $\mathcal{O}(p^4)$ entries. Thus, we have $\mathcal{O}(N_J \log N_J (pd)^2)$ as the overall complexity of the \mathcal{H} -matrix-vector multiplication. □

IV.5.3 Nested Cluster Bases

The cost complexity of the matrix-vector multiplication can be improved to $\mathcal{O}(N_J (pd)^2)$ by exploiting the fact that the explicit computation of the moment matrices $\mathbf{M}_{|\lambda|}$ for each particular level can be avoided by the concept of *nested cluster bases* which amounts to the \mathcal{H}^2 -matrix representation, cf. [Bör10, HB02].

Since the polynomial degree for each cluster is p , the Lagrangian polynomials of the father cluster can obviously be represented by those of the son clusters. Let

$$\{x_m^{(0)}\}_{m=0}^p = \left\{ \frac{x_m}{2} \right\}_{m=0}^p \quad \text{and} \quad \{x_m^{(1)}\}_{m=0}^p = \left\{ \frac{x_m + 1}{2} \right\}_{m=0}^p,$$

respectively, be the interpolation points in the son clusters. It holds $\{x_m^{(0)}\}_{m=0}^p \subset [0, 0.5]$ and $\{x_m^{(1)}\}_{m=0}^p \subset [0.5, 1]$. Denoting the corresponding Lagrangian polynomials by $L_m^{(0)}(x)$ and $L_m^{(1)}(x)$, respectively, one can now exactly represent the Lagrangian polynomials of the father cluster according to

$$L_m(x) = \sum_{i=0}^p L_m(x_i^{(0)}) L_i^{(0)}(x) \quad \text{for } x \in [0, 0.5]$$

and

$$L_m(x) = \sum_{i=0}^p L_m(x_i^{(1)}) L_i^{(1)}(x) \quad \text{for } x \in [0.5, 1],$$

see Figure IV.6 for an illustration.

This gives rise to the *transfer matrices*

$$\mathbf{C}^{(0)} := [L_i(x_j^{(0)})]_{i,j=0}^p \quad \text{and} \quad \mathbf{C}^{(1)} := [L_i(x_j^{(1)})]_{i,j=0}^p$$

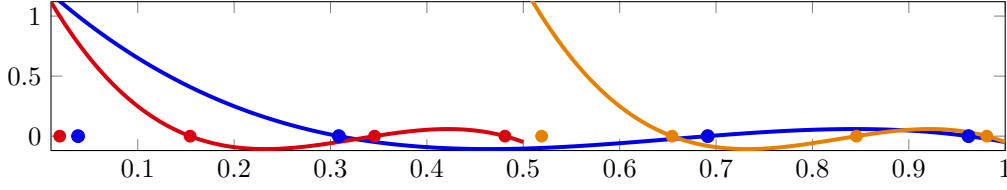


Figure IV.6: The Lagrange polynomial $L_0(x)$ in the father cluster and the Lagrange polynomials $L_0^{(0)}(x)$ and $L_0^{(1)}(x)$ in the son clusters for $p = 3$.

and yields the representation

$$\mathbf{M}_{|\lambda|} = \begin{bmatrix} \mathbf{M}_{|\lambda|+1}(\mathbf{C}^{(0)})^\top \\ \mathbf{M}_{|\lambda|+1}(\mathbf{C}^{(1)})^\top \end{bmatrix}.$$

Now, by tensor product construction, one obtains the four transfer matrices

$$\mathbf{C}_{2^i+2^j}^\square := \mathbf{C}^{(i)} \otimes \mathbf{C}^{(j)}, \quad i, j = 0, 1,$$

for the reference domain \square with the refinement relation

$$\mathbf{M}_{|\lambda|}^\square = \begin{bmatrix} \mathbf{M}_{|\lambda|+1}^\square(\mathbf{C}_0^\square)^\top \\ \mathbf{M}_{|\lambda|+1}^\square(\mathbf{C}_2^\square)^\top \\ \mathbf{M}_{|\lambda|+1}^\square(\mathbf{C}_3^\square)^\top \\ \mathbf{M}_{|\lambda|+1}^\square(\mathbf{C}_1^\square)^\top \end{bmatrix}. \quad (\text{IV.5.3})$$

Notice that the peculiar order of the transfer matrices results from the hierarchical, counter clock-wise ordering of the elements, see Figure IV.3. Fortunately, the transfer matrices $\mathbf{C}_0^\square, \mathbf{C}_1^\square, \mathbf{C}_2^\square, \mathbf{C}_3^\square$ are independent of the respective cluster and even independent of the level $|\lambda|$. Moreover, the transfer matrices are independent of the ansatz functions chosen for the Galerkin discretization.

In order to make use of the efficient implementation of the \mathcal{H}^2 -matrix-vector multiplication as in [Bör10, HB02], only \mathbf{M}_J^\square and $\mathbf{C}_0^\square, \mathbf{C}_1^\square, \mathbf{C}_2^\square, \mathbf{C}_3^\square$ have to be stored. This leads, together with the hierarchical ordering of the elements, to some simplifications in the \mathcal{H}^2 -matrix-vector multiplication. The algorithm, tailored to the framework of parametric surfaces, is split in three parts: Algorithms IV.1, IV.2 and IV.3.

Theorem IV.5.5. *The \mathcal{H}^2 -matrix-vector multiplication of the FMM as stated in Algorithm IV.1 has a complexity of $\mathcal{O}(N_J(pd)^2)$.*

Proof. To estimate the complexity of Algorithm IV.2, note that applying $N_{J-j_{\min}}$ times the moment matrices with $\mathcal{O}(N_{j_{\min}}(pd)^2)$ entries takes at most $\mathcal{O}(N_J(pd)^2)$ operations. The application of the transfer matrices to level $j+1$ requires $4p^4$ operations for each of the N_j clusters on level j . Hence, in a similar way as in the proof of Theorem IV.5.2, one concludes that the overall complexity of Algorithm IV.2 is

$$\mathcal{O}(N_J(pd)^2) + 4p^4 \sum_{j=0}^{J-j_{\min}} N_j = \mathcal{O}(N_J(pd)^2) + 4p^4 \mathcal{O}(N_J(p/d)^{-2}) = \mathcal{O}(N_J(pd)^2).$$

In complete analogy, the complexity of Algorithm IV.3 is given by $\mathcal{O}(N_J(pd)^2)$. The complexity of the multiplication with the farfield coincides with the complexity of its memory consumption as it was derived in Theorem IV.5.2. The complexity for the nearfield is the same as in the classical \mathcal{H} -matrix-vector multiplication, which was estimated in Theorem IV.5.4. This yields a total complexity of $\mathcal{O}(N_J(pd)^2)$ for the \mathcal{H}^2 -matrix-vector multiplication. \square

Algorithm IV.1 \mathcal{H}^2 -matrix-vector multiplication, $\mathbf{y} += \mathbf{H}\mathbf{x}$, see [Bör10, Algorithm 8]

```

function  $\mathcal{H}^2$ -MATRIX-VECTOR( $\mathbf{y}$ ,  $\mathbf{H}$ ,  $\mathbf{x}$ )
   $\mathbf{u} = \text{FORWARDTRANSFORM}(\mathbf{x})$  ▷ Handle farfield
  for  $\lambda \times \lambda' \in \mathcal{F}$  do
     $\mathbf{v}_\lambda += \mathbf{K}_{\lambda \times \lambda'} \cdot \mathbf{u}_{\lambda'}$ 
  end for
   $\mathbf{y} += \text{BACKWARDTRANSFORM}(\mathbf{v})$ 
  for  $\lambda \times \lambda' \in \mathcal{N}$  do ▷ Handle nearfield
     $\mathbf{y}|_\lambda += \mathbf{H}|_{\lambda \times \lambda'} \cdot \mathbf{x}|_{\lambda'}$ 
  end for
end function

```

Algorithm IV.2 Forward transformation of \mathbf{x} to \mathbf{u} , see [Bör10, Algorithm 6]

```

function FORWARDTRANSFORM( $\mathbf{x}$ )
  for  $(i, j', k) \in \mathcal{T}$ ,  $j' = J$  do
     $\mathbf{u}_{(i, j', k)} = (\mathbf{M}_J^\square)^\top \mathbf{x}|_{(i, j', k)}$ 
  end for
  for  $j = J - 1, \dots, 1$  do
    for  $(i, j', k) \in \mathcal{T}$ ,  $j' = j$  do
       $\mathbf{u}_{(i, j', k)} = \mathbf{C}_0^\square \mathbf{u}_{(i, j+1, 4k)}$ 
       $\quad + \mathbf{C}_2^\square \mathbf{u}_{(i, j+1, 4k+1)}$ 
       $\quad + \mathbf{C}_3^\square \mathbf{u}_{(i, j+1, 4k+2)}$ 
       $\quad + \mathbf{C}_1^\square \mathbf{u}_{(i, j+1, 4k+3)}$ 
    end for
  end for
end function

```

Algorithm IV.3 Backward transformation of \mathbf{v} to \mathbf{y} , see [Bör10, Algorithm 7]

```

function BACKWARDTRANSFORM( $\mathbf{v}$ )
  for  $j = 1, \dots, J - 1$  do
    for  $(i, j', k) \in \mathcal{T}$ ,  $j' = j$  do
      
$$\begin{bmatrix} \mathbf{v}_{(i, j+1, 4k)} \\ \mathbf{v}_{(i, j+1, 4k+1)} \\ \mathbf{v}_{(i, j+1, 4k+2)} \\ \mathbf{v}_{(i, j+1, 4k+3)} \end{bmatrix} = \begin{bmatrix} (\mathbf{C}_0^\square)^\top \\ (\mathbf{C}_2^\square)^\top \\ (\mathbf{C}_3^\square)^\top \\ (\mathbf{C}_1^\square)^\top \end{bmatrix} \mathbf{v}_{(i, j', k)}$$

    end for
  end for
  for  $(i, j', k) \in \mathcal{T}$ ,  $j' = J$  do
     $\mathbf{y}|_{(i, j', k)} = \mathbf{M}_J^\square \mathbf{v}_{(i, j', k)}$ 
  end for
end function

```

IV.5.4 Error Estimates

In view of Definition IV.3.2, the following result for the convergence of the FMM holds. It refers to the situation, when 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, cf. [HB02, HP13].

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

$$\begin{aligned} & \left\| k_{\lambda, \lambda'}(\mathbf{s}, \mathbf{t}) - \sum_{\|\mathbf{m}\|_\infty, \|\mathbf{m}'\|_\infty \leq p} k_{\lambda, \lambda'}(\mathbf{x}_m, \mathbf{x}_{m'}) \mathbf{L}_m(\mathbf{s}) \mathbf{L}_{m'}(\mathbf{t}) \right\|_{L^\infty(\square \times \square)} \\ & \lesssim \left(\frac{\eta}{c_2} \right)^{p+1} 2^{-4|\lambda|} \|\gamma_\lambda(\mathbf{s}) - \gamma_{\lambda'}(\mathbf{t})\|_{L^\infty(\square \times \square)}^{-2-2q} \end{aligned}$$

with $c_2 > 0$ being the constant from Definition IV.3.2 and η being the constant from (II.5.3).

Remark IV.5.7. *The previous theorem indicates that the convergence behaviour of the FMM directly depends on the properties of the interpolated kernel function. Especially, one has to choose $\eta < c_2$ in (II.5.3) in order to obtain convergence. For example, for the fundamental solution of the Helmholtz equation, the constant c_2 can become quite small, which in turn increases the nearfield.*

From Theorem IV.5.6, one immediately derives an error estimate for the bilinear form which is associated with the variational formulation (IV.3.2), cf. [Gie01, HP13].

Theorem IV.5.8. *Let $\sigma > 0$ be arbitrary 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*

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

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

Hence, in order to maintain the approximation order of the Galerkin method, one has to choose $p \sim \log N_J$. This leads to an over-all complexity of $\mathcal{O}(N_J(\log N_J)^2 d^2)$ for the computation and the storage of the farfield. Nevertheless, in view of singular kernels, one has to deal with singular integrals, e.g., by the Duffy trick, cf. [Duf82]. Thus, one has to increase the degree of the quadrature for all singular integrals proportionally to $|\log h_J|$, where $h_J = 2^{-J}$ is the mesh size and the constant depends on the order of the ansatz functions. This yields an effort of $\mathcal{O}((\log N_J)^4)$ for each singular integral if a tensor product quadrature is used. Thus, if the quadrature degree is properly decreased with the distance of the elements, one ends up with a complexity of $\mathcal{O}(N_J(\log N_J)^4 d^2)$ for the nearfield. The reader is referred to [SS97] for more details concerning the numerical quadrature.

IV.6 Higher Order Continuous Ansatz Functions

One of the issues to address for continuous, higher order ansatz functions is the clustering strategy. In the classical \mathcal{H} - and \mathcal{H}^2 -matrix framework, usually a per degree of freedom cluster strategy is employed, see, e.g., [Bör10, Hac15]. In the context of higher order ansatz functions, this strategy was applied to collocation matrices in [ZMBF14] to compress the system matrices by using adaptive cross approximation. However, a per degree of freedom cluster strategy requires to iterate over the degrees of freedom during the matrix assembly. For the Galerkin scheme, this means for every degree of freedom that all elements in the support

of the associated ansatz function have to be taken into account. Thus, for continuous, higher order ansatz functions, every element is visited several times during the matrix assembly and function evaluations for the numerical quadrature are possibly done multiple times in the same quadrature point.

In order to overcome this obstruction, one therefore often iterates over the elements for the matrix assembly. To maintain this element-wise strategy for the matrix assembly of a higher order FMM, we propose to keep the element-wise cluster strategy introduced in Chapter IV.2. In the sequel, an easy procedure to extend the FMM for discontinuous, element-wise polynomial ansatz functions from Chapter IV.5 to globally continuous ansatz functions shall be provided. This means that, from now on, ansatz spaces $V_j^c = V_j \cap C(\partial D)$ are considered. Clearly, there exists then a transformation matrix \mathbf{T} such that

$$\mathbf{TA}_J \mathbf{T}^\top \boldsymbol{\rho}_J^c = \mathbf{Tf}_J, \quad (\text{IV.6.1})$$

where \mathbf{A}_J is the system matrix and \mathbf{f}_J is the right hand side with respect to the discontinuous, element-wise polynomial ansatz functions from (IV.4.5) and $\boldsymbol{\rho}_J^c$ are the coefficients of the globally continuous ansatz functions in V_j^c . The transformation matrix \mathbf{T} is a sparse matrix if the supports of the ansatz functions in V_j^c only contain a few elements, as it is, e.g., the case for B-splines, which are used in isogeometric analysis. This situation will be illustrated in the following.

Denote by \hat{V}_j^c the space spanned by the tensor product B-splines of order d on the reference domain. The tensor product B-splines are obtained by tensorization of the B-spline basis of order d on the interval $[0, 1]$. To that end, introduce the partition

$$0 = t_1 = \dots = t_d < \dots < t_{n+d+1} = \dots = t_{n+2d} = 1.$$

Now, setting

$$B_{j,1}(x) = \begin{cases} 1, & \text{if } t_j \leq x < t_{j+1}, \\ 0, & \text{otherwise,} \end{cases} \quad j = 1, \dots, n + 2d - 1,$$

and using the recursion formula

$$B_{j,k}(x) = \frac{x - t_j}{t_{j+k-1} - t_j} B_{j,k-1}(x) + \frac{t_{j+k} - x}{t_{j+k} - t_{j+1}} B_{j+1,k-1}(x), \quad j = 1, \dots, n + 2d - k,$$

up to $k = d$ will result in the $n + d$ B-spline basis functions of order d on $[0, 1]$, cf. [DB78, Chapter 9]. The B-spline bases up to order 3 are depicted in Figure IV.7.

It holds $\hat{V}_j^c \subset \hat{V}_j$, such that every function in \hat{V}_j^c can be expressed as a linear combination of functions in \hat{V}_j . Let $\hat{\mathbf{U}} = [\hat{\varphi}_1^c, \dots, \hat{\varphi}_{\hat{n}_j^c}^c]$ denote the tensor product B-spline basis of \hat{V}_j^c and let $\hat{\mathbf{V}} = [\hat{\varphi}_1, \dots, \hat{\varphi}_{\hat{n}_j}]$ denote the piecewise polynomial basis of \hat{V}_j , where $\hat{n}_j^c = \dim(\hat{V}_j^c)$ and $\hat{n}_j = \dim(\hat{V}_j)$. Then the patchwise transformation matrix $\hat{\mathbf{T}}$ is uniquely determined by the relation $\hat{\mathbf{U}} = \hat{\mathbf{T}}\hat{\mathbf{V}}$. Unfortunately, the functions in the composed space

$$\tilde{V}_j = \{\hat{\varphi} \circ \boldsymbol{\gamma}_i^{-1} : \hat{\varphi} \in \hat{V}_j^c, i = 1, \dots, M\} \subset L^2(\partial D)$$

are, in general, discontinuous on the boundaries of the patches $\partial\Gamma_i$. The ansatz space V_j^c of globally continuous, tensorized B-splines on level j is given by

$$V_j^c := \tilde{V}_j \cap C(\partial D).$$

Since $V_j^c \subset \tilde{V}_j$, every function in V_j^c can be expressed as a linear combination of functions in \tilde{V}_j with corresponding transformation matrix \mathbf{I} . To express functions in V_j^c using functions in V_j , let $[\varphi_1^c, \dots, \varphi_{n_j^c}^c]$ with $n_j^c = \dim(V_j^c)$ denote a basis of V_j^c and

$$a = \sum_{i=1}^{n_j^c} a_i \varphi_i^c \in V_j^c, \quad b = \sum_{i=1}^M \sum_{k=1}^{\hat{n}_j} b_{i,k} (\hat{\varphi}_k \circ \boldsymbol{\gamma}_i^{-1}) \in V_j.$$

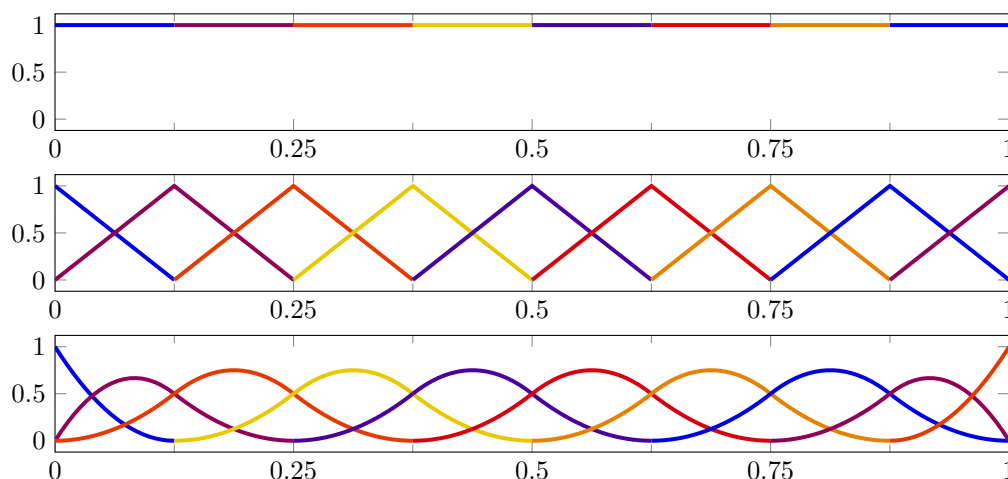


Figure IV.7: B-spline bases on the unit interval $[0, 1]$ of order 1 (top), order 2 (middle), and order 3 (bottom).

Setting $\mathbf{b}_i = [b_{i,1}, \dots, b_{i,\hat{n}_j}]$ finally yields

$$\begin{bmatrix} a_1 \\ \vdots \\ a_{n_j^c} \end{bmatrix} = \mathbf{I} \begin{bmatrix} \hat{\mathbf{T}} & & \\ & \ddots & \\ & & \hat{\mathbf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{bmatrix}.$$

The matrices $\hat{\mathbf{T}}$ and \mathbf{I} are sparse and so is their product \mathbf{T} . Thus, the transformation from V_j to V_j^c can be done in an efficient manner.

At first glance, the simplicity of the presented method comes at a high price. The memory consumption of the uncompressed matrix \mathbf{A}_J in (IV.6.1) is n_J^2 , where $n_J = \dim(V_J)$, instead of $(n_J^c)^2$, i.e., the memory consumption will grow by a factor $(n_J/n_J^c)^2$, whereas the number of degrees of freedom only grows by a factor n_J/n_J^c . Although nowadays memory consumption can be considered as a minor problem, this also means that in case of uncompressed matrices the computational effort for the matrix-vector multiplication will grow like $\mathcal{O}((n_J/n_J^c)^2)$. Compared to this, the FMM compression presented in the previous section reduces the growth of the memory consumption and the operations for the \mathcal{H}^2 -matrix-vector multiplication to $\mathcal{O}(n_J/n_J^c)$.

IV.7 Numerical Examples

Besides presenting numerical examples for the convergence of the fast multipole method, this section contains also a comparison of the computational cost versus accuracy. All computations of the following examples were carried out on a single core of a computing server with two Intel(R) Xeon(R) E5-2670 CPUs with a clock rate of 2.60 GHz and a main memory of 256 GB.

The experiments focus on the numerical solution of the boundary integral equations (IV.1.8), (IV.1.9), (IV.1.11), and (IV.1.12) arising from the potential ansatzes for the Laplace equation with either Dirichlet boundary conditions (IV.1.5) or Neumann boundary conditions (IV.1.6). The special numerical treatment of the hypersingular operator (IV.1.12) is the topic of the next subsection. Note that no special treatment is needed for the double layer operator and the adjoint double layer operator, since they are both analytically standard.

The ansatz functions shall be B-splines of order $d = 1, 2, 3$, which are glued together at the patch interfaces to achieve global continuity for $d = 2, 3$. This means that the system of linear

equations is computed with the help of the transformation matrices which were introduced in Chapter IV.6. To solve the system (IV.6.1) of linear equations, a conjugate gradient method (CG) for (IV.1.8) and (IV.1.12) and the generalized minimal residual method (GMRES) with restart after 100 inner iterations for (IV.1.9) and (IV.1.11) are used, cf. [Saa03]. The construction of appropriate preconditioners for the iterative solvers exceeds the scope of this thesis and is left as further work.

IV.7.1 Regularization of the Hypersingular Operator

The *hypersingular operator* \mathcal{W} from (IV.1.12) is formally given by the integral operator

$$(\mathcal{W}\rho)(\mathbf{x}) = \frac{1}{4\pi} \int_{\partial D} \left(\frac{\langle \mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}} \rangle}{\|\mathbf{x} - \mathbf{y}\|^3} - 3 \frac{\langle \mathbf{n}_{\mathbf{x}}, \mathbf{x} - \mathbf{y} \rangle \langle \mathbf{n}_{\mathbf{y}}, \mathbf{x} - \mathbf{y} \rangle}{\|\mathbf{x} - \mathbf{y}\|^5} \right) \rho(\mathbf{y}) \, d\sigma_{\mathbf{y}}, \quad (\text{IV.7.1})$$

which has a strong singularity in its kernel function. Therefore, a regularization is needed in order to apply quadrature rules for its discretization. The following representation, derived firstly in [Néd82], reduces the singularity of the integral operator in the inner product to the same singularity as in (IV.1.8):

$$(\mathcal{W}\rho, \vartheta)_{L^2(\partial D)} = (\mathcal{V} \operatorname{curl}_{\partial D} \rho, \operatorname{curl}_{\partial D} \vartheta)_{L^2(\partial D)} \quad \text{for all } \rho, \vartheta \in H^1(\partial D). \quad (\text{IV.7.2})$$

It remains to find an easy means to compute the surface curl of a given function in $H^1(\partial D)$.

For each patch $\Gamma_i = \gamma_i(\square)$, consider the first fundamental tensor of differential geometry, cf. [CK13, Chapter 2.1],

$$\mathbf{K}_i(\mathbf{s}) = \begin{bmatrix} \langle \partial_{s_1} \gamma_i(\mathbf{s}), \partial_{s_1} \gamma_i(\mathbf{s}) \rangle & \langle \partial_{s_1} \gamma_i(\mathbf{s}), \partial_{s_2} \gamma_i(\mathbf{s}) \rangle \\ \langle \partial_{s_2} \gamma_i(\mathbf{s}), \partial_{s_1} \gamma_i(\mathbf{s}) \rangle & \langle \partial_{s_2} \gamma_i(\mathbf{s}), \partial_{s_2} \gamma_i(\mathbf{s}) \rangle \end{bmatrix}, \quad \mathbf{s} = [s_1, s_2]^\top \in \square.$$

Since γ_i is a diffeomorphism, \mathbf{K}_i is a symmetric and positive definite matrix and every function $\rho|_{\Gamma_i}$ has a unique representation $\tilde{\rho}_i = \rho \circ \gamma_i$ on \square . Thus, for $\mathbf{x} = \gamma_i(\mathbf{s}) \in \Gamma_i$, one can define the surface gradient of ρ by

$$\operatorname{grad}_{\partial D} \rho(\mathbf{x}) = \begin{bmatrix} \partial_{s_1} \gamma_i(\mathbf{s}) & \partial_{s_2} \gamma_i(\mathbf{s}) \end{bmatrix} \mathbf{K}_i^{-1}(\mathbf{s}) \begin{bmatrix} \partial_{s_1} \tilde{\rho}_i(\mathbf{s}) \\ \partial_{s_2} \tilde{\rho}_i(\mathbf{s}) \end{bmatrix}, \quad (\text{IV.7.3})$$

cf. [CK13, Chapter 2.1]. Note that this definition is independent of the actual chosen parametrization γ_i .

Having the definition of the surface gradient at hand, the surface curl is defined by

$$\operatorname{curl}_{\partial D} \rho(\mathbf{x}) = \mathbf{n}_{\mathbf{x}} \times \operatorname{grad}_{\partial D} \rho(\mathbf{x}), \quad \mathbf{x} \in \partial D. \quad (\text{IV.7.4})$$

Inserting (IV.7.3) into (IV.7.4) yields

$$\operatorname{curl}_{\partial D} \rho(\mathbf{x}) = \frac{1}{\sqrt{\det \mathbf{K}_i(\mathbf{s})}} (\partial_{s_1} \tilde{\rho}_i(\mathbf{s}) \partial_{s_2} \gamma_i(\mathbf{s}) - \partial_{s_2} \tilde{\rho}_i(\mathbf{s}) \partial_{s_1} \gamma_i(\mathbf{s})), \quad (\text{IV.7.5})$$

where $\sqrt{\det \mathbf{K}_i(\mathbf{s})} = \kappa_i(\mathbf{s})$ is the surface measure of Γ_i from (IV.2.2).

Finally, recall that an ansatz function φ_i on the surface Γ_i is given by $\varphi_i = \hat{\varphi} \circ \gamma_i^{-1}$, where $\hat{\varphi} \in \hat{V}_j$ is defined as in (IV.4.1). It therefore holds

$$\partial_{s_k} \tilde{\varphi}_i(\mathbf{s}) = \partial_{s_k} (\hat{\varphi} \circ \gamma_i^{-1} \circ \gamma_i)(\mathbf{s}) = \partial_{s_k} \hat{\varphi}(\mathbf{s}), \quad k = 1, 2.$$

Thus, in order to numerically compute (IV.7.5) for the ansatz and the test space, one only has to provide the derivatives of the tensorized local shape functions on the unit square from Table IV.1. Hence, the construction from Chapter IV.5.1, applied to (IV.7.2) with $\operatorname{curl}_{\partial D}$ computed as in (IV.7.5), results in a slightly modified approximation for admissible matrix blocks:

$$\mathbf{A}_{\lambda, \lambda'} \approx \begin{bmatrix} \mathbf{M}_{|\lambda|}^{1, \square} & \mathbf{M}_{|\lambda|}^{2, \square} \end{bmatrix} \begin{bmatrix} \mathbf{K}_{\lambda, \lambda'}^{(1,1)} & \mathbf{K}_{\lambda, \lambda'}^{(1,2)} \\ \mathbf{K}_{\lambda, \lambda'}^{(2,1)} & \mathbf{K}_{\lambda, \lambda'}^{(2,2)} \end{bmatrix} \begin{bmatrix} (\mathbf{M}_{|\lambda|}^{1, \square})^\top \\ (\mathbf{M}_{|\lambda|}^{2, \square})^\top \end{bmatrix}.$$

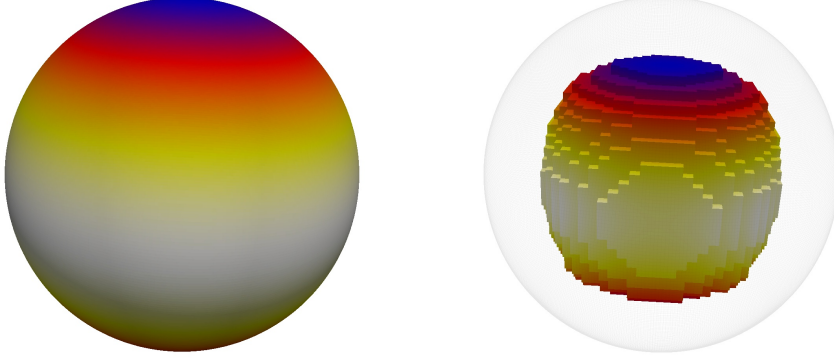


Figure IV.8: The spherical harmonic Y_0^2 (left) and the related potential (right) for the unit ball.

Herein, the modified moment matrices are given by

$$\begin{aligned} (\mathbf{M}_{|\lambda|}^\partial)_{m_1, \ell} &= \int_0^1 L_{m_1}(s_1) (\partial_{s_1} \hat{\phi}_\ell^{(1)})(s_1) ds_1, \\ \mathbf{M}_{|\lambda|}^{1, \square} &= \mathbf{M}_{|\lambda|}^\partial \otimes \mathbf{M}_{|\lambda|}, \quad \mathbf{M}_{|\lambda|}^{2, \square} = \mathbf{M}_{|\lambda|} \otimes \mathbf{M}_{|\lambda|}^\partial, \end{aligned}$$

and the modified kernel matrices by

$$\begin{aligned} [\mathbf{K}_{\lambda, \lambda'}^{(1,1)}]_{\mathbf{m}, \mathbf{m}'} &= k(\gamma_\lambda(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}(\mathbf{x}_{\mathbf{m}'})) \langle \gamma_\lambda^{\partial_2}(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}^{\partial_2}(\mathbf{x}_{\mathbf{m}'}), \rangle, \\ [\mathbf{K}_{\lambda, \lambda'}^{(1,2)}]_{\mathbf{m}, \mathbf{m}'} &= -k(\gamma_\lambda(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}(\mathbf{x}_{\mathbf{m}'})) \langle \gamma_\lambda^{\partial_2}(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}^{\partial_1}(\mathbf{x}_{\mathbf{m}'}), \rangle, \\ [\mathbf{K}_{\lambda, \lambda'}^{(2,1)}]_{\mathbf{m}, \mathbf{m}'} &= -k(\gamma_\lambda(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}(\mathbf{x}_{\mathbf{m}'})) \langle \gamma_\lambda^{\partial_1}(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}^{\partial_2}(\mathbf{x}_{\mathbf{m}'}), \rangle, \\ [\mathbf{K}_{\lambda, \lambda'}^{(2,2)}]_{\mathbf{m}, \mathbf{m}'} &= k(\gamma_\lambda(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}(\mathbf{x}_{\mathbf{m}'})) \langle \gamma_\lambda^{\partial_1}(\mathbf{x}_\mathbf{m}), \gamma_{\lambda'}^{\partial_1}(\mathbf{x}_{\mathbf{m}'}), \rangle, \end{aligned}$$

where the *localized parametrization derivatives* are defined by $\gamma_\lambda^{\partial_\ell} = \partial_{s_\ell} \gamma_i \circ \tau_{j,k}$, $\ell = 1, 2$.

Note that the very same refinement relation (IV.5.3) holds between $\mathbf{M}_{|\lambda|}^{i, \square}$ and its predecessors $\mathbf{M}_{|\lambda|+1}^{i, \square}$, $i = 1, 2$. Thus, by a small but straightforward modification, the \mathcal{H}^2 -matrix-vector multiplication can also be applied in the case of the regularized hypersingular operator.

IV.7.2 Convergence

The first example shall be concerned with the boundary value problems (IV.1.5) and (IV.1.6) in the unit ball with a prescribed analytical solution based on the spherical harmonic Y_0^2 ,

$$u(\mathbf{x}) = \|\mathbf{x}\| Y_0^2\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right), \quad \mathbf{x} \in D, \quad Y_0^2(\mathbf{x}) = \sqrt{\frac{5}{16\pi}} (3x_3^2 - 1), \quad \mathbf{x} \in \partial D,$$

see Figure IV.8 for a visualization. The Dirichlet and Neumann data for (IV.1.5) and (IV.1.6) are then given by

$$f(\mathbf{x}) = Y_0^2(\mathbf{x}), \quad \mathbf{x} \in \partial D, \quad g(\mathbf{x}) = (\nabla u(\mathbf{x}), \mathbf{x}), \quad \mathbf{x} \in \partial D.$$

The boundary of the sphere shall be represented by six patches.

In view of Theorem IV.4.1, one obtains the following error estimate for the approximate potential u_J .

Theorem IV.7.1. *Let $\rho \in H^d(\partial D)$ be the solution of (IV.1.8), (IV.1.9), (IV.1.11), or (IV.1.12). Moreover, let u be the corresponding potential and u_J its numerical approximation. Then, there holds the error estimate*

$$|u(\mathbf{x}) - u_J(\mathbf{x})| \lesssim 2^{2J(q-d)} \|k(\mathbf{x}, \cdot)\|_{H^{-2q+d}(\partial D)} \|\rho\|_{H^d(\partial D)}, \quad \mathbf{x} \in D,$$

where $2q = -1$ in case of (IV.1.8), $2q = 0$ in case of (IV.1.9) and (IV.1.11), and $2q = 1$ in case of (IV.1.12).

Proof. Together with Theorem IV.4.1, there holds

$$\begin{aligned} |u(\mathbf{x}) - u_J(\mathbf{x})| &= \left| \int_{\partial D} k(\mathbf{x}, \mathbf{y}) (\rho(\mathbf{y}) - \rho_J(\mathbf{y})) \, d\sigma_{\mathbf{y}} \right| \\ &\leq \|k(\mathbf{x}, \cdot)\|_{H^{-2q+d}(\partial D)} \|\rho - \rho_J\|_{H^{2q-d}(\partial D)} \\ &\lesssim 2^{2J(q-d)} \|k(\mathbf{x}, \cdot)\|_{H^{2q-d}(\partial D)} \|\rho\|_{H^d(\partial D)} \end{aligned}$$

for all four cases under consideration. □

Furthermore, on the sphere, the spherical harmonic is an eigenfunction to all integral operators under consideration, namely, there holds

$$\begin{aligned} \mathcal{V}Y_0^2 &= \frac{1}{5}Y_0^2, & \left(\frac{1}{2} - \mathcal{K}\right)Y_0^2 &= -\frac{3}{5}Y_0^2, \\ \left(\frac{1}{2} + \mathcal{K}^*\right)Y_0^2 &= \frac{2}{5}Y_0^2, & \mathcal{W}Y_0^2 &= \frac{6}{5}Y_0^2. \end{aligned}$$

Hence, the analytical solution of (IV.1.8), (IV.1.9), (IV.1.11), and (IV.1.12) is known and one can thus compute the $L^2(\partial D)$ -error of the approximate density. The following theorem can be seen as a combination of Theorem IV.4.1 and a small modification of [Ste08, Lemma 12.2].

Theorem IV.7.2. *Let $\rho \in H^d(\partial D)$ be the solution of (IV.1.8), (IV.1.9), (IV.1.11), or (IV.1.12) and ρ_J its numerical approximation. Then, there holds the error estimate*

$$\|\rho - \rho_J\|_{L^2(\partial D)} \lesssim 2^{-Jd} \|\rho\|_{H^d(\partial D)}.$$

Figure IV.9 and IV.10 validate that the proposed FMM provides the theoretical convergence rates on smooth domains in case of the Dirichlet problem. Figure IV.11 and IV.12 validate this for the Neumann problem. Note that the hypersingular operator in Figure IV.12 requires globally continuous ansatz functions, thus the order of the ansatz functions for this particular example has to be at least $d = 2$. In both cases, the ℓ^∞ -error of the potential is measured in the 18,999 vertices of 16,616 cubes which lie in the interior of the ball, as depicted on the right of Figure IV.8. The polynomial degree p for the FMM is chosen such that the overall accuracy is sustained. Hence, in accordance with Theorem IV.5.8, the number of interpolation points on the interval grows linearly with the discretization level J . The numbers of local and global degrees of freedom n_J and n_J^c , respectively, associated with the discretization level J , are tabulated in Table IV.2. Note that both numbers coincide in case of piecewise constant boundary elements, i.e., for $d = 1$.

IV.7.3 Convergence on a More Complex Geometry

The second example shall deal with the more complex gear worm geometry depicted in Figure IV.13, which is represented by 290 patches. The harmonic polynomial

$$u(\mathbf{x}) = 4x_1^2 - 3x_2^2 - x_3^2$$

is prescribed as potential and used to obtain the Dirichlet and Neumann data for (IV.1.5) and (IV.1.6). For sake of brevity, we restrict ourselves to one example for Dirichlet problems,

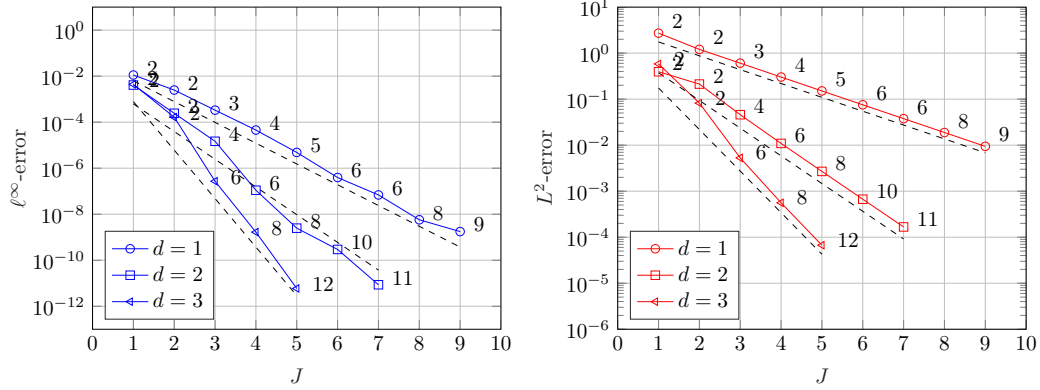


Figure IV.9: ℓ^∞ -error of the discrete potential (left) and L^2 -error of the density (right) for the Dirichlet problem on the sphere for the single layer operator. The corresponding theoretical convergence rates are h^3 , h^5 and h^7 for the potential and h^1 , h^2 and h^3 for the density. The accompanying numbers are the polynomial degrees of the interpolation.

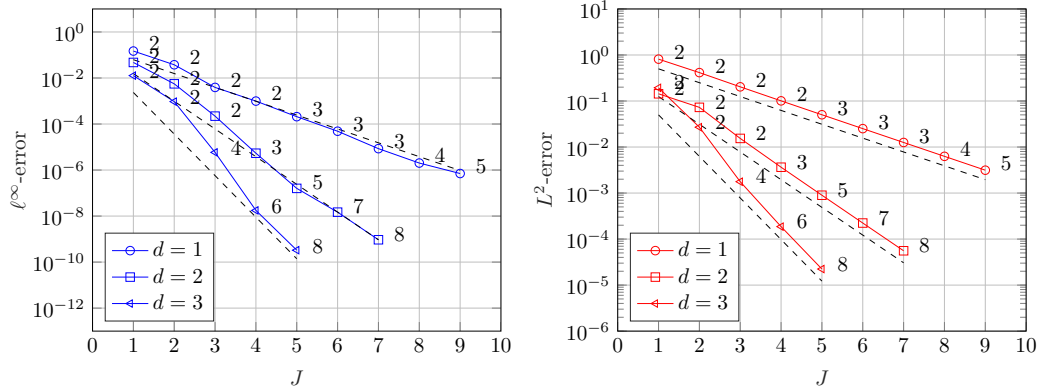


Figure IV.10: ℓ^∞ -error of the discrete potential (left) and L^2 -error of the density (right) for the Dirichlet problem on the sphere for the double layer operator. The corresponding theoretical convergence rates are h^2 , h^4 and h^6 for the potential and h^1 , h^2 and h^3 for the density. The accompanying numbers are the polynomial degrees of the interpolation.

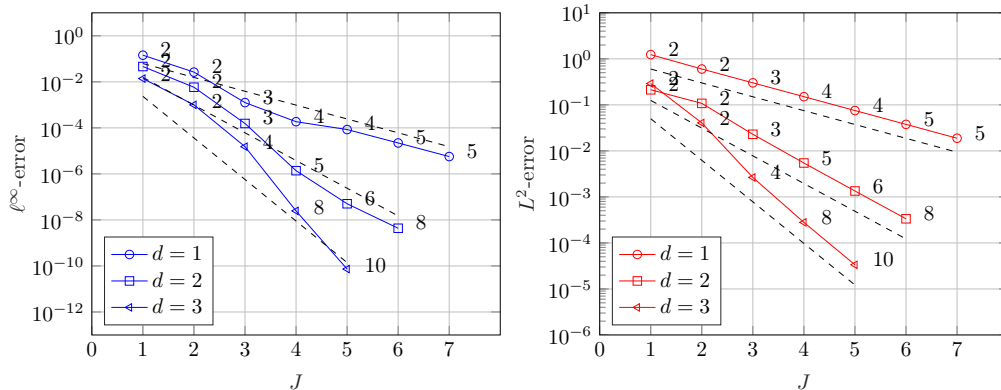


Figure IV.11: ℓ^∞ -error of the discrete potential (left) and L^2 -error of the density (right) for the Neumann problem on the sphere for the adjoint double layer operator. The corresponding theoretical convergence rates are h^2 , h^4 and h^6 for the potential and h^1 , h^2 and h^3 for the density. The accompanying numbers are the polynomial degrees of the interpolation.

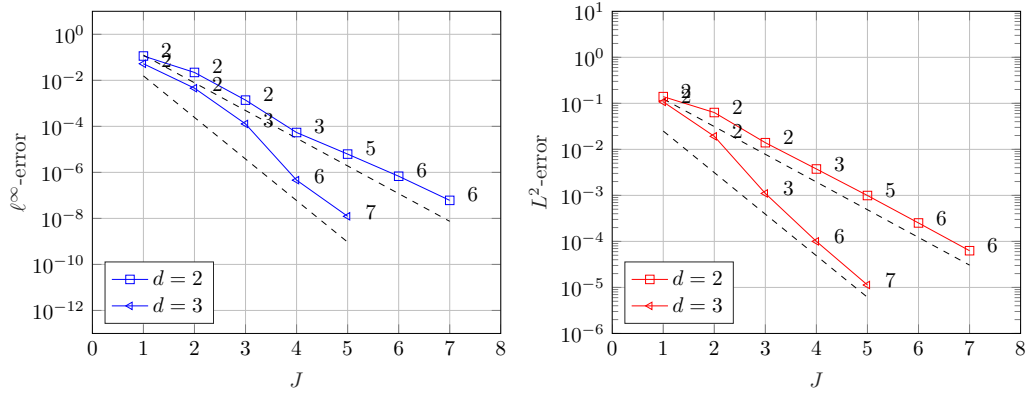


Figure IV.12: ℓ^∞ -error of the discrete potential (left) and L^2 -error of the density (right) for the Neumann problem on the sphere for the hypersingular operator. The corresponding theoretical convergence rates are h^3 and h^5 for the potential and h^2 and h^3 for the density. The accompanying numbers are the polynomial degrees of the interpolation.

		$n_J = \dim V_J$			$n_J^c = \dim V_J^c$			
		J	$d = 1$	$d = 2$	$d = 3$	$d = 2$	$d = 3$	$d = 3$
sphere	1		24	96	216	26 (3.7)	56 (3.9)	
	2		96	384	864	98 (3.9)	152 (5.7)	
	3		384	1,536	3,456	386 (4.0)	488 (7.1)	
	4		1,536	6,144	13,824	1,538 (4.0)	1,736 (8.0)	
	5		6,144	24,576	55,296	6,146 (4.0)	6,536 (8.5)	
	6		24,576	98,304		24,578 (4.0)		
	7		98,304	393,216		98,306 (4.0)		
	8		393,216					
	9		1,572,864					
gear worm	1		1,160	4,640	10,440	1,160 (4.0)	2,610 (4.0)	
	2		4,640	18,560	41,760	4,640 (4.0)	7,250 (5.8)	
	3		18,560	74,240	167,040	18,560 (4.0)	23,490 (7.1)	
	4		74,240	296,960	668,160	74,240 (4.0)	83,810 (8.0)	
	5		296,960	1,187,840		296,960 (4.0)		
	6		1,187,840					

Table IV.2: Dimensions n_J and n_J^c of the ansatz spaces V_J and V_J^c , respectively, for the sphere and the gear worm for different polynomial orders. The associated ratios n_J/n_J^c are given in the parentheses.

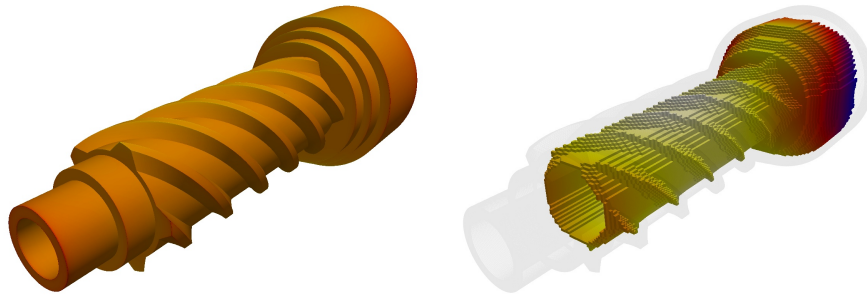


Figure IV.13: The approximate density of the single layer operator for Dirichlet data (left) and the related potential (right) for the gear worm.

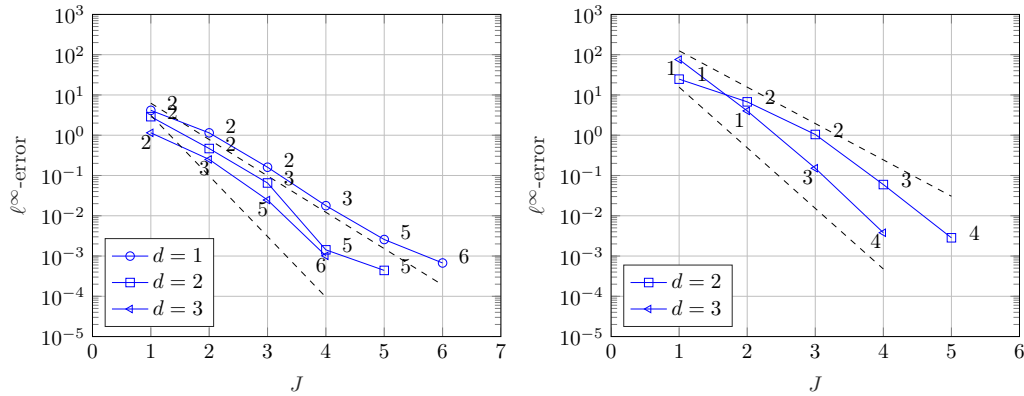


Figure IV.14: ℓ^∞ -error of the discrete potential for the Dirichlet problem for the single layer operator (left) and the Neumann problem for the hypersingular operator (right) on the gear worm. The corresponding convergence rates are h^3 and h^5 . The accompanying numbers are the polynomial degrees of the interpolation.

namely (IV.1.8), and one example for Neumann problems, namely (IV.1.12). Since the density ρ is unknown, the error of the potential is measured on the 115,241 vertices of a grid of 83,437 cubes fitted inside the domain. A visualization of the cubes together with the computed density for the single layer operator for Dirichlet data can be found in Figure IV.13.

Since the gear worm only has a Lipschitz continuous boundary, the theoretical convergence rates are limited to at most h^3 for the single layer operator and to h^1 for the hypersingular operator. Figure IV.14 illustrates that these convergence rates are achieved for all ansatz functions under consideration. In fact, the higher order ansatz functions even seem to produce a convergence rate of up to h^5 in case of the single layer operator and the hypersingular operator. Again, the numbers of local and global degrees of freedom n_J and n_J^c , respectively, associated with the discretization level J , are tabulated in Table IV.2.

IV.7.4 Computational Cost and Accuracy

In a first example, the benefit of the \mathcal{H}^2 -matrix-vector multiplication compared to the \mathcal{H} -matrix-vector multiplication shall be demonstrated. To that end, the computation times for a non-symmetric matrix-vector multiplication of an \mathcal{H} -matrix and of an \mathcal{H}^2 -matrix are measured, both stemming from the discretization of the double layer operator on the sphere or on the gear worm. The polynomial degree for the FMM is set to $p = 2$. Figure IV.15 illustrates that an asymptotic complexity of $\mathcal{O}(N_J(pd)^2)$ for the \mathcal{H}^2 -matrix-vector multiplication compared to the complexity of $\mathcal{O}(N_J \log N_J(pd)^2)$ for the \mathcal{H} -matrix-vector multiplication is reached, leading to the conclusion that the \mathcal{H}^2 -matrix-vector multiplication is the method of choice. Thus, all of the following experiments are based on the \mathcal{H}^2 -matrix-vector multiplication.

In a second example, the effectiveness of higher order ansatz functions shall be illustrated for the examples from Chapters IV.7.2 and IV.7.3. To that end, the ℓ^∞ -error of the potential is compared to the computation time of the matrix and to the computation time of the matrix plus the solving time. The results with respect to the sphere are depicted in Figure IV.16 for the Dirichlet problem and in Figure IV.17 for the Neumann problem. The results with respect to the gear worm are depicted in Figure IV.18. They indicate that the higher order ansatz functions achieve asymptotically a higher precision at a lower computation time. It therefore seems that the higher order FMM is favourable even on non-smooth surfaces. Note that the increased solving times for the gear worm geometry are due to a higher number of iterations in the solving process.

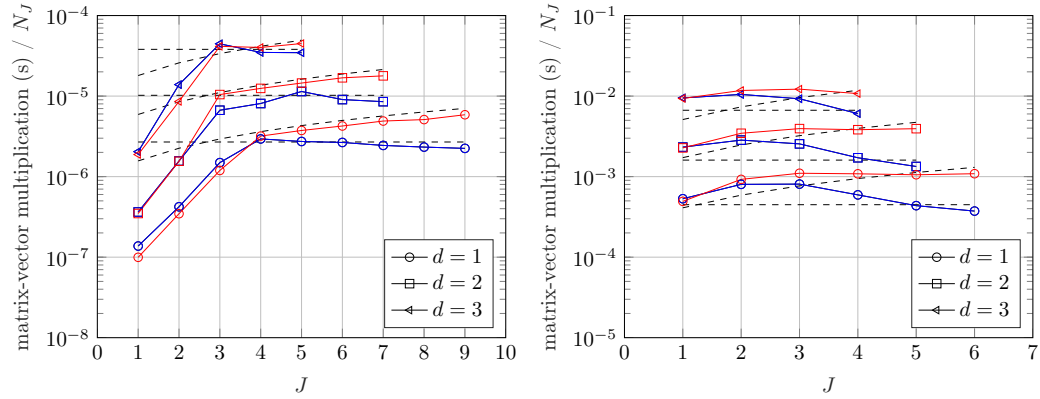


Figure IV.15: Computation times per element for the \mathcal{H}^2 -matrix-vector multiplication (blue) and the \mathcal{H} -matrix-vector multiplication (red) on the sphere (left) and on the gear worm (right). The dashed lines illustrate the complexity rates $\mathcal{O}(N_J(pd)^2)$ and $\mathcal{O}(N_J \log N_J(pd)^2)$.

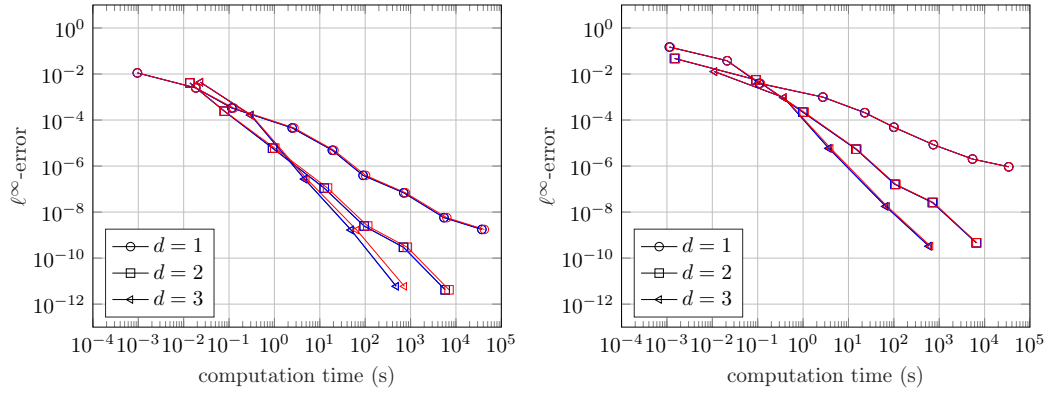


Figure IV.16: ℓ^∞ -error of the discrete potential versus the computation time of the matrix (blue) and the computation time of the matrix plus the solving time (red) for the single layer operator (left) and the double layer operator (right) on the sphere for Dirichlet data.

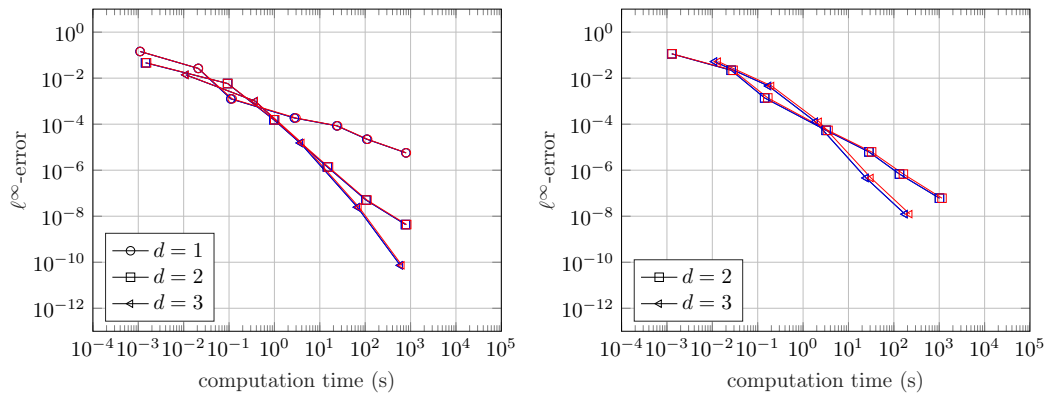


Figure IV.17: ℓ^∞ -error of the discrete potential versus the computation time of the matrix (blue) and the computation time of the matrix plus the solving time (red) for the adjoint double layer operator (left) and the hypersingular operator (right) on the sphere for Neumann data.

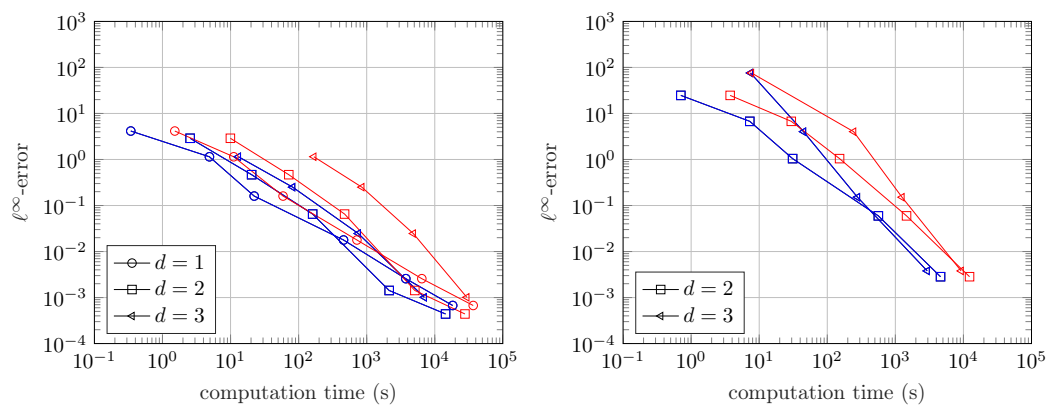


Figure IV.18: ℓ^∞ -error of the discrete potential versus the computation time of the matrix (blue) and the computation time of the matrix plus the solving time (red) for the single layer operator and Dirichlet data (left) and the hypersingular operator and Neumann data (right) on the gear worm.

IV.8 Conclusion

Parametric surfaces are easily accessible from computer aided design. They are recently of interest in isogeometric analysis, the goal of which is the direct integration of the finite element or even the boundary element analysis into the design process. This chapter proposes a fast boundary element method, namely an \mathcal{H}^2 -matrix fast multipole method based on the interpolation of the related integral kernel on the reference domain. This approach perfectly exploits the features of the parametric surface representation. By a tensor product construction and appropriate transformation matrices, one can easily deal with higher order ansatz functions. The complexity estimates as well as the numerical examples demonstrate the superior performance of the presented method.

Chapter V

PDEs with Random Dirichlet Data

V.1 Problem Formulation

After having discussed the solution of correlation equations which originate from PDEs with random load in Chapter III, the topic of this chapter is the solution of correlation equations which originate from PDEs with random Dirichlet data. More precisely, on a bounded Lipschitz domain $D \subset \mathbb{R}^3$, we consider the Laplace equation

$$\begin{aligned} -\Delta u(\omega, \mathbf{x}) &= 0 & \text{for } \mathbf{x} \in D, \\ u(\omega, \mathbf{x}) &= f(\omega, \mathbf{x}) & \text{for } \mathbf{x} \in \partial D, \end{aligned} \tag{V.1.1}$$

with $f \in L^2_{\mathbb{P}}(\Omega; H^{1/2}(\partial D))$. As discussed in the previous chapter, there exists a unique solution $u(\omega, \cdot) \in H^1(D)$ for \mathbb{P} -almost every $\omega \in \Omega$. Employing a single layer potential ansatz leads to the solution representation

$$\begin{aligned} u(\omega, \mathbf{x}) &= \tilde{\mathcal{V}}\rho(\omega, \mathbf{x}) & \text{for } \mathbf{x} \in D, \\ \mathcal{V}\rho(\omega, \mathbf{x}) &= f(\omega, \mathbf{x}) & \text{for } \mathbf{x} \in \partial D. \end{aligned}$$

We then may look at each of these equations separately to see that

$$\begin{aligned} \mathbb{E}[u] &= \tilde{\mathcal{V}}\mathbb{E}[\rho] & \text{in } D, \\ \mathcal{V}\mathbb{E}[\rho] &= \mathbb{E}[f] & \text{on } \partial D, \end{aligned}$$

and

$$\begin{aligned} \text{Cor}[u] &= (\tilde{\mathcal{V}} \otimes \tilde{\mathcal{V}}) \text{Cor}[\rho] & \text{in } D \times D, \\ (\mathcal{V} \otimes \mathcal{V}) \text{Cor}[\rho] &= \text{Cor}[f] & \text{on } \partial D \times \partial D. \end{aligned}$$

Remark V.1.1. *We consider an indirect ansatz with the single layer potential since it provides a higher order of convergence to the solution of (V.1.1). Of course, an ansatz with the double layer potential also is possible. A direct approach is also feasible, but would require the computation of $\text{Cor}[f, \partial u / \partial \mathbf{n}]$ and $\text{Cor}[\partial u / \partial \mathbf{n}]$. Thus, it would be computationally more expensive compared to the indirect methods.*

Due to the mapping properties of $\tilde{\mathcal{V}}$ and \mathcal{V} , see Theorems IV.1.1 and IV.1.3, Lemma II.1.7, and the considerations in Chapter II.4, it holds

$$\begin{aligned} \mathbb{E}[\rho] &\in H^{-1/2}(\partial D), & \text{Cor}[\rho] &\in H_{\text{mix}}^{-1/2, -1/2}(\partial D \times \partial D), \\ \mathbb{E}[u] &\in H^1(D), & \text{Cor}[u] &\in H_{\text{mix}}^{1, 1}(D \times D), \end{aligned}$$

and all of these quantities are well defined. The mapping properties of \mathcal{V} also imply that the sparse tensor product approach may yield only a reduced convergence rate compared to

the full tensor product approach when $\text{Cor}[f]$ provides little smoothness, see [DHP15] for a detailed discussion. This makes the \mathcal{H} -matrix approach attractive in the present setting.

The coefficients of the Galerkin approximations of the mean and the correlation of ρ are given by

$$\mathbf{V}\mathbf{e}_\rho = \mathbf{e}_f$$

for the mean, and by

$$\mathbf{V}\mathbf{C}_\rho\mathbf{V}^\top = \mathbf{C}_f \tag{V.1.2}$$

for the correlation. In both cases, \mathbf{V} denotes the system matrix of the single layer operator.

Exploiting that $\mathcal{V} \in OPS_{cl,s}^{-1}(\partial D)$ for all $s \geq 1$, see [HW08, Chapter 9.2], a version of Theorem II.5.18 for $\text{Cor}[\rho]$ is then given by the following corollary.

Corollary V.1.2. *Let ∂D be a compact, analytic manifold and assume that the correlation $\text{Cor}[f]$ gives rise to an operator $\mathcal{K}_{\text{Cor}[f]} \in OPS_{cl,s}^\theta(\partial D)$ of Gevrey class $s > 1$. Then, the correlation $\text{Cor}[\rho]$ is the Schwartz kernel of an operator $\mathcal{K}_{\text{Cor}[\rho]} \in OPS_{cl,s}^{\theta+2}(\partial D)$.*

Moreover, in local coordinates, $\text{Cor}[\rho]$ is smooth on $\partial D \times \partial D$ outside of the diagonal and, for local charts $\kappa_{\mathbf{x}}$ and $\kappa_{\mathbf{y}}$, there holds the pointwise estimate

$$|\partial_{\mathbf{x}}^\alpha \partial_{\mathbf{y}}^\beta \text{Cor}[\rho](\kappa_{\mathbf{x}}(\mathbf{x}), \kappa_{\mathbf{y}}(\mathbf{y}))| \leq c \mathcal{A}^{|\alpha+\beta|} (|\alpha|!)^s \beta! \|\mathbf{x} - \mathbf{y}\|^{-\theta-|\alpha|-|\beta|}$$

away from the diagonal for all $\alpha, \beta \in \mathbb{N}_0^{n-1}$. The constants c and \mathcal{A} depend only on ∂D and the symbol of $\mathcal{K}_{\text{Cor}[f]}$.

Following the discussion in Chapter II.5, we thus know that, on sufficiently smooth domains, $\text{Cor}[\rho]$ is asymptotically smooth in one variable in local coordinates. Thus, if we assume $\text{Cor}[f]$ to be analytically standard and to fulfil the assumptions of the corollary, we can represent $\text{Cor}[f]$ and, if the manifold is sufficiently smooth, also $\text{Cor}[\rho]$ by an \mathcal{H} -matrix. As the system matrix of the single layer operator can be represented by an \mathcal{H} -matrix, it is reasonable to represent all matrices occurring in (V.1.2) by \mathcal{H} -matrices and it only remains to compute the unknown \mathbf{C}_ρ .

For the underlying \mathcal{H} -matrix structure, we may use the cluster and block-cluster tree tailored to parametric surfaces from Chapter IV.2. For this special case, it turns out that several simplifications in the \mathcal{H} -matrix arithmetic can be made due to the special structure of the block-cluster tree. The \mathcal{H} -matrix equation (V.1.2) can then be solved using the solution algorithm from Chapter III.3. However, one may also use any other \mathcal{H} -matrix structure together with the general \mathcal{H} -matrix arithmetic in Chapter II.6.

V.2 \mathcal{H} -matrix Arithmetic for Parametric Surfaces

In Chapter II.6, we discussed the operation $\mathbf{H}_3 += \mathbf{H}_1 * \mathbf{H}_2$ for $\mathbf{H}_i \in \mathcal{H}(\mathcal{B}, k)$, $i = 1, 2, 3$. It was based on the hierarchical block structure (II.5.4) of \mathcal{H} -matrices induced by the block-cluster tree. This block structure becomes slightly simpler when dealing with parametric surfaces. Namely, using the notation from Chapter IV, it holds

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}|_{(1,0,0) \times (1,0,0)} & \cdots & \mathbf{H}|_{(1,0,0) \times (M,0,0)} \\ \vdots & & \vdots \\ \mathbf{H}|_{(M,0,0) \times (1,0,0)} & \cdots & \mathbf{H}|_{(M,0,0) \times (M,0,0)} \end{bmatrix} \in \mathbb{R}^{(M4^J) \times (M4^J)} \tag{V.2.1}$$

for every $\mathbf{H} \in \mathcal{H}(\mathcal{B}, k)$, where each $\mathbf{H}|_{(i,0,0) \times (i',0,0)} \in \mathbb{R}^{4^J \times 4^J}$ is a low-rank matrix, or a dense matrix if the corresponding block-cluster is a leaf, or an \mathcal{H} -matrix otherwise. \mathcal{H} -matrix blocks $\mathbf{H}|_{(i,j,\ell) \times (i',j,\ell')}$ with $j > 0$ have the structure

$$\mathbf{H}|_{(i,j,\ell) \times (i',j,\ell')} = \begin{bmatrix} \mathbf{H}|_{(i,j+1,4\ell) \times (i',j+1,4\ell')} & \cdots & \mathbf{H}|_{(i,j+1,4\ell) \times (i',j+1,4\ell'+3)} \\ \vdots & & \vdots \\ \mathbf{H}|_{(i,j+1,4\ell+3) \times (i',j+1,4\ell')} & \cdots & \mathbf{H}|_{(i,j+1,4\ell+3) \times (i',j+1,4\ell'+3)} \end{bmatrix} \in \mathbb{R}^{4^{J-j} \times 4^{J-j}}. \tag{V.2.2}$$

For ease of notation, we may write

$$\mathbf{H}|_{\lambda \times \lambda'} = \begin{bmatrix} \mathbf{H}|_{\lambda_1 \times \lambda'_1} & \cdots & \mathbf{H}|_{\lambda_1 \times \lambda'_p} \\ \vdots & & \vdots \\ \mathbf{H}|_{\lambda_p \times \lambda'_1} & \cdots & \mathbf{H}|_{\lambda_p \times \lambda'_p} \end{bmatrix} \quad (\text{V.2.3})$$

instead of (V.2.1) and (V.2.2), with $p = M$ in case of (V.2.1) and $p = 4$ in case of (V.2.2). Moreover, since the cluster tree is perfectly balanced, we can directly say that a leaf can only be a dense matrix if its level equals the depth of the tree. On the other hand, \mathcal{H} -matrix blocks can never be on the same level as dense matrices.

While these specific properties of \mathcal{H} -matrices do not change much for the algorithm of the \mathcal{H} -matrix addition, we can still use Algorithm II.6, the algorithm for the \mathcal{H} -matrix multiplication simplifies. In particular, in contrast to Chapter II.6.2, no interaction between dense matrices and \mathcal{H} -matrices occurs, i.e., we only have the following interactions for the products

*	\mathcal{H} -matrix	low-rank matrix	dense matrix
\mathcal{H} -matrix	recursively	exactly	
low-rank matrix	exactly	exactly	exactly
dense matrix		exactly	exactly

and for the compound sums

$+=_{k_3}$	\mathcal{H} -matrix	low-rank matrix	dense matrix
\mathcal{H} -matrix	recursively	recursively	
low-rank matrix	approximately	approximately	approximately
dense matrix		exactly	exactly

As in Chapter II.6, the operands in the rows of the table coincide with the target format of the respective operation.

In contrast to the idea of the *hierarchical conversion* for the operation $\mathbf{R}+=_k \mathbf{H}$, as in the original \mathcal{H} -matrix multiplication described in Chapter II.6.2, we may take another approach here: we can exploit the fact that we can multiply any \mathcal{H} -matrix \mathbf{H} in $\mathbb{R}^{4^{J-j} \times 4^{J-j}}$ to a vector $\mathbf{v} \in \mathbb{R}^{4^{J-j}}$ with a complexity of $\mathcal{O}(2k(J-j)4^{J-j})$, see Chapter II.5. Thus, it seems reasonable to directly compute the truncated singular value decomposition of \mathbf{H} up to rank k by means of an eigensolver which only requires matrix-vector multiplications as for example ARPACK, cf. [LSY98]. Then, we are again in the situation $\mathbf{R}_1+=_k \mathbf{R}_2$.

For practical issues in the product of \mathcal{H} -matrices, we rather have to consider the case $\mathbf{R}+=_k \mathbf{H}_1 * \mathbf{H}_2$. Therefore, we may exploit that the product of two \mathcal{H} -matrices $\mathbf{H}_1, \mathbf{H}_2$ of size $4^{J-j} \times 4^{J-j}$ can exactly be applied to a vector. To compute the k largest eigenvalues and eigenvectors of the product using ARPACK, let

$$\mathbf{W} := \begin{bmatrix} \mathbf{0} & \mathbf{H}_1 \mathbf{H}_2 \\ \mathbf{H}_2^T \mathbf{H}_1^T & \mathbf{0} \end{bmatrix}$$

be the *Jordan-Wielandt matrix* with respect to $\mathbf{H}_1 \mathbf{H}_2$. The positive eigenvalues of this matrix coincide with the singular values of $\mathbf{H}_1 \mathbf{H}_2$, cf. [GK65, Lan61]. The complexity of applying \mathbf{W} to a vector $\mathbf{x} \in \mathbb{R}^{2 \cdot 4^{J-j}}$ is of order $\mathcal{O}(8k(J-j)4^{J-j})$. Thus, the computation of the k largest singular values of \mathbf{W} with the corresponding singular vectors can be performed within a complexity of $\mathcal{O}(8k(J-j)4^{J-j} \text{ncv}^2)$, where ncv corresponds to the size of the Krylov subspace used for the eigenvalue approximation, cf. [LSY98]. From the singular values and singular vectors of \mathbf{W} , one can easily derive an SVD of $\mathbf{H}_1 \mathbf{H}_2$, to which we refer to as the *indirect singular value decomposition* or *iSVD*.

Remark V.2.1. We may replace $\mathbf{H}_1\mathbf{H}_2$ with $\mathbf{H}_1^{(1)}\mathbf{H}_2^{(1)} + \dots + \mathbf{H}_1^{(p)}\mathbf{H}_2^{(p)}$ in the Jordan-Wielandt matrix. Then, the iSVD can directly compute the best approximation of this sum by a low-rank matrix \mathbf{R} . Realizing that such a sum has the same structure as required in the block matrix multiplication of two matrices, we may use the iSVD to compute the block inner-product of a block-row and a block-column in the product of two \mathcal{H} -matrices $\mathbf{H}_1|_{\lambda \times \lambda''}$ and $\mathbf{H}_2|_{\lambda'' \times \lambda}$ on level $j-1$ structured like (V.2.3):

$$\mathbf{R} = \mathcal{T}_k \left(\sum_{\ell=1}^p \mathbf{H}_1|_{\lambda_i \times \lambda''_\ell} \mathbf{H}_2|_{\lambda''_\ell \times \lambda_j} \right).$$

Here, we have either $p = M$ for level 0 or $p = 4$ for any other level. Then, the complexity is $\mathcal{O}(8k(J-j)4^{J-j}p \cdot \text{ncv}^2)$.

Applying recursively the procedure from the previous remark actually yields the best approximation of the product $\mathbf{H}_1\mathbf{H}_2$ in $\mathcal{H}(\mathcal{B}, k)$. A naive realization of the product $\mathbf{H}_1 * \mathbf{H}_2 \in \mathcal{H}(\mathcal{B}, k)$ is provided by Algorithm V.1. Note that the calling sequence for Algorithm V.1 is initiated with $p' = 1$.

Algorithm V.1 Compute $\mathbf{H}_3|_{\lambda \times \lambda'} \leftarrow \sum_{i=1}^{p'} \mathbf{H}_1^{(i)}|_{\lambda \times \lambda''} \mathbf{H}_2^{(i)}|_{\lambda'' \times \lambda'}$

```

function  $\mathbf{H}_3|_{\lambda \times \lambda'} \leftarrow \text{BESTMULT}\mathcal{H}(\{\mathbf{H}_1^{(i)}|_{\lambda \times \lambda''}\}_{i=1}^{p'}, \{\mathbf{H}_2^{(i)}|_{\lambda'' \times \lambda'}\}_{i=1}^{p'})$ 
  if  $\lambda \times \lambda' \notin \mathcal{L}(\mathcal{B})$  then
    for  $\ell, \ell' = 1, \dots, p$  do
       $\mathcal{L} = \cup_{i=1}^{p'} \{ \mathbf{H}_1^{(i)}|_{\lambda_\ell \times \lambda''_1}, \dots, \mathbf{H}_1^{(i)}|_{\lambda_\ell \times \lambda''_p} \}$ 
       $\mathcal{R} = \cup_{i=1}^{p'} \{ \mathbf{H}_2^{(i)}|_{\lambda''_1 \times \lambda_{\ell'}}, \dots, \mathbf{H}_2^{(i)}|_{\lambda''_p \times \lambda_{\ell'}} \}$ 
       $\mathbf{H}_3|_{\lambda \times \lambda'} = \text{BESTMULT}\mathcal{H}(\mathcal{L}, \mathcal{R})$ 
    end for
  else
    Compute  $\mathbf{H}_3 = \mathfrak{T}(\sum_{i=1}^{p'} \mathbf{H}_1^{(i)}\mathbf{H}_2^{(i)})$  with iSVD or as full matrix
  end if
end function

```

An efficient implementation of the algorithm requires a more careful handling of the recursion to avoid the exponential growth of \mathcal{L} and \mathcal{R} , which has in the meantime also been investigated in [Bör17] using accumulated low-rank updates. Although one can use the same recursion scheme for the use with an iterative eigensolver, the accumulated low-rank updates seem to have an advantage over the \mathcal{H} -matrix arithmetic from [GH03], while our numerical experiments indicate that the iterative eigensolver from ARPACK is slower. We have reason to believe that this is caused by the slow convergence of the eigensolver in case of a clustering of the eigenvalues. Therefore, for practical purposes, we rather use Algorithm II.4 with a modified operation $\mathbf{R} \leftarrow_k \mathbf{H}_1 * \mathbf{H}_2$ by computing a low-rank approximation $\tilde{\mathbf{R}}$ to $\mathbf{H}_1\mathbf{H}_2$ with the iSVD and then computing $\mathbf{R} \leftarrow_k \tilde{\mathbf{R}}$.

V.3 Numerical Examples

The following numerical experiments are divided into three parts. The first part is dedicated to the verification of the asymptotic complexity of the developed \mathcal{H} -matrix arithmetic tailored to parametric surfaces. The second part shall be concerned with the application of the numerical solver from Chapter III.3 to (V.1.2), whereas the third part is concerned with a more complex example. All of the computations in the following experiments were carried out on a computing server with two Intel(R) Xeon(R) CPU E5-2643 v3 with a clock rate of 3.40GHz and a main memory of 256GB. Each of the CPUs provides 12 physical cores, thus, with Hyper-Threading enabled, we may access up to 24 logical cores in total. We use the \mathcal{H} -matrix arithmetic from the previous section for all numerical experiments.

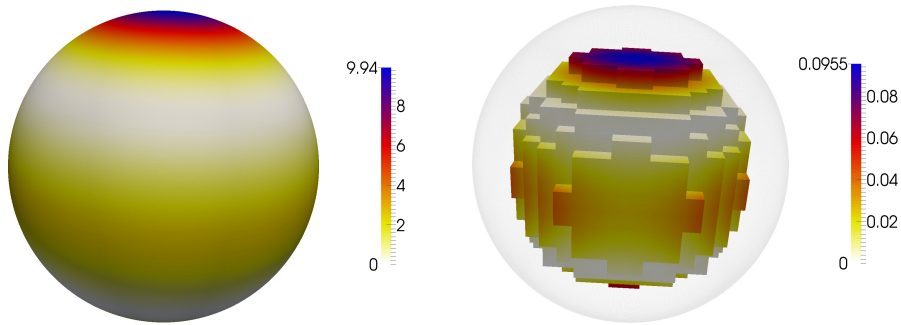


Figure V.1: Visualization of the diagonal of the correlation of the density (left) and the potential (right) on the unit sphere.

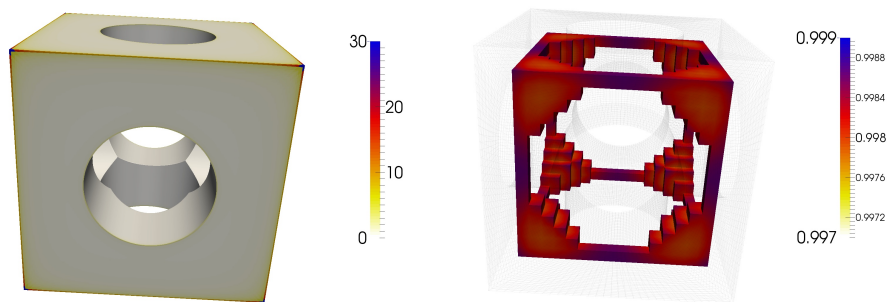


Figure V.2: Visualization of the diagonal of the correlation of the density (left) and the potential (right) on the toy geometry.

V.3.1 Tests for the \mathcal{H} -matrix Arithmetic

We consider the unit sphere parameterized by six patches and a pierced cube with circular holes on each face, to which we will refer to as toy geometry, represented by 48 patches. A visualization of the sphere is shown in Figure V.1, while a visualization of the toy geometry can be found in Figure V.2.

On each of the geometries, we assemble two \mathcal{H} -matrices $\mathbf{V}, \mathbf{S} \in \mathcal{H}(\mathcal{B}, 25)$ using the FMM from Chapter IV with $p = 5$ and piecewise (discontinuous) polynomial boundary elements of order $d = 1, 2, 3$. The matrix \mathbf{V} corresponds to the discrete single layer operator for the Laplace equation, whereas \mathbf{S} corresponds to the exponential kernel with correlation length $\ell = 1$, i.e., setting $\mu = 1/2$ in (II.4.5):

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|).$$

The upper threshold for the rank in \mathcal{H} -matrix arithmetic is set to $k = p^2 = 25$.

In addition to the operators $+_{25}$ and $*_{25}$ for non-symmetric matrices, we also consider their symmetric versions $\hat{+}_{25}$ and $\hat{*}_{25}$ which compute only the lower triangular part of the result. Additionally, we compute the Cholesky factorization of \mathbf{V} and its application to \mathbf{S} . The computations are run on a single core.

The computation times consumed for all considered operations are tabulated in Table V.1 in case of the unit sphere and in Table V.2 in case of the toy geometry. Additionally, Figures V.3, V.4, and V.5 show the asymptotic behavior of the computational times. For the sake of simplicity, we only show the timings for the symmetric addition and multiplication. It

	J	n_J	$\mathbf{V}^{\hat{+}_{25}\mathbf{S}}$	$\mathbf{V}^{+_{25}\mathbf{S}}$	$\mathbf{V}^{\hat{*}_{25}\mathbf{S}}$	$\mathbf{V}^{*_{25}\mathbf{S}}$	Chol(\mathbf{V})	For+Back
$d=1$	1	24	4.1e-05	3.5e-05	7.7e-05	0.000125	5.3e-05	9.3e-05
	2	96	6.5e-05	0.0001	0.000443	0.000686	0.00026	0.000799
	3	384	0.000302	0.00119	0.022912	0.052417	0.009747	0.043785
	4	1,536	0.044713	0.090498	3.5116	7.06414	1.98042	7.91452
	5	6,144	0.867404	1.72426	52.055	107.143	41.8888	146.623
	6	24,576	4.46785	8.64202	316.613	654.01	287.448	913.447
	7	98,304	21.463	42.2797	1,798.45	3,828.5	1,868.96	5,393.18
	8	393,216	99.8646	201.213	8,990.93	19,848.9	10,005.6	25,913.5
$d=2$	1	96	5.1e-05	9.2e-05	0.000411	0.00068	0.000277	0.000801
	2	384	0.000425	0.000978	0.022478	0.041053	0.009165	0.042428
	3	1,536	0.043059	0.087436	3.37257	6.78586	1.88509	7.70771
	4	6,144	0.787595	1.58746	46.8514	95.5437	37.0832	133.922
	5	24,576	4.07454	8.22546	285.494	586.663	247.38	808.526
	6	98,304	19.7881	39.5271	1,606.97	3,351.7	1,534.99	4,470.37
	7	393,216	97.5343	199.849	8,703.37	19,216.6	9,513.69	24,675.5
$d=3$	1	216	0.000144	0.00035	0.003378	0.00467	0.001933	0.005735
	2	864	0.001641	0.005739	0.193533	0.290137	0.100178	0.301842
	3	3,456	0.090337	0.176859	11.3008	20.5841	5.78048	22.0896
	4	13,824	1.42118	2.624	104.773	206.03	73.3897	267.537
	5	55,296	7.45	14.9491	656.893	1,327.73	517.241	1,758.13
	6	221,184	39.8873	78.4722	3,842.1	7,947.72	3,282.52	10,256.1

 Table V.1: Computational times for each particular \mathcal{H} -matrix operation on the unit sphere.

seems that we obtain a rate of $N_J \log N_J$ for the \mathcal{H} -matrix addition and $N_J(\log N_J)^2$ for the other operations, which is in accordance with the complexity estimates from Chapter II.6.

As in Chapter III, we are also interested in the quality of the approximate Cholesky factorization $\mathbf{V} \approx \hat{\mathbf{L}}\hat{\mathbf{L}}^\top$. Again, we use ten power iterations to estimate the error $\hat{\mathbf{L}}\hat{\mathbf{L}}^\top - \mathbf{V}$ in the spectral norm. The estimated errors are plotted in Figures V.6, V.7, and V.8. We remark that the estimated errors are in the same range as the errors computed with the sparse eigensolver ARPACK. The jumps in the graphs occur when the first low-rank blocks appear in the \mathcal{H} -matrix partition and the arithmetic changes from the exact arithmetic of dense matrices to the approximate \mathcal{H} -matrix arithmetic.

V.3.2 Tests for the Iterative Solver

Having verified the almost linear scaling of our implementation of the \mathcal{H} -matrix arithmetic, we shall now see how it performs in solving the system of linear equations (V.1.2). We provide two simple numerical examples in order to show that the solution strategy from Chapter III.3 also works in the case of boundary element matrices. To obtain computational efficiency, we exploit the symmetric structure whenever possible, i.e., we only store and compute the lower part of the matrices.

In the first example, we consider the unit ball as a domain and

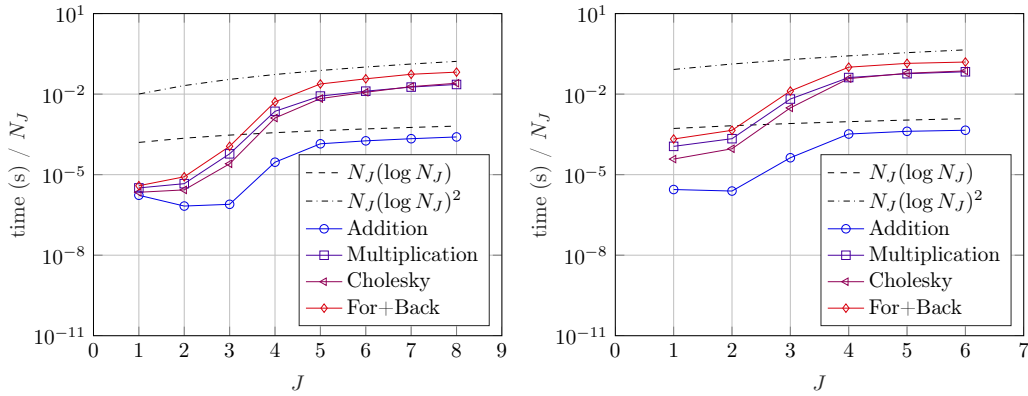
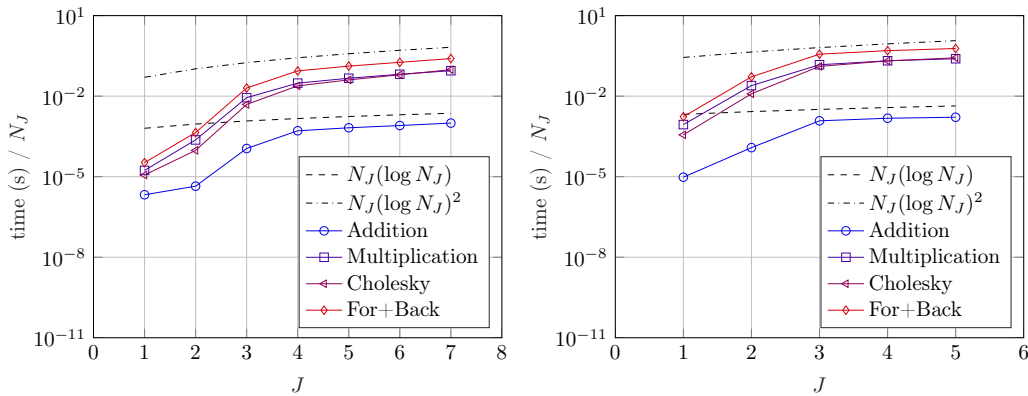
$$\text{Cor}[f] = Y_0^2 \otimes Y_0^2,$$

where $Y_0^2(\mathbf{x}) = \sqrt{5/(16\pi)}(3x_3^2 - 1)$ is the spherical harmonic, which we also considered in the numerical experiments of Chapter IV. Due to the tensor product structure of $\text{Cor}[f]$, we can then directly say that

$$\text{Cor}[u] = Y_0^2 \otimes Y_0^2.$$

This provides us with an analytical reference solution, to which we compare our numerical approximation on the 2,351 vertices of the cubes visualized in Figure V.1.

	J	n_J	$\mathbf{V}_{+25}^* \mathbf{S}$	$\mathbf{V}_{+25} \mathbf{S}$	$\mathbf{V}_{*25}^* \mathbf{S}$	$\mathbf{V}_{*25} \mathbf{S}$	Chol(\mathbf{V})	For+Back
$d=1$	1	192	0.000536	0.001095	0.021754	0.045326	0.007315	0.04081
	2	768	0.001884	0.004775	0.167881	0.316927	0.070815	0.344601
	3	3,072	0.130925	0.18407	20.1432	39.3158	9.43328	40.0813
	4	12,288	4.00772	7.97778	508.534	1,032.57	465.559	1,238.12
	5	49,152	20.2409	39.0739	2,828.7	5,780.24	2,948.81	6,868.8
	6	196,608	88.6684	178.078	13,362.7	26,472.9	14,404.8	30,795.8
$d=2$	1	768	0.00184	0.004926	0.167472	0.314753	0.069343	0.328008
	2	3,072	0.0927	0.177984	18.981	36.9432	9.40632	40.1611
	3	12,288	3.70498	7.43748	453.663	914.985	394.168	1,126.11
	4	49,152	18.4753	36.9277	2,556.9	5,139.18	2,524.57	6,083.86
	5	196,608	80.5977	165.36	12,158.2	24,717.3	13,175.8	29,280
$d=3$	1	1,728	0.009992	0.025794	1.40425	2.25854	0.664645	2.32553
	2	6,912	0.230159	0.506579	95.5978	163.796	45.5567	171.079
	3	27,648	6.11852	12.1955	1,013.91	1,984.87	745.913	2,222.24
	4	110,592	34.3278	68.8413	5,616.83	11,232	4,959.89	12,687.3

 Table V.2: Computational times for each particular \mathcal{H} -matrix operation on the toy geometry.

 Figure V.3: Asymptotic behavior of the computation times per element on the unit sphere (left) and on the toy geometry (right) for $d = 1$.

 Figure V.4: Asymptotic behavior of the computation times per element on the unit sphere (left) and on the toy geometry (right) for $d = 2$.

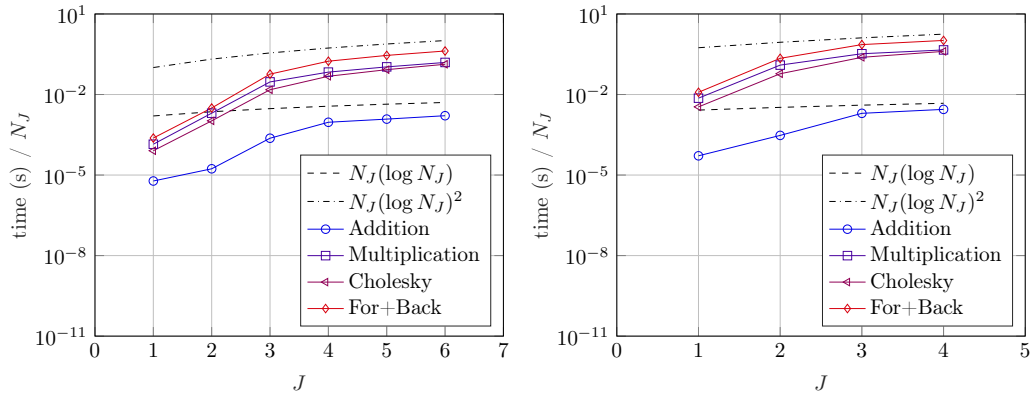


Figure V.5: Asymptotic behavior of the computation times per element on the unit sphere (left) and on the toy geometry (right) for $d = 3$.

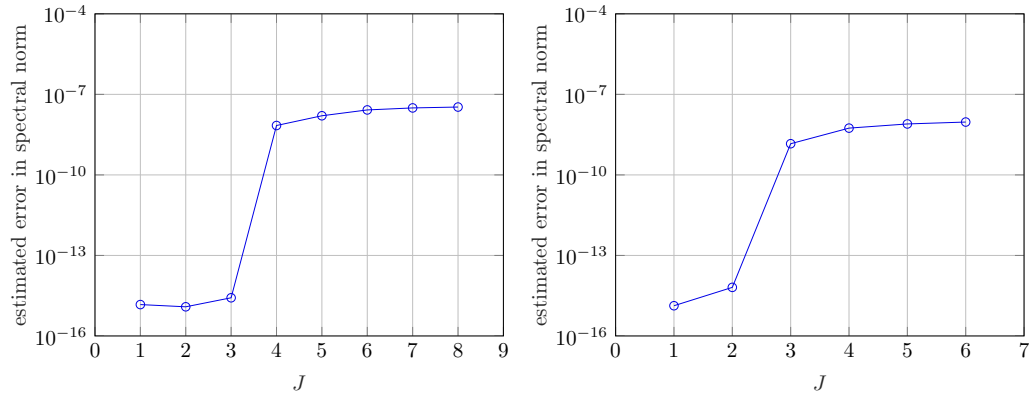


Figure V.6: Deviation $\hat{\mathbf{L}}\hat{\mathbf{L}}^\top - \mathbf{V}$ with respect to the estimated spectral norm for fixed rank on the unit sphere (left) and on the toy geometry (right) for $d = 1$.

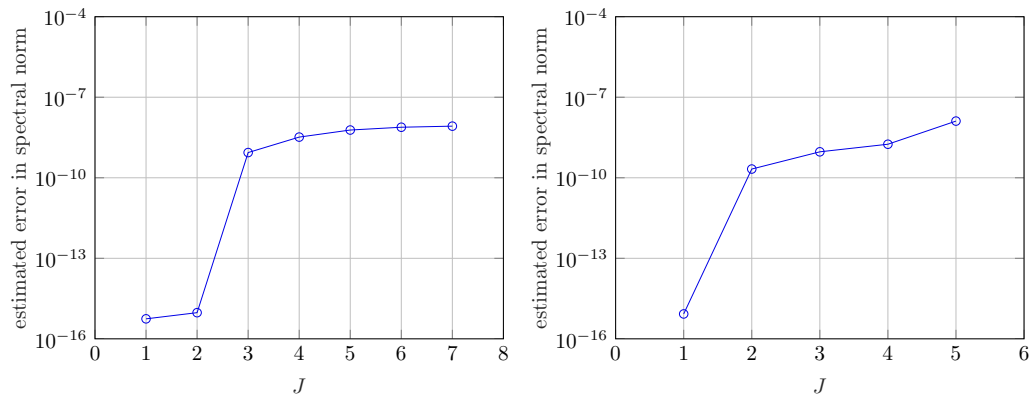


Figure V.7: Deviation $\hat{\mathbf{L}}\hat{\mathbf{L}}^\top - \mathbf{V}$ with respect to the estimated spectral norm for fixed rank on the unit sphere (left) and on the toy geometry (right) for $d = 2$.

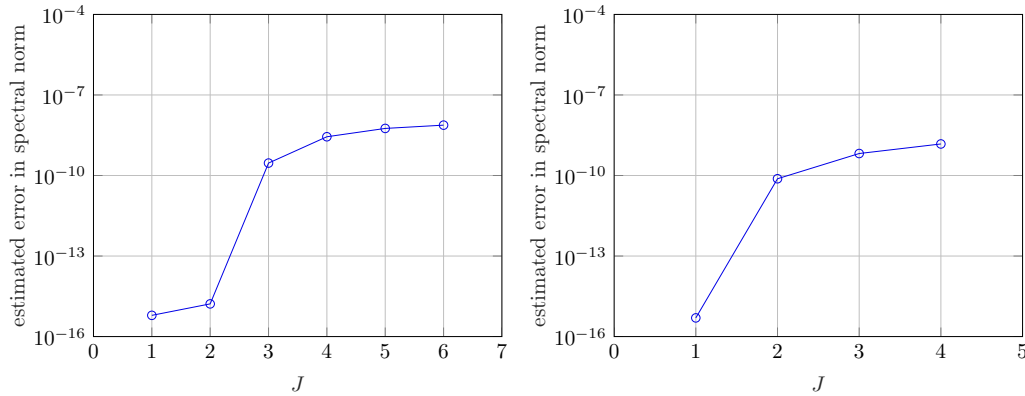


Figure V.8: Deviation $\hat{\mathbf{L}}\hat{\mathbf{L}}^\top - \mathbf{V}$ with respect to the estimated spectral norm for fixed rank on the unit sphere (left) and on the toy geometry (right) for $d = 3$.

For the second example, we consider the toy geometry as a domain and choose a Gaussian kernel as input data, i.e., we set

$$\text{Cor}[f](\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\ell^2}\right)$$

with $\ell \approx 4\text{diam}(D)$. Since we cannot analytically compute $\text{Cor}[u]$ in this case, we may compute a reference solution with a similar procedure as in the numerical experiments of Chapter III. That is, we compute an overkill solution to $\text{Cor}[\rho]$ on a finer mesh by computing a low-rank factorization of \mathbf{C}_f , using the pivoted Cholesky factorization, to obtain a low-rank factorization of \mathbf{C}_ρ . A subsequent potential evaluation yields a reference solution to $\text{Cor}[u]$, to which we compare our solution on the 1,352 vertices of the cubes visualized in Figure V.2.

For the numerical computations, we allow a maximal rank of $k = 25$ and stop the iteration when the relative error of the residuals' Frobenius norm is smaller than $\varepsilon = 10^{-6}$. For the Galerkin discretization, we use the discontinuous, higher order boundary elements of order $d = 1, 2, 3$ from the previous chapter. The errors in Figure V.9 indicate that we indeed reach the theoretical possible convergence rates on the sphere and also on the toy geometry. The computation times for the iterative refinement are tabulated in Table V.3 and visualized in Figure V.10. They seem to confirm the almost linear behaviour of the solver, although the computation times on the sphere for $d = 3$ seem to be in the preasymptotic regime. A visualization of the diagonal of the density's and the potential's correlation can be found in Figure V.1 for the sphere and in Figure V.2 for the toy geometry.

Notice that, for both examples, we had to perform only one step to achieve convergence to the prescribed tolerance.

V.3.3 Stochastic Application

Having verified the convergence of the iterative refinement also in the case of correlation equations from a discretization with a single layer potential ansatz, we shall now exemplarily compute the solution's correlation on the crankshaft geometry from Figure IV.2, represented by 142 patches. The prescribed correlation on the boundary is either the Gaussian kernel

$$\text{Cor}[f](\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2}\right)$$

or the exponential kernel

$$\text{Cor}[f](\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|).$$

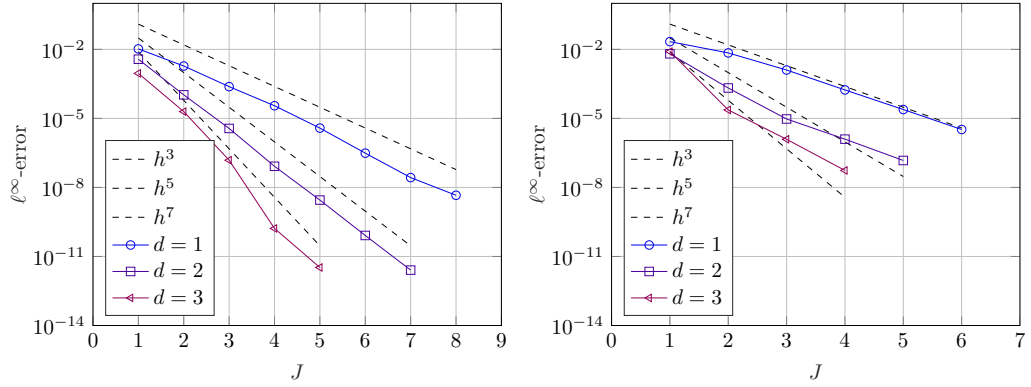


Figure V.9: Errors for the correlation of the potential on the sphere (left) and on the toy geometry (right).

J	sphere			toy		
	$d = 1$	$d = 2$	$d = 3$	$d = 1$	$d = 2$	$d = 3$
1	0.000429	0.002644	0.020124	0.140119	1.28216	9.75094
2	0.016963	0.143579	1.17804	24.1919	173.213	676.472
3	3.16681	19.9488	67.9204	3,555.34	6,928.16	12,975.3
4	20.2207	259.415	520.829	7,767.52	27,714.5	83,684.4
5	181.909	1,323.76	8,381.58	37,308.7	142,455	
6	1,083.54	7,423.32		172,704		
7	6,088.71	41,599.9				
8	40,843.9					

Table V.3: Computation times for the iterative refinement on the sphere and the toy geometry.

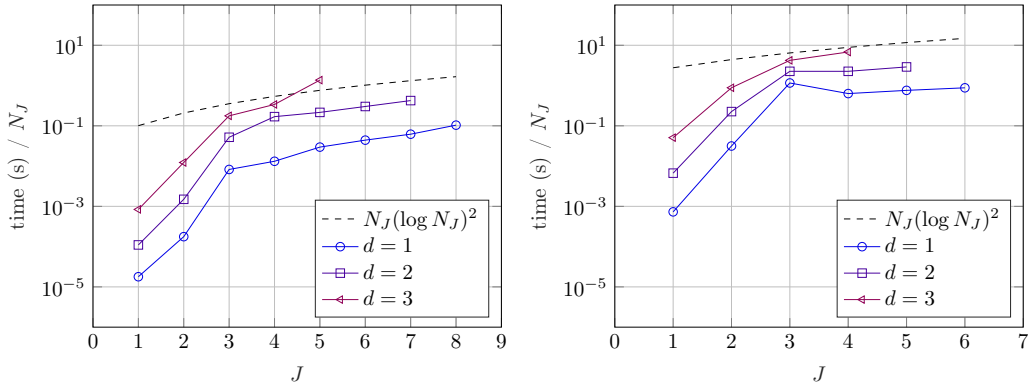


Figure V.10: Computation times per element for the iterative refinement on the sphere (left) and on the toy geometry (right).

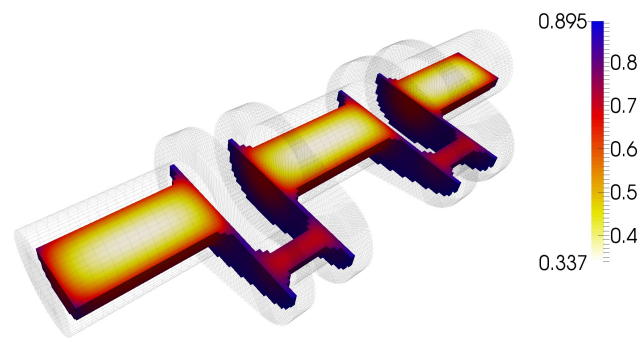


Figure V.11: The approximate diagonal of the correlation of the potential for the Gaussian kernel.

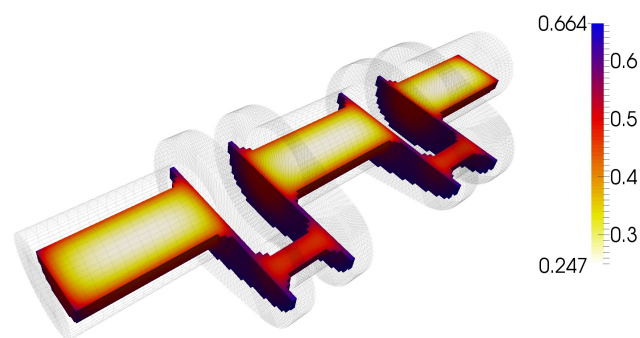


Figure V.12: The approximate diagonal of the correlation of the potential for the exponential kernel.

Since the diameter of the crankshaft is approximately 13, the ratio of correlation length to geometry diameter is around $1/13$. We use discontinuous, piecewise linear boundary elements on level four, yielding 145,408 ansatz functions and, thus, a matrix equation with $2.1 \cdot 10^{10}$ unknowns. We employ a simple OpenMP parallelization on the first recursion level of the \mathcal{H} -matrix algorithms to accelerate the computations and refer to [Kri05], where a more sophisticated parallelization was discussed. To avoid memory racing conditions, we modify the original ARPACK code to remove static variables. The maximal admissible rank for the \mathcal{H} -matrix arithmetic is again set to 25.

Figure V.11 and Figure V.12 show the diagonal of the correlation of the potential for the Gaussian kernel and for the exponential kernel, respectively. In both cases, the iterative refinement needed one iteration to converge. The computation of the Cholesky factorization takes about 1,800 seconds, whereas the iterative refinement takes about 23,000 seconds.

V.4 Conclusion

In this chapter, we have solved correlation equations stemming from PDEs with random Dirichlet data with the \mathcal{H} -matrix approach. Using an indirect boundary integral formulation, we have computed the correlation of the density. A subsequent potential evaluation on the tensor product domain allows for the computation of the correlation of the PDE's solution. Using the \mathcal{H} -matrix framework for parametric surfaces, we end up with specially structured \mathcal{H} -matrices. The combination of this structure with a sparse eigensolver results in an efficient \mathcal{H} -matrix arithmetic. The numerical examples validate the almost linear cost complexity for the \mathcal{H} -matrix arithmetic. Furthermore, the numerical examples confirm that the iterative solver is well suited to numerically solve the correlation equation. Finally, results are given

for stochastic problems with the Gaussian kernel and the exponential kernel on a nontrivial geometry.

Chapter VI

PDEs on Random Domains

VI.1 Random Domains

Up to now, we have computed the correlation of solutions of PDEs with random load or random boundary values. In both cases, the derivation of the correlation equation was essentially the tensoration of the original linear equation and subsequent integration over the probability space. We shall now consider PDEs on random domains, where, in contrast to the previous examples, the solution depends non-linearly on the uncertain input data.

VI.1.1 Basic Definitions

For the following considerations, let $D_0 \subset \mathbb{R}^n$ be a reference domain with, in order to ensure $C^{3,1}$ -continuity of its outer normal, a $C^{4,1}$ -boundary. On a separable, complete probability space $(\Omega, \Sigma, \mathbb{P})$, consider a random vector field $\mathbf{V} \in L^2_{\mathbb{P}}(\Omega; C^{3,1}(\partial D_0; \mathbb{R}^n))$ with $\|\mathbf{V}(\omega, \cdot)\|_{C^{3,1}(\partial D_0; \mathbb{R}^n)} \lesssim 1$ uniformly for all $\omega \in \Omega$, which perturbs the boundary of the reference domain ∂D_0 in accordance with $\partial D_{\varepsilon}(\omega) := \partial D_0 + \varepsilon \mathbf{V}(\omega, \partial D_0)$ for some given $\varepsilon > 0$. A *random domain* $D_{\varepsilon}(\omega)$ is then given by the interior of the perturbed boundary $\partial D_{\varepsilon}(\omega)$. For later considerations, we also introduce a compact set K , which is contained in all possible boundary variations and, in particular, in the reference domain, i.e.,

$$K \Subset D_0 \cap D_{\varepsilon}^{\cap \Omega}, \quad D_{\varepsilon}^{\cap \Omega} := \bigcap_{\omega \in \Omega} D_{\varepsilon}(\omega).$$

The complete setting of the introduced sets is illustrated in Figure VI.1.

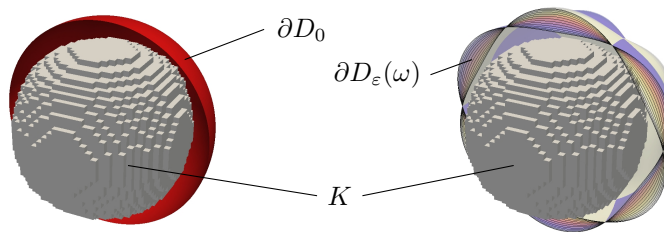


Figure VI.1: The boundary of the reference domain ∂D_0 , a family of perturbed domains $\partial D_{\varepsilon}(\omega)$, and the compactum K .

Note that, in contrast to the domain mapping approach, which requires a vector field on the whole reference domain, see Chapter I and the references therein, the perturbation approach only requires a vector field on the boundary. A correspondence between the two approaches is given by the fact that every vector field on the boundary ∂D_0 can smoothly be extended into the interior of D_0 in such a way that it vanishes on the compactum K .

For all further considerations, let $g \in H^3(D_0 \cup D_\varepsilon^{\cup\Omega})$, where $D_\varepsilon^{\cup\Omega}$ denotes the hold-all domain

$$D_\varepsilon^{\cup\Omega} := \bigcup_{\omega \in \Omega} D_\varepsilon(\omega).$$

Then, it follows $u_0 \in H^3(D_0)$ for the unique solution u_0 of the unperturbed problem

$$\begin{aligned} \Delta u_0 &= 0 && \text{in } D_0, \\ u_0 &= g && \text{on } \partial D_0, \end{aligned} \tag{VI.1.1}$$

see [Wlo87, Theorem 20.4] for example. Here and in the following, the Dirichlet and Neumann data have to be understood in the trace sense, see Theorems II.2.20 and II.2.21.

VI.1.2 Shape Calculus for Parametrized Domains

We look at the randomly perturbed boundary value problem

$$\begin{aligned} \Delta u_\varepsilon(\omega) &= 0 && \text{in } D_\varepsilon(\omega), \\ u_\varepsilon(\omega) &= g && \text{on } \partial D_\varepsilon(\omega), \end{aligned} \tag{VI.1.2}$$

which is posed on the random domain $D_\varepsilon(\omega)$. Given the reference domain D_0 , and the vector field $\mathbf{V} \in L^2_{\mathbb{P}}(\Omega; C^{3,1}(\partial D_0; \mathbb{R}^n))$ with $\|\mathbf{V}(\omega, \cdot)\|_{C^{3,1}(\partial D_0; \mathbb{R}^n)} \lesssim 1$ uniformly for all $\omega \in \Omega$, one can expand u_ε into a *shape Taylor expansion*

$$u_\varepsilon(\omega, \mathbf{x}) = u_0(\mathbf{x}) + \varepsilon \delta u[\mathbf{V}(\omega)](\mathbf{x}) + \frac{\varepsilon^2}{2} \delta^2 u[\mathbf{V}(\omega), \mathbf{V}(\omega)](\mathbf{x}) + \mathcal{O}(\varepsilon^3), \quad \mathbf{x} \in K \Subset D_0 \cap D_\varepsilon(\omega), \tag{VI.1.3}$$

which holds for all $0 < \varepsilon \leq \varepsilon_0$ for some $\varepsilon_0 > 0$ small enough. Here, the *first order local shape derivative* $\delta u[\mathbf{V}(\omega)] \in H^2(D_0)$ is given by

$$\begin{aligned} \Delta \delta u[\mathbf{V}(\omega)] &= 0 && \text{in } D_0, \\ \delta u[\mathbf{V}(\omega)] &= \langle \mathbf{V}(\omega), \mathbf{n} \rangle \frac{\partial(g - u_0)}{\partial \mathbf{n}} && \text{on } \partial D_0, \end{aligned} \tag{VI.1.4}$$

cf. [SZ92, Chapter 3.1]. Given a second vector field $\mathbf{V}' \in L^2_{\mathbb{P}}(\Omega; C^{3,1}(D_0; \mathbb{R}^n))$ for which it holds $\|\mathbf{V}'(\omega, \cdot)\|_{C^{3,1}(D_0; \mathbb{R}^n)} \lesssim 1$ uniformly for all $\omega \in \Omega$, the *second order local shape derivative* $\delta^2 u[\mathbf{V}(\omega), \mathbf{V}'(\omega)] \in H^1(D_0)$ is given, cf. [Epp00, Theorem 1], by

$$\begin{aligned} \Delta \delta^2 u[\mathbf{V}(\omega), \mathbf{V}'(\omega)] &= 0 && \text{in } D_0, \\ \delta^2 u[\mathbf{V}(\omega), \mathbf{V}'(\omega)] &= \frac{\partial^2(g - u_0)}{\partial \mathbf{n}^2} \langle \mathbf{V}(\omega), \mathbf{n} \rangle \langle \mathbf{V}'(\omega), \mathbf{n} \rangle \\ &\quad - \langle \mathbf{V}(\omega), \mathbf{n} \rangle \frac{\partial \delta u[\mathbf{V}'(\omega)]}{\partial \mathbf{n}} \\ &\quad - \langle \mathbf{V}'(\omega), \mathbf{n} \rangle \frac{\partial \delta u[\mathbf{V}(\omega)]}{\partial \mathbf{n}} && \text{on } \partial D_0. \end{aligned} \tag{VI.1.5}$$

Of course, according to (VI.1.3), we only need the second order local shape derivative in case of identical fields, i.e., $\mathbf{V} = \mathbf{V}'$ and it only remains to explain how to compute $\partial^2(g - u_0)/\partial \mathbf{n}^2$ on ∂D_0 . On a C^2 -boundary and for $\psi \in H^3(D)$, we know that, cf. [SZ92, Proposition 2.68],

$$\Delta \psi = \Delta_\Gamma \psi + 2\mathcal{H} \frac{\partial \psi}{\partial \mathbf{n}} + \frac{\partial^2 \psi}{\partial \mathbf{n}^2} \quad \text{on } \partial D_0, \tag{VI.1.6}$$

where Δ_Γ denotes the *Laplace-Beltrami operator* and \mathcal{H} denotes the *mean curvature* of ∂D_0 given by $2\mathcal{H} = -\nabla \cdot \mathbf{n}$. Since $u_0 = g$ on ∂D_0 , see (VI.1.1), we have $\Delta_\Gamma(g - u_0) = 0$, and obtain

$$\frac{\partial^2(g - u_0)}{\partial \mathbf{n}^2} = \Delta g - 2\mathcal{H} \frac{\partial(g - u_0)}{\partial \mathbf{n}} \quad \text{on } \partial D_0 \tag{VI.1.7}$$

by the use of (VI.1.6).

VI.1.3 Statistical Moments on Random Domains

For the following considerations, we can assume without loss of generality that the boundary perturbations are centered, i.e.,

$$\mathbb{E}[\mathbf{V}] = \mathbf{0}, \quad \text{and thus formally } \mathbb{E}[D_\varepsilon] = D_0,$$

which is not a restriction, since one can easily recenter the random field by considering $\mathbf{V}' = \mathbf{V} - \mathbb{E}[\mathbf{V}]$. This especially means that the mean of the first order local shape derivative (VI.1.4) vanishes, which implies

$$\mathbb{E}[u_\varepsilon](\mathbf{x}) = u_0(\mathbf{x}) + \frac{\varepsilon^2}{2} \mathbb{E}[\delta^2 u](\mathbf{x}) + \mathcal{O}(\varepsilon^3), \quad \mathbf{x} \in K \Subset D_\varepsilon^{\cap \Omega}. \quad (\text{VI.1.8})$$

By tensorizing (VI.1.3) and integrating over Ω , we further obtain

$$\begin{aligned} \text{Cor}[u_\varepsilon](\mathbf{x}, \mathbf{y}) &= u_0(\mathbf{x})u_0(\mathbf{y}) + \varepsilon^2 \text{Cor}[\delta u](\mathbf{x}, \mathbf{y}) \\ &+ \frac{\varepsilon^2}{2} \left(u_0(\mathbf{x})\mathbb{E}[\delta^2 u](\mathbf{y}) + \mathbb{E}[\delta^2 u](\mathbf{x})u_0(\mathbf{y}) \right) + \mathcal{O}(\varepsilon^3), \quad \mathbf{x}, \mathbf{y} \in K \Subset D_\varepsilon^{\cap \Omega}. \end{aligned} \quad (\text{VI.1.9})$$

Finally, subtracting (VI.1.8) from both sides in (VI.1.3), tensorizing and integrating over Ω yields

$$\text{Cov}[u_\varepsilon](\mathbf{x}, \mathbf{y}) = \varepsilon^2 \text{Cor}[\delta u](\mathbf{x}, \mathbf{y}) + \mathcal{O}(\varepsilon^3), \quad \mathbf{x}, \mathbf{y} \in K \Subset D_\varepsilon^{\cap \Omega}. \quad (\text{VI.1.10})$$

In order to compute the quantities $\mathbb{E}[u_\varepsilon]$, $\text{Cor}[u_\varepsilon]$, and $\text{Cov}[u_\varepsilon]$, appearing in (VI.1.8), (VI.1.9), and (VI.1.10), we have to solve for $\mathbb{E}[\delta^2 u]$ and $\text{Cor}[\delta u]$. To this end, we combine (VI.1.5) and (VI.1.7) to arrive at, see also [HSS08b],

$$\begin{aligned} \Delta \mathbb{E}[\delta^2 u[\mathbf{V}, \mathbf{V}]] &= 0 && \text{in } D_0, \\ \mathbb{E}[\delta^2 u[\mathbf{V}, \mathbf{V}]] &= \left(\Delta g - 2\mathcal{H} \frac{\partial(g - u_0)}{\partial \mathbf{n}} \right) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \Big|_{\mathbf{x}=\mathbf{y}} - 2\mathbb{E} \left[\langle \mathbf{V}, \mathbf{n} \rangle \frac{\partial \delta u[\mathbf{V}]}{\partial \mathbf{n}} \right] && \text{on } \partial D_0. \end{aligned} \quad (\text{VI.1.11})$$

Next, tensorizing (VI.1.4) and integrating over Ω yields

$$\begin{aligned} (\Delta \otimes \Delta) \text{Cor}[\delta u] &= 0 && \text{in } D_0 \times D_0, \\ (\Delta \otimes \text{Id}) \text{Cor}[\delta u] &= 0 && \text{on } D_0 \times \partial D_0, \\ (\text{Id} \otimes \Delta) \text{Cor}[\delta u] &= 0 && \text{on } \partial D_0 \times D_0, \\ \text{Cor}[\delta u] &= \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \left(\frac{\partial(g - u_0)}{\partial \mathbf{n}} \otimes \frac{\partial(g - u_0)}{\partial \mathbf{n}} \right) && \text{on } \partial D_0 \times \partial D_0. \end{aligned} \quad (\text{VI.1.12})$$

The asymptotic expansions for $\mathbb{E}[u_\varepsilon]$, $\text{Cor}[u_\varepsilon]$, and $\text{Cov}[u_\varepsilon]$ can, under certain circumstances, be improved to fourth order accuracy with help of the following lemma, which is inspired by [HP15, Lemma 2.3].

Lemma VI.1.1. *Assume that the boundary perturbations in normal direction are given by an expansion*

$$\langle \mathbf{V}(\omega, \mathbf{x}), \mathbf{n} \rangle = \sum_{i=1}^M \kappa_i(\mathbf{x}) X_i(\omega),$$

where X_i , $i = 1, \dots, M$, are independent and identically distributed random variables. Then, it holds

$$\begin{aligned} \delta u[\mathbf{V}(\omega)] &= \sum_{i=1}^M \delta u[\kappa_i \cdot \mathbf{n}] X_i(\omega), \\ \delta^2 u[\mathbf{V}(\omega), \mathbf{V}(\omega)] &= \sum_{i,j=1}^M \delta^2 u[\kappa_i \cdot \mathbf{n}, \kappa_j \cdot \mathbf{n}] X_i(\omega) X_j(\omega), \end{aligned}$$

and

$$\delta^3 u[\mathbf{V}(\omega), \mathbf{V}(\omega), \mathbf{V}(\omega)] = \sum_{i,j,k=1}^M \delta^3 u[\kappa_i \cdot \mathbf{n}, \kappa_j \cdot \mathbf{n}, \kappa_k \cdot \mathbf{n}] X_i(\omega) X_j(\omega) X_k(\omega),$$

provided that the third order local shape derivative $\delta^3 u$, as usual given as the local shape derivative of the second order local shape derivative, exists.

Proof. The first two expressions were already provided in [HP15, Lemma 2.3], whereas the third one is analogously derived by exploiting the trilinearity of $\delta^3 u$. \square

Obviously, due to the independence of the random variables $(X_i)_{i=1}^M$, it holds

$$\mathbb{E}[\delta^3 u] = \sum_{i=1}^M \delta^3 u[\kappa_i \cdot \mathbf{n}, \kappa_i \cdot \mathbf{n}, \kappa_i \cdot \mathbf{n}] \mathbb{E}[X_i^3] = 0,$$

if the probability distribution of the X_i is symmetric around zero. The expansion of the mean $\mathbb{E}[u_\varepsilon]$ (VI.1.8) is thus fourth order accurate. Similarly, it holds under the same assumptions

$$\text{Cor}[\delta^2 u, \delta u] = \sum_{i=1}^M \delta^2 u[\kappa_i \cdot \mathbf{n}, \kappa_i \cdot \mathbf{n}] \delta u_0[\kappa_i \cdot \mathbf{n}] \mathbb{E}[X_i^3] = 0.$$

Hence, the expansion for $\text{Cor}[u_\varepsilon]$ (VI.1.9) becomes fourth order accurate, and, likewise, the expansion for $\text{Cov}[u_\varepsilon]$ (VI.1.10) also becomes fourth order accurate.

VI.2 Boundary Integral Equations

We shall use boundary integral equations to compute the asymptotic expansions (VI.1.8), (VI.1.9), and (VI.1.10) for the statistics of the random solution. To this end, observe that the boundary conditions for the PDEs of $\mathbb{E}[\delta^2 u]$ and $\text{Cor}[\delta u]$, (VI.1.11) and (VI.1.12), depend on the Neumann data $t^{(u_0)} = \partial u_0 / \partial \mathbf{n}$ of the solution u_0 of the unperturbed boundary value problem (VI.1.1). Having the Dirichlet data g of u_0 at hand, the corresponding Neumann data are given by the Dirichlet-to-Neumann map \mathcal{S} from (IV.1.13).

To obtain the second term of the Dirichlet data of $\mathbb{E}[\delta^2 u]$ in (VI.1.11), we employ that it can be rewritten as the diagonal of a correlation, call it A ,

$$\mathbb{E} \left[\langle \mathbf{V}, \mathbf{n} \rangle \frac{\partial \delta u[\mathbf{V}]}{\partial \mathbf{n}} \right] = \mathbb{E} \left[\langle \mathbf{V}(\cdot, \mathbf{x}), \mathbf{n}_{\mathbf{x}} \rangle \frac{\partial \delta u[\mathbf{V}(\cdot, \mathbf{y})]}{\partial \mathbf{n}_{\mathbf{y}}} \right] \Big|_{\mathbf{x}=\mathbf{y}} =: A(\mathbf{x}, \mathbf{y}) \Big|_{\mathbf{x}=\mathbf{y}}. \quad (\text{VI.2.1})$$

The Dirichlet-to-Neumann map and the definition of the first order local shape derivative (VI.1.4) then yield

$$\begin{aligned} A(\mathbf{x}, \mathbf{y}) &:= \mathbb{E} \left[\langle \mathbf{V}(\cdot, \mathbf{x}), \mathbf{n}_{\mathbf{x}} \rangle \frac{\partial \delta u[\mathbf{V}(\cdot, \mathbf{y})]}{\partial \mathbf{n}_{\mathbf{y}}} \right] \\ &= \mathbb{E} \left[\langle \mathbf{V}(\cdot, \mathbf{x}), \mathbf{n}_{\mathbf{x}} \rangle \mathcal{S} \left(\langle \mathbf{V}(\cdot, \mathbf{y}), \mathbf{n}_{\mathbf{y}} \rangle \left(\frac{\partial g}{\partial \mathbf{n}_{\mathbf{y}}} - t^{(u_0)} \right) \right) \right] \\ &= (\text{Id} \otimes \mathcal{S}) \left(\mathbb{E} \left[\langle \mathbf{V}(\cdot, \mathbf{x}), \mathbf{n}_{\mathbf{x}} \rangle \langle \mathbf{V}(\cdot, \mathbf{y}), \mathbf{n}_{\mathbf{y}} \rangle \right] \left(\mathbb{1} \otimes \left(\frac{\partial g}{\partial \mathbf{n}_{\mathbf{y}}} - t^{(u_0)} \right) \right) \right) \\ &= (\text{Id} \otimes \mathcal{S}) \left(\left(\mathbb{1} \otimes \left(\frac{\partial g}{\partial \mathbf{n}_{\mathbf{y}}} - t^{(u_0)} \right) \right) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \right). \end{aligned} \quad (\text{VI.2.2})$$

Given the Dirichlet data

$$g^{(\mathbb{E}[\delta^2 u])} = \left(\Delta g - 2\mathcal{H} \frac{\partial(g - u_0)}{\partial \mathbf{n}} \right) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \Big|_{\mathbf{x}=\mathbf{y}} - 2A \Big|_{\mathbf{x}=\mathbf{y}} \quad (\text{VI.2.3})$$

and the Neumann data

$$t^{\mathbb{E}[\delta^2 u]} = \mathcal{S}g^{\mathbb{E}[\delta^2 u]}$$

of $\mathbb{E}[\delta^2 u]$, the solution $\mathbb{E}[\delta^2 u]$ of (VI.1.11) can be represented inside the domain by using the representation formula (IV.1.2), which yields

$$\mathbb{E}[\delta^2 u](\mathbf{x}) = \tilde{\mathcal{V}}\left(t^{\mathbb{E}[\delta^2 u]}\right)(\mathbf{x}) - \tilde{\mathcal{K}}\left(g^{\mathbb{E}[\delta^2 u]}\right)(\mathbf{x}), \quad \mathbf{x} \in D_{\cap\Omega}. \quad (\text{VI.2.4})$$

Similarly, given the Dirichlet data of $\text{Cor}[\delta u]$,

$$g^{\text{Cor}[\delta u]} = \left(\frac{\partial(g - u_0)}{\partial \mathbf{n}} \otimes \frac{\partial(g - u_0)}{\partial \mathbf{n}} \right) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle],$$

and solving

$$(\mathcal{V} \otimes \mathcal{V})\rho^{\text{Cor}[\delta u]} = g^{\text{Cor}[\delta u]}, \quad (\text{VI.2.5})$$

the second order correction term $\text{Cor}[\delta u]$ can be represented as solution of (VI.1.12) by the representation formula

$$\text{Cor}[\delta u](\mathbf{x}, \mathbf{y}) = (\tilde{\mathcal{V}} \otimes \tilde{\mathcal{V}})\left(\rho^{\text{Cor}[\delta u]}\right)(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in D_{\cap\Omega}. \quad (\text{VI.2.6})$$

For the computation of $\text{Cor}[\delta u]$ we are thus in a similar setting as in the previous chapter.

VI.3 Galerkin Discretization

For the following discussion of the Galerkin discretizations, let V_h be the N -dimensional space of piecewise polynomial ansatz functions of order d on ∂D_0 , generated from a quasi-uniform mesh on ∂D_0 . Since the Neumann data $t^{(u_0)}$ of the solution u_0 of the unperturbed boundary value problem (VI.1.1) are needed for the computation of both of the correction terms $\mathbb{E}[\delta^2 u]$ and $\text{Cor}[\delta u]$, we will first consider the discretization of the Dirichlet-to-Neumann map (IV.1.13). Although the discretization is pretty standard in boundary element methods, cf., e.g., [Ste08], we believe its repetition is useful to establish notation and will help the reader to understand the following more involved steps.

VI.3.1 Dirichlet-to-Neumann Map

The variational formulation of the Dirichlet-to-Neumann map is given as follows:

Find $t^{(u_0)} \in H^{-1/2}(\partial D_0)$ such that

$$(\mathcal{V}t^{(u_0)}, v)_{L^2(\partial D_0)} = \left(\left(\frac{1}{2} + \mathcal{K} \right) g, v \right)_{L^2(\partial D_0)}$$

for all $v \in H^{-1/2}(\partial D_0)$.

Replacing the energy space $H^{-1/2}(\partial D_0)$ by the finite dimensional space $V_h \subset H^{-1/2}(\partial D_0)$, choosing a basis $(\varphi_i)_{i=1}^N$ of V_h , and replacing g with its L^2 -projection $\Pi_h g$ onto V_h , we end up with the system of linear equations

$$\mathbf{V}\mathbf{t}^{(u_0)} = \left(\frac{1}{2}\mathbf{M} + \mathbf{K} \right) \mathbf{M}^{-1}\mathbf{g}. \quad (\text{VI.3.1})$$

The corresponding system matrices are given by

$$\mathbf{V} = [(\mathcal{V}\varphi_j, \varphi_i)_{L^2(\partial D_0)}]_{i,j=1}^N, \quad \mathbf{K} = [(\mathcal{K}\varphi_j, \varphi_i)_{L^2(\partial D_0)}]_{i,j=1}^N, \quad \mathbf{M} = [(\varphi_j, \varphi_i)_{L^2(\partial D_0)}]_{i,j=1}^N,$$

while the vectors are given by

$$\mathbf{g} = \left[(g, \varphi_i)_{L^2(\partial D_0)} \right]_{i=1}^N, \quad \mathbf{t}^{(u_0)} = \left[t_i^{(u_0)} \right]_{i=1}^N,$$

where the coefficients $t_i^{(u_0)}$ are the coefficients of the basis expansion

$$t_h^{(u_0)} = \sum_{i=1}^N t_i^{(u_0)} \varphi_i.$$

VI.3.2 Computation of $\text{Cor}[\delta u]$

For determining $\text{Cor}[\delta u]$ via the representation formula (VI.2.6), we have to compute $\rho^{(\text{Cor}[\delta u])}$, being the solution of the tensor equation (VI.2.5), which is a special case of (II.4.4).

The variational formulation of (VI.2.5) is then given as follows:

$$\begin{aligned} & \text{Find } \rho^{(\text{Cor}[\delta u])} \in H_{\text{mix}}^{-1/2, -1/2}(\partial D_0 \times \partial D_0) \text{ such that} \\ & \left((\mathcal{V} \otimes \mathcal{V}) \rho^{(\text{Cor}[\delta u])}, v \right)_{L^2(\partial D_0 \times \partial D_0)} = \left(g^{(\text{Cor}[\delta u])}, v \right)_{L^2(\partial D_0 \times \partial D_0)} \\ & \text{for all } v \in H_{\text{mix}}^{-1/2, -1/2}(\partial D_0 \times \partial D_0). \end{aligned}$$

Replacing the energy space $H_{\text{mix}}^{-1/2, -1/2}(\partial D_0 \times \partial D_0)$ by the finite dimensional ansatz space $V_h \otimes V_h \subset H_{\text{mix}}^{-1/2, -1/2}(\partial D_0 \times \partial D_0)$, choosing a basis $(\varphi_i \otimes \varphi_j)_{i,j=1}^N$ of $V_h \otimes V_h$, and replacing $g^{(\text{Cor}[\delta u])}$ by the approximation

$$g_h^{(\text{Cor}[\delta u])} := \left(\left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right) \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right) \right) \left((\Pi_h \otimes \Pi_h) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \right),$$

we end up with the system of linear equations

$$(\mathbf{V} \otimes \mathbf{V}) \text{vec}(\mathbf{C}_{\rho, \text{Cor}[\delta u]}) = (\mathbf{N} \otimes \mathbf{N})(\mathbf{M} \otimes \mathbf{M})^{-1} \text{vec}(\mathbf{C}_{\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]}). \quad (\text{VI.3.2})$$

The corresponding system matrices are given by

$$\begin{aligned} \mathbf{N} &= \left[\left[\left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right) \varphi_j, \varphi_i \right]_{L^2(\partial D_0)} \right]_{i,j=1}^N, \\ \mathbf{C}_{\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]} &= \left[\left(\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle], \varphi_i \otimes \varphi_j \right)_{L^2(\partial D_0 \times \partial D_0)} \right]_{i,j=1}^N, \\ \mathbf{C}_{\rho, \text{Cor}[\delta u]} &= \left[c_{ij}^{\rho, \text{Cor}[\delta u]} \right]_{i,j=1}^N, \end{aligned}$$

where the coefficients $c_{ij}^{\rho, \text{Cor}[\delta u]}$ are the coefficients of the basis expansion

$$\rho_h^{(\text{Cor}[\delta u])} = \sum_{i,j=1}^N c_{ij}^{\rho, \text{Cor}[\delta u]} (\varphi_i \otimes \varphi_j).$$

Following the considerations at the end of Chapter II.4, we may rewrite (VI.3.2) as

$$\mathbf{V} \mathbf{C}_{\rho, \text{Cor}[\delta u]} \mathbf{V}^\top = \mathbf{N} \mathbf{M}^{-1} \mathbf{C}_{\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]} \mathbf{M}^{-\top} \mathbf{N}^\top. \quad (\text{VI.3.3})$$

VI.3.3 Computation of $\mathbb{E}[\delta^2 u]$

For the computation of $\mathbb{E}[\delta^2 u]$, we shall first consider the numerical treatment of the computation of the correlation A as given by (VI.2.2). Its variational formulation is:

Find $A \in H_{\text{mix}}^{0,-1/2}(\partial D_0 \times \partial D_0)$ such that

$$\begin{aligned} \left((\text{Id} \otimes \mathcal{V})A, v \right)_{L^2(\partial D_0 \times \partial D_0)} = \\ \left(\left(\text{Id} \otimes \left(\frac{1}{2} + \mathcal{K} \right) \right) \left(\mathbb{1} \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)} \right) \right) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle], v \right)_{L^2(\partial D_0 \times \partial D_0)} \end{aligned}$$

for all $v \in H_{\text{mix}}^{0,-1/2}(\partial D_0 \times \partial D_0)$.

Choosing a basis $(\varphi_i \otimes \varphi_j)_{i,j=1}^N$ of $V_h \otimes V_h \subset H_{\text{mix}}^{0,-1/2}(\partial D_0 \times \partial D_0)$ and employing L^2 -projections leads, as in the previous section, to a finite dimensional variational problem. It corresponds to the system of linear equations

$$(\mathbf{M} \otimes \mathbf{V}) \text{vec}(\mathbf{A}) = \left(\mathbf{M} \otimes \left(\frac{1}{2} \mathbf{M} + \mathbf{K} \right) \right) (\mathbf{M} \otimes \mathbf{M})^{-1} (\mathbf{M} \otimes \mathbf{N}) (\mathbf{M} \otimes \mathbf{M})^{-1} \text{vec}(\mathbf{C}_{\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]}),$$

where \mathbf{A} contains the coefficients of the basis expansion of the Galerkin solution $A_h \in V_h \otimes V_h$. Rearranging this system as before yields the matrix equation

$$\mathbf{V} \mathbf{A} \mathbf{M}^\top = \left(\frac{1}{2} \mathbf{M} + \mathbf{K} \right) \mathbf{M}^{-1} \mathbf{N} \mathbf{M}^{-1} \mathbf{C}_{\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]}. \quad (\text{VI.3.4})$$

Having approximated the correlation A , the Dirichlet data of $\mathbb{E}[\delta^2 u]$ can now be computed. Following the considerations of the discretization of the Dirichlet-to-Neumann map from Chapter VI.3.1 and employing L^2 -projections, the Neumann data $t^{(\mathbb{E}[\delta^2 u])}$ of $\mathbb{E}[\delta^2 u]$ are given by

$$\mathbf{v}_t^{(\mathbb{E}[\delta^2 u])} = \left(\frac{1}{2} \mathbf{M} + \mathbf{K} \right) \mathbf{M}^{-1} \left((\mathbf{G} - \mathbf{L} \mathbf{M}^{-1} \mathbf{N}) \mathbf{M}^{-1} \mathbf{d} - 2 \mathbf{b} \right) \quad (\text{VI.3.5})$$

with corresponding vectors and matrices

$$\begin{aligned} \mathbf{d} &= \left[(\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] |_{\mathbf{x}=\mathbf{y}}, \varphi_i)_{L^2(\partial D_0)} \right]_{i=1}^N, & \mathbf{b} &= \left[(A_h |_{\mathbf{x}=\mathbf{y}}, \varphi_i)_{L^2(\partial D_0)} \right]_{i=1}^N, \\ \mathbf{L} &= \left[(2\mathcal{H}\varphi_j, \varphi_i)_{L^2(\partial D_0)} \right]_{i,j=1}^N, & \mathbf{G} &= \left[(\Delta g \varphi_j, \varphi_i)_{L^2(\partial D_0)} \right]_{i,j=1}^N. \end{aligned}$$

The correction term $\mathbb{E}[\delta^2 u]$ itself is then given by the representation formula (VI.2.4).

The error estimation of the approximate solutions is the topic of the next section.

VI.4 Error Estimates

For the following error estimates, it is important to carefully distinguish between the regularity requirements of the involved shape calculus and the regularity assumptions for the Galerkin discretization. More specifically, let us remark that a regularity assumption on the boundary ∂D_0 of the reference domain does not necessarily imply the same regularity to the perturbations and the perturbed domains $D(\omega)$. Throughout our proofs, we require that the domain D_0 is of class $C^{d,1}$ and that it holds $g \in H^{d+1}(\partial D_0)$ for the Dirichlet data on the reference domain.

VI.4.1 Preliminaries

We start by restating a convergence result for the Dirichlet-to-Neumann map.

Lemma VI.4.1. *For the Neumann data $t^{(u_0)}$ of the solution u_0 of the unperturbed problem (VI.1.1) and their approximation $t_h^{(u_0)}$, it holds for $-d \leq s \leq 0$ that*

$$\left\| t^{(u_0)} - t_h^{(u_0)} \right\|_{H^s(\partial D_0)} \lesssim h^{d-s}.$$

Proof. The proof for the case $-d \leq s \leq -1/2$ is standard, we refer to [Ste08, Theorem 12.7] and remark that $\mathcal{K}: H^s(\partial D_0) \rightarrow H^{s+1}(\partial D_0)$ is a continuous operator on $C^{d,1}$ -boundaries. To extend the result to the case $-1/2 < s \leq 0$, consider the inequality

$$\left\| t^{(u_0)} - t_h^{(u_0)} \right\|_{H^s(\partial D_0)} \leq \left\| t^{(u_0)} - \Pi_h t^{(u_0)} \right\|_{H^s(\partial D_0)} + \left\| \Pi_h t^{(u_0)} - t_h^{(u_0)} \right\|_{H^s(\partial D_0)}.$$

The first term on the right hand side can again be estimated by the standard approximation property of the L^2 -projection, whereas, for the second term on the right hand side, we employ the inverse estimate to obtain

$$\begin{aligned} \left\| \Pi_h t^{(u_0)} - t_h^{(u_0)} \right\|_{H^s(\partial D_0)} &\leq h^{s-1/2} \left\| \Pi_h t^{(u_0)} - t_h^{(u_0)} \right\|_{H^{-1/2}(\partial D_0)} \\ &\leq h^{s-1/2} \left(\left\| \Pi_h t^{(u_0)} - t^{(u_0)} \right\|_{H^{-1/2}(\partial D_0)} + \left\| t^{(u_0)} - t_h^{(u_0)} \right\|_{H^{-1/2}(\partial D_0)} \right). \end{aligned}$$

This yields the assertion together with the first part of the lemma and the standard approximation estimates of the L^2 -projection. \square

The following technical lemma is needed for the error estimation and is inspired in parts by the proof of [HSS08b, Theorem 7.3].

Lemma VI.4.2. *Assume that $a, b \in H^d(\partial D_0)$, $\|a - a_h\|_{L^2(\partial D_0)} \lesssim h^d$, and $\|a - a_h\|_{H^s(\partial D_0)} \lesssim h^{d-s}$ for $-d \leq s < -(n-1)/2$. Then, we have*

$$\left\| ab - a_h \Pi_h b \right\|_{H^s(\partial D_0)} \lesssim h^{d-s}.$$

Proof. We start by splitting the error into two parts

$$\left\| ab - a_h \Pi_h b \right\|_{H^s(\partial D_0)} \leq \left\| (a - a_h)b \right\|_{H^s(\partial D_0)} + \left\| a_h(\text{Id} - \Pi_h)b \right\|_{H^s(\partial D_0)}.$$

Let $u, v \in H^{-s}(\partial D_0)$. From the estimate $\|uv\|_{H^{-s}(\partial D_0)} \lesssim \|u\|_{H^{-s}(\partial D_0)} \|v\|_{H^{-s}(\partial D_0)}$, cf. [BH15, Zol77], we conclude by duality

$$\begin{aligned} \|uv\|_{H^s(\partial D_0)} &= \sup_{\|w\|_{H^{-s}(\partial D_0)}=1} (uv, w)_{L^2(\partial D_0)} \\ &= \sup_{\|w\|_{H^{-s}(\partial D_0)}=1} (u, vw)_{L^2(\partial D_0)} \\ &\leq \|u\|_{H^s(\partial D_0)} \sup_{\|w\|_{H^{-s}(\partial D_0)}=1} \|vw\|_{H^{-s}(\partial D_0)} \\ &\lesssim \|u\|_{H^s(\partial D_0)} \|v\|_{H^{-s}(\partial D_0)}. \end{aligned}$$

Thus, the first part of the error can easily be estimated by

$$\left\| (a - a_h)b \right\|_{H^s(\partial D_0)} \lesssim \|a - a_h\|_{H^s(\partial D_0)} \|b\|_{H^{-s}(\partial D_0)},$$

whereas the second part of the error is treated in accordance with

$$\begin{aligned} &\left\| a_h(\text{Id} - \Pi_h)b \right\|_{H^s(\partial D_0)} \\ &= \sup_{\|w\|_{H^{-s}(\partial D_0)}=1} (a_h(\text{Id} - \Pi_h)b, w)_{L^2(\partial D_0)} \\ &= \sup_{\|w\|_{H^{-s}(\partial D_0)}=1} \left(\left((a_h - a)(\text{Id} - \Pi_h)b, w \right)_{L^2(\partial D_0)} + (a(\text{Id} - \Pi_h)b, w)_{L^2(\partial D_0)} \right). \end{aligned}$$

Since $H^{-s}(\partial D_0)$ is continuously embedded into $L^\infty(\partial D_0)$ on $C^{d,1}$ -smooth boundaries for $-s > (n-1)/2$, one can estimate

$$\begin{aligned} \left((a_h - a)(\text{Id} - \Pi_h)b, w \right)_{L^2(\partial D_0)} &\leq \left\| (a_h - a)(\text{Id} - \Pi_h)b \right\|_{L^1(\partial D_0)} \|w\|_{L^\infty(\partial D_0)} \\ &\lesssim \|a - a_h\|_{L^2(\partial D_0)} \left\| (\text{Id} - \Pi_h)b \right\|_{L^2(\partial D_0)} \|w\|_{H^{-s}(\partial D_0)}, \end{aligned}$$

and, since it holds $(\text{Id} - \Pi_h)^*(\text{Id} - \Pi_h) = (\text{Id} - \Pi_h)$ due to the properties of the L^2 -projection,

$$\begin{aligned} (a(\text{Id} - \Pi_h)b, w)_{L^2(\partial D_0)} &= ((\text{Id} - \Pi_h)b, aw)_{L^2(\partial D_0)} \\ &= ((\text{Id} - \Pi_h)b, (\text{Id} - \Pi_h)(aw))_{L^2(\partial D_0)} \\ &\leq \left\| (\text{Id} - \Pi_h)b \right\|_{L^2(\partial D_0)} \left\| (\text{Id} - \Pi_h)(aw) \right\|_{L^2(\partial D_0)}. \end{aligned}$$

The assertion follows by applying the standard approximation estimates of the L^2 -projection, i.e., it holds

$$\left\| (\text{Id} - \Pi_h)w \right\|_{H^{-\tilde{t}}(\partial D_0)} \lesssim h^{t+\tilde{t}} \|w\|_{H^t(\partial D_0)} \quad (\text{VI.4.1})$$

for $0 \leq \tilde{t} \leq t \leq d$ and $w \in H^t(\partial D_0)$, see, e.g., [SS11, Chapter 4]. Thus, the first part of product is bounded by $\lesssim h^d$ and the second part of product is bounded by $\lesssim h^{-s}$. \square

By a tensor product argument, we arrive at a similar statement on the tensor product domain.

Corollary VI.4.3. *Assume that $a, b \in H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$, $\|a - a_h\|_{L^2(\partial D_0 \times \partial D_0)} \lesssim h^d$, and $\|a - a_h\|_{H_{\text{mix}}^{s,s}(\partial D_0 \times \partial D_0)} \lesssim h^{d-s}$ for $-d \leq s < -(n-1)/2$. Then, it holds that*

$$\left\| ab - a_h(\Pi_h \otimes \Pi_h)b \right\|_{H_{\text{mix}}^{s,s}(\partial D_0 \times \partial D_0)} \lesssim h^{d-s}.$$

Proof. The proof is very similar to the proof of the preceding lemma. We refer to the proof of [HSS08b, Theorem 7.3], where the details are given for the case $s = -d$, using the estimate $\|uv\|_{H^d(\partial D_0)} \leq \|u\|_{C^{d-1,1}(\partial D_0)} \|v\|_{H^d(\partial D_0)}$ instead of $\|uv\|_{H^d(\partial D_0)} \lesssim \|u\|_{H^d(\partial D_0)} \|v\|_{H^d(\partial D_0)}$. \square

Remark VI.4.4. *The focus of the present chapter is on the cases $n = 2, 3$ such that $(n-1)/2 < d$ implies $d \geq 1$ for $n = 2$ and $d \geq 2$ for $n = 3$, respectively. To simplify the presentation of the following results, we will therefore restrict ourselves to the case $d \geq 2$, i.e., we are considering discretizations with at least piecewise linear continuous ansatz functions.*

VI.4.2 Approximation Error of $\text{Cor}[\delta u]$

In order to bound the approximation error of $\text{Cor}[\delta u]$ given by the representation formula (VI.2.6), we first have to bound the approximation error of its boundary values. This is in parts due to the following lemma.

Lemma VI.4.5. *Let $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \in H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$. Then, it holds*

$$\left\| g^{(\text{Cor}[\delta u])} - g_h^{(\text{Cor}[\delta u])} \right\|_{H_{\text{mix}}^{-d,-d}(\partial D_0 \times \partial D_0)} \lesssim h^{2d}.$$

Proof. The assertion is an immediate consequence of Corollary VI.4.3 with

$$a = \left(\frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)} \right) \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)} \right), \quad a_h = \left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right) \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right),$$

and $b = \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]$. The required convergence in the $H^{-d}(\partial D_0)$ -norm and in the $L^2(\partial D_0)$ -norm is proven in Lemma VI.4.1 for the non-tensor product case. For the tensor product case, consider for $-d \leq s \leq 0$ that

$$\begin{aligned} & \|a - a_h\|_{H_{\text{mix}}^{s,s}(\partial D_0 \times \partial D_0)} \\ &= \left\| \left(\frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)} \right) \otimes \left(t_h^{(u_0)} - t^{(u_0)} \right) + \left(t_h^{(u_0)} - t^{(u_0)} \right) \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right) \right\|_{H_{\text{mix}}^{s,s}(\partial D_0 \times \partial D_0)} \\ &\leq \left\| \frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)} \right\|_{H^s(\partial D_0)} \left\| t^{(u_0)} - t_h^{(u_0)} \right\|_{H^s(\partial D_0)} \\ &\quad + \left\| t^{(u_0)} - t_h^{(u_0)} \right\|_{H^s(\partial D_0)} \left\| \frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right\|_{H^s(\partial D_0)} \\ &\lesssim h^{d-s}. \end{aligned}$$

This completes the proof. \square

We are finally in the position to estimate the error of $\text{Cor}[\delta u]_h$.

Lemma VI.4.6. *Let $\text{Cor}[\delta u]$ be given as in (VI.2.6) and let $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \in H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$. It then holds*

$$|\text{Cor}[\delta u](\mathbf{x}, \mathbf{y}) - \text{Cor}[\delta u]_h(\mathbf{x}, \mathbf{y})| \lesssim h^{2d} \quad \text{for all } \mathbf{x}, \mathbf{y} \in K \Subset D_{\cap \Omega}.$$

Proof. By employing the representation formula (VI.2.6), we obtain

$$\begin{aligned} & |\text{Cor}[\delta u](\mathbf{x}, \mathbf{y}) - \text{Cor}[\delta u]_h(\mathbf{x}, \mathbf{y})| \\ &= \left| (\tilde{\mathcal{V}} \otimes \tilde{\mathcal{V}}) \left(\rho^{(\text{Cor}[\delta u])} - \rho_h^{(\text{Cor}[\delta u])} \right) (\mathbf{x}, \mathbf{y}) \right| \\ &\lesssim \left\| \frac{1}{16\pi^2 \|\mathbf{x} - \cdot\| \otimes \|\mathbf{y} - \cdot\|} \right\|_{H_{\text{mix}}^{d+1,d+1}(\partial D_0 \times \partial D_0)} \cdot \left\| \rho^{(\text{Cor}[\delta u])} - \rho_h^{(\text{Cor}[\delta u])} \right\|_{H_{\text{mix}}^{-d-1,-d-1}(\partial D_0 \times \partial D_0)}. \end{aligned}$$

Using Strang's first lemma and Lemma VI.4.5, we further derive

$$\left\| \rho^{(\text{Cor}[\delta u])} - \rho_h^{(\text{Cor}[\delta u])} \right\|_{H_{\text{mix}}^{-d-1,-d-1}(\partial D_0 \times \partial D_0)} \lesssim h^{2d},$$

which, in view of $K \Subset \partial D_0$ and thus $\text{dist}(K, \partial D_0) > 0$, implies the assertion. \square

Remark VI.4.7. *The assumptions of the presented result are slightly weaker than the related result from [HSS08b]. Whereas [HSS08b] requires $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]$ to be in $C^{d-1,1}(\partial D_0) \otimes C^{d-1,1}(\partial D_0)$, we only require it to be in $H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$. This means that we do not need any extra regularity of the boundary perturbation in addition to the regularity required for the first order shape derivative and the Galerkin approximation of $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]$.*

VI.4.3 Approximation Error of $\mathbb{E}[\delta^2 u]$

In order to estimate the discretization error of $\mathbb{E}[\delta^2 u]$ given by (VI.1.11), we need to know the discretization error of its Dirichlet data. Especially, we need to estimate the error of the correlation A as given by (VI.2.2).

Lemma VI.4.8. *Let $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \in H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$. It then holds for*

$$A_h = (\text{Id} \otimes \mathcal{V}^{-1}) \left(\text{Id} \otimes \left(\frac{1}{2} + \mathcal{K}\Pi_h \right) \right) \left(\left(\mathbb{1} \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right) \right) (\Pi_h \otimes \Pi_h) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \right)$$

that

$$\|A - A_h\|_{H_{\text{mix}}^{-d+1,-d+1}(\partial D_0 \times \partial D_0)} \lesssim h^{2(d-1)}.$$

Proof. For

$$a = \mathbb{1} \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)} \right), \quad a_h = \mathbb{1} \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right), \quad b = \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle],$$

it holds

$$\begin{aligned} & \|A - A_h\|_{H_{\text{mix}}^{-d+1, -d+1}(\partial D_0 \times \partial D_0)} \\ & \leq \| \text{Id} \otimes \mathcal{V}^{-1} \|_{H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0) \rightarrow H_{\text{mix}}^{-d+1, -d+1}(\partial D_0 \times \partial D_0)} \\ & \quad \left\| \left(\text{Id} \otimes \left(\frac{1}{2} + \mathcal{K} \right) \right) (ab) - \left(\text{Id} \otimes \left(\frac{1}{2} + \mathcal{K} \Pi_h \right) \right) (a_h (\Pi_h \otimes \Pi_h) b) \right\|_{H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0)} \\ & \lesssim \| \text{Id} \otimes \mathcal{K} \|_{H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0) \rightarrow H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0)} \\ & \quad \left\| ab - (\text{Id} \otimes \Pi_h)(ab) \right\|_{H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0)} \\ & \quad + \left\| \text{Id} \otimes \left(\frac{1}{2} + \mathcal{K} \Pi_h \right) \right\|_{H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0) \rightarrow H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0)} \\ & \quad \left\| ab - a_h (\Pi_h \otimes \Pi_h) b \right\|_{H_{\text{mix}}^{-d+2, -d+2}(\partial D_0 \times \partial D_0)}. \end{aligned}$$

The assertion now follows from the approximation property of the L^2 -projection (VI.4.1), Lemma VI.4.1, and Corollary VI.4.3. \square

The previous lemma allows us to bound the error of the Dirichlet data of the boundary value problem (VI.1.11) for $\mathbb{E}[\delta^2 u]$.

Lemma VI.4.9. *Let $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]|_{\mathbf{x}=\mathbf{y}} \in H^d(\partial D_0)$ and $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \in H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$. For the Dirichlet data of $\mathbb{E}[\delta^2 u]$ and their numerical approximation*

$$g_h^{(\mathbb{E}[\delta^2 u])} = \left(\Delta g - 2\mathcal{H} \Pi_h \left(\frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)} \right) \right) \Pi_h \left(\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]|_{\mathbf{x}=\mathbf{y}} \right) - 2A_h|_{\mathbf{x}=\mathbf{y}},$$

it holds

$$\left\| g^{(\mathbb{E}[\delta^2 u])} - g_h^{(\mathbb{E}[\delta^2 u])} \right\|_{H^{-d}(\partial D_0)} \lesssim h^{2(d-1)} \quad (\text{VI.4.2})$$

and

$$\left\| g^{(\mathbb{E}[\delta^2 u])} - \Pi_h g_h^{(\mathbb{E}[\delta^2 u])} \right\|_{H^{-d}(\partial D_0)} \lesssim h^{2(d-1)}. \quad (\text{VI.4.3})$$

Proof. By exploiting $\Pi_h t_h^{(u_0)} = t_h^{(u_0)}$ and setting

$$a = \frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)}, \quad a_h = \Pi_h \frac{\partial g}{\partial \mathbf{n}} - t_h^{(u_0)}, \quad b = \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]|_{\mathbf{x}=\mathbf{y}},$$

we conclude

$$\begin{aligned} \left\| g^{(\mathbb{E}[\delta^2 u])} - g_h^{(\mathbb{E}[\delta^2 u])} \right\|_{H^{-d}(\partial D_0)} & \leq \| (\Delta g)(b - \Pi_h b) \|_{H^{-d}(\partial D_0)} \\ & \quad + \left\| 2\mathcal{H}(ab - a_h \Pi_h b) \right\|_{H^{-d}(\partial D_0)} \\ & \quad + 2 \left\| A|_{\mathbf{x}=\mathbf{y}} - A_h|_{\mathbf{x}=\mathbf{y}} \right\|_{H^{-d}(\partial D_0)}. \end{aligned}$$

Herein, the first term on the right hand side can be bounded by $\lesssim h^{2d}$ by applying the standard approximation estimates of the L^2 -projection (VI.4.1). The second term on the right hand can

be estimated by using the inequality $\|uv\|_{H^{-d+2}(\partial D_0)} \lesssim \|u\|_{H^{d-2}(\partial D_0)}\|v\|_{H^{-d+2}(\partial D_0)}$ derived in the proof of Lemma VI.4.2:

$$\begin{aligned} \|2\mathcal{H}(ab - a_h\Pi_h b)\|_{H^{-d}(\partial D_0)} &\leq \|2\mathcal{H}(ab - a_h\Pi_h b)\|_{H^{-d+2}(\partial D_0)} \\ &\lesssim \|2\mathcal{H}\|_{H^{d-2}(\partial D_0)}\|ab - a_h\Pi_h b\|_{H^{-d+2}(\partial D_0)}. \end{aligned}$$

To estimate the third term on the right hand side, we follow the arguments in [DDH15, Chapter 3.3] and remark that the diagonal operator $H_{\text{mix}}^{s,s}(\partial D_0 \times \partial D_0) \rightarrow W^{s,1}(\partial D_0)$ is continuous for $s \geq 1$. Exploiting that the embedding $W^{s,1}(\partial D_0) \hookrightarrow H^{s-1}(\partial D_0)$ is also continuous, we conclude by a shift argument that

$$\left\| A|_{\mathbf{x}=\mathbf{y}} - A_h|_{\mathbf{x}=\mathbf{y}} \right\|_{H^{-d}(\partial D_0)} \lesssim \|A - A_h\|_{H_{\text{mix}}^{-d+1, -d+1}(\partial D_0 \times \partial D_0)}.$$

In view of Lemmata VI.4.2 and VI.4.8, this implies the estimate (VI.4.2).

Estimate (VI.4.3) follows finally from (VI.4.2) and Strang's first lemma by considering the Galerkin projection as the solution of $\text{Id}\psi = g$ with a perturbed right-hand side. \square

We are now in the position to bound the error of the Neumann data $t^{(\mathbb{E}[\delta^2 u])}$ of $\mathbb{E}[\delta^2 u]$.

Lemma VI.4.10. *Let $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]|_{\mathbf{x}=\mathbf{y}} \in H^d(\partial D_0)$ and $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \in H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$. Then, the Neumann data $t^{(\mathbb{E}[\delta^2 u])}$ are approximated with the rate*

$$\left\| t^{(\mathbb{E}[\delta^2 u])} - t_h^{(\mathbb{E}[\delta^2 u])} \right\|_{H^{-d-1}(\partial D_0)} \lesssim h^{2(d-1)}.$$

Proof. Since $\mathcal{V}: H^d(\partial D_0) \rightarrow H^{d+1}(\partial D_0)$ is self-adjoint, continuous, and boundedly invertible in case of a $C^{d,1}$ -boundary, it holds

$$\begin{aligned} \left\| t^{(u_0)} - t_h^{(u_0)} \right\|_{H^{-d-1}(\partial D_0)} &= \sup_{\|v\|_{H^{d+1}(\partial D_0)}=1} \left(t^{(u_0)} - t_h^{(u_0)}, v \right)_{L^2(\partial D_0)} \\ &= \sup_{\|w\|_{H^d(\partial D_0)}=1} \frac{\left(\mathcal{V} \left(t^{(u_0)} - t_h^{(u_0)} \right), w \right)_{L^2(\partial D_0)}}{\|\mathcal{V}w\|_{H^{d+1}(\partial D_0)}} \\ &\leq \sup_{\|w\|_{H^d(\partial D_0)}=1} \frac{\left(\mathcal{V} \left(t^{(u_0)} - t_h^{(u_0)} \right), w - \Pi_h w \right)_{L^2(\partial D_0)}}{\|\mathcal{V}w\|_{H^{d+1}(\partial D_0)}} \\ &\quad + \sup_{\|w\|_{H^d(\partial D_0)}=1} \frac{\left(\mathcal{V} \left(t^{(u_0)} - t_h^{(u_0)} \right), \Pi_h w \right)_{L^2(\partial D_0)}}{\|\mathcal{V}w\|_{H^{d+1}(\partial D_0)}}. \end{aligned}$$

Using the continuity and bounded invertibility of \mathcal{V} , we further estimate

$$\begin{aligned} &\sup_{\|w\|_{H^d(\partial D_0)}=1} \frac{\left(\mathcal{V} \left(t^{(u_0)} - t_h^{(u_0)} \right), w - \Pi_h w \right)_{L^2(\partial D_0)}}{\|\mathcal{V}w\|_{H^{d+1}(\partial D_0)}} \\ &\leq \left\| \mathcal{V} \left(t^{(u_0)} - t_h^{(u_0)} \right) \right\|_{H^{1/2}(\partial D_0)} \sup_{\|w\|_{H^d(\partial D_0)}=1} \frac{\|w - \Pi_h w\|_{H^{-1/2}(\partial D_0)}}{\|\mathcal{V}w\|_{H^{d+1}(\partial D_0)}} \\ &\lesssim \left\| t^{(u_0)} - t_h^{(u_0)} \right\|_{H^{-1/2}(\partial D_0)} \sup_{\|w\|_{H^d(\partial D_0)}=1} \frac{\|w - \Pi_h w\|_{H^{-1/2}(\partial D_0)}}{\|w\|_{H^d(\partial D_0)}}. \end{aligned}$$

Using standard error estimates for the Galerkin method, the error of the Neumann data in the $H^{-1/2}(\partial D_0)$ -norm is bounded by $\lesssim h^{d+1/2}$, whereas the second factor can be estimated by the standard error estimates for the L^2 -projection.

To estimate the second part of the sum, consider

$$\left(\mathcal{V}\left(t^{(u_0)} - t_h^{(u_0)}\right), v_h \right)_{L^2(\partial D_0)} = \left(\frac{1}{2}(g - g_h) + \mathcal{K}(g - \Pi_h g_h), v_h \right)_{L^2(\partial D_0)} \quad \text{for all } v_h \in V_h,$$

and conclude

$$\begin{aligned} & \left(\mathcal{V}\left(t^{(u_0)} - t_h^{(u_0)}\right), \Pi_h w \right)_{L^2(\partial D_0)} \\ &= \left(\frac{1}{2}(g - g_h) + \mathcal{K}(g - \Pi_h g_h), \Pi_h w \right)_{L^2(\partial D_0)} \\ &= \left(\frac{1}{2}(g - g_h) + \mathcal{K}(g - \Pi_h g_h), w \right)_{L^2(\partial D_0)} \\ &\quad + \left(\mathcal{K}(g - \Pi_h g_h), \Pi_h w - w \right)_{L^2(\partial D_0)} + \frac{1}{2} \left(g - g_h, \Pi_h w - w \right)_{L^2(\partial D_0)} \\ &\lesssim \left(\|g - g_h\|_{H^{-d}(\partial D_0)} + \|\mathcal{K}(g - \Pi_h g_h)\|_{H^{-d}(\partial D_0)} \right) \|w\|_{H^d(\partial D_0)} \\ &\quad + \|\mathcal{K}(g - \Pi_h g_h)\|_{H^1(\partial D_0)} \|w - \Pi_h w\|_{H^{-1}(\partial D_0)} \\ &\quad + \|g - g_h\|_{L^2(\partial D_0)} \|w - \Pi_h w\|_{L^2(\partial D_0)}. \end{aligned}$$

The assertion now follows by exploiting that $\mathcal{K}: H^s(\partial D_0) \rightarrow H^s(\partial D_0)$ is a continuous operator for $s = -d$ and $s = 1$ on $C^{d,1}$ -boundaries, the previous lemma, and inverse estimates in complete analogy to the proof of Lemma VI.4.1. \square

As an immediate consequence of the error estimates, we finally obtain an error estimate for the potential of the second order local shape derivative $\mathbb{E}[\delta^2 u]$. The following approximation result can be derived by standard arguments, see [Ste08, Chapter 12.1] for instance.

Lemma VI.4.11. *Let $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]_{\mathbf{x}=\mathbf{y}} \in H^d(\partial D_0)$ and $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \in H_{\text{mix}}^{d,d}(\partial D_0 \times \partial D_0)$. Then, for the mean of the second order local shape derivative $\mathbb{E}[\delta^2 u]$ from (VI.1.11), it holds that*

$$|\mathbb{E}[\delta^2 u](\mathbf{x}) - \mathbb{E}[\delta^2 u]_h(\mathbf{x})| \lesssim h^{2(d-1)} \quad \text{for all } \mathbf{x} \in K \Subset D_{\Omega}.$$

Remark VI.4.12. *Although we can only prove a reduced convergence rate of $h^{2(d-1)}$, we will see in the numerical experiments that we reach the same convergence rate as for the correlation, i.e., h^{2d} .*

VI.5 Hierarchical Matrix Compression

As discussed in the previous chapters, all explicitly given matrices in (VI.3.3), (VI.3.4), and (VI.3.5) can be represented by hierarchical matrices, if we restrict ourselves to asymptotically smooth correlations. It remains to discuss the \mathcal{H} -matrix approximability of the unknowns $\mathbf{C}_{\rho, \text{Cor}[\delta u]}$ in (VI.3.3) and \mathbf{A} in (VI.3.4). Unfortunately, Theorem II.5.18 does not immediately show the asymptotical smoothness of A and $\rho^{(\text{Cor}[\delta u])}$, but it can be extended to the following slightly modified version.

Corollary VI.5.1. *Let ∂D_0 be a compact, analytic manifold and let $\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle]$ give rise to a pseudodifferential operator of Gevrey class $s > 1$ on ∂D_0 . Let further g be of Gevrey class s . Then, the solutions of the equations (VI.2.2) and (VI.2.5) are, in local coordinates, asymptotically smooth functions in one variable.*

Proof. We first remark that the single layer operator belongs to $OPS_{cl,s}^{-1}$ and the double layer operator belongs to $OPS_{cl,s}^0$ for $s > 1$, cf. [HW08, Chapter 9.2]. Let us further remark that

the multiplication of two functions $g(\mathbf{x})f(\mathbf{x})$ can be written as an application of an integral operator

$$g(\mathbf{x})f(\mathbf{x}) = \int_{\partial D_0} k(\mathbf{x}, \mathbf{x} - \mathbf{y})f(\mathbf{y}) d\sigma_{\mathbf{y}} =: (\Psi f)(\mathbf{x}),$$

with distributional Schwartz kernel $k(\mathbf{x}, \mathbf{x} - \mathbf{y}) = g(\mathbf{x})\delta_0(\|\mathbf{x} - \mathbf{y}\|)$. Since the Fourier transform of the delta distribution is a constant, the multiplication by a function gives thus rise to a pseudodifferential operator $OPS_{cl,s}^0$. Rewriting finally (VI.2.2) as

$$(\text{Id} \otimes \mathcal{V})A = \left(\text{Id} \otimes \left(\frac{1}{2} + \mathcal{K} \right) \right) \left(\left(\mathbb{1} \otimes \left(\frac{\partial g}{\partial \mathbf{n}} - t^{(u_0)} \right) \text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle] \right) \right),$$

the rest of the proof is in complete analogy to [DHS17, Theorems 1,2, and 3]. \square

Knowing that it is reasonable to represent all matrices in (VI.3.3) and (VI.3.4) as \mathcal{H} -matrices, it remains to discuss how to actually compute the representations for $\mathbf{C}_{\rho, \text{Cor}[\delta u]}$ in (VI.3.3) and \mathbf{A} in (VI.3.4). Therefore, we may compute the right-hand sides using the \mathcal{H} -matrix arithmetic. To solve the matrix equations, we may use the same strategy as in the previous chapters and use the iterative refinement from (III.3.1). For the more general case of an non-symmetric matrix equation $\Psi \mathbf{X} \Phi^T = \mathbf{R}$, the iterative refinement can be modified as follows. Let $\Xi \approx \hat{\mathbf{L}}_{\Xi} \hat{\mathbf{U}}_{\Xi}$, where $\hat{\mathbf{L}}_{\Xi}, \hat{\mathbf{U}}_{\Xi} \in \mathcal{H}(\mathcal{B}, k)$, $\Xi \in \{\Psi, \Phi\}$, be approximate LU-factorizations to Ψ and Φ , e.g., computed by the \mathcal{H} -matrix arithmetic. Starting with the initial guess $\mathbf{X}_0 = \hat{\mathbf{U}}_{\Psi}^{-1} \hat{\mathbf{L}}_{\Psi}^{-1} \mathbf{R} \hat{\mathbf{L}}_{\Phi}^T \hat{\mathbf{U}}_{\Phi}^T$, we iterate

$$\Theta_i = \mathbf{R} - \Psi \mathbf{X}_i \Phi^T, \quad \mathbf{X}_{i+1} = \mathbf{X}_i + \hat{\mathbf{U}}_{\Psi}^{-1} \hat{\mathbf{L}}_{\Psi}^{-1} \Theta_i \hat{\mathbf{L}}_{\Phi}^T \hat{\mathbf{U}}_{\Phi}^T, \quad i = 0, 1, \dots$$

Before we conclude this section, we remark that the approximate LU-factorization can be replaced by an approximate Cholesky factorization if the corresponding matrix is symmetric and positive definite, such as the system matrix of the single layer operator.

VI.6 Numerical Examples

The following numerical experiments are divided into three parts. The first part is concerned with the convergence of the Galerkin scheme for the correction terms $\text{Cor}[\delta u]_h$ and $\mathbb{E}[\delta^2 u]_h$ with respect to the mesh size h . The second part is concerned with the asymptotics of the perturbation approach in ε , whereas the third part deals with an example on non-smooth domains.

All of the computations in the following experiments were carried out on a computing server with two Intel(R) Xeon(R) CPU E5-2643 v3 with a clock rate of 3.40GHz and a main memory of 256GB. Each of the CPUs provides 12 physical cores, thus, with Hyper-Threading enabled, we may access up to 24 logical cores in total. For the discretization and the assembly of the \mathcal{H} -matrices, we employ the black-box higher order FMM from Chapter IV. The computations in the product domain $\partial D_0 \times \partial D_0$ are based on discontinuous (elementwise) polynomial ansatz functions and the computations in the non-product domain on globally continuous B-splines of the same order. For the arithmetic \mathcal{H} -matrix operations, we apply the \mathcal{H} -matrix arithmetic tailored to parametric surfaces from the previous chapter with its simple OpenMP parallelization.

VI.6.1 Convergence in h

To construct an example, where the solution is analytically known, we consider the unit ball whose boundary is perturbed in normal direction in accordance with

$$\varepsilon \langle \mathbf{V}(\omega, \mathbf{x}), \mathbf{n}_{\mathbf{x}} \rangle = Y_m^{\ell}(\mathbf{x}) X(\omega),$$

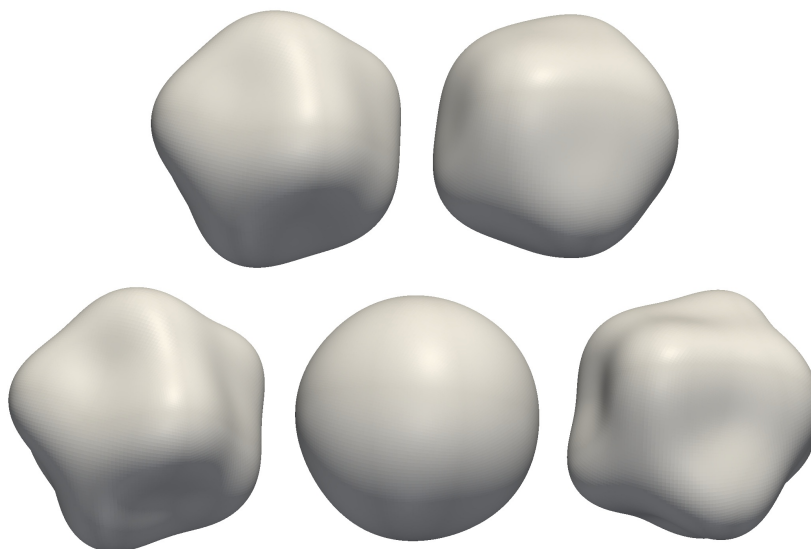


Figure VI.2: Different realizations of the deformed unit ball for the spherical harmonic Y_3^5 and $\varepsilon = 0.3$.

where Y_m^ℓ is a spherical harmonic and $X(\omega)$ is a uniformly distributed random variable on $[-\varepsilon, \varepsilon]$. The correlation of this boundary perturbation thus implies

$$\text{Cor}[\langle \mathbf{V}, \mathbf{n} \rangle](\mathbf{x}, \mathbf{y}) = \frac{1}{3} Y_m^\ell(\mathbf{x}) Y_m^\ell(\mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \partial D_0.$$

Several possible shapes are depicted in Figure VI.2, while the used compactum K was visualized in Figure VI.1.

As boundary values on the hold-all domain, we choose

$$g(\mathbf{x}) = \|\mathbf{x}\|^2, \quad \mathbf{x} \in D_\varepsilon^{\cup\Omega}.$$

Since $\|\mathbf{x}\|^\ell Y_m^\ell(\mathbf{x}/\|\mathbf{x}\|)$ is harmonic in \mathbb{R}^3 , it is then a short computation to show that it holds

$$\text{Cor}[\delta u](\mathbf{x}, \mathbf{y}) = \frac{4}{3} \|\mathbf{x}\|^\ell \|\mathbf{y}\|^\ell Y_m^\ell\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right) Y_m^\ell\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right), \quad \mathbf{x}, \mathbf{y} \in D_0.$$

Since the spherical harmonic Y_m^ℓ is an eigenfunction of the Dirichlet-to-Neumann map with eigenvalue ℓ on the sphere, cf. [CK12], and since the unit sphere has the constant mean curvature of $\mathcal{H} = -1$, cf. [Gra06], one verifies that there holds

$$g^{(\mathbb{E}[\delta^2 u])}(\mathbf{x}) = \frac{2 - 4\ell}{3} Y_m^\ell(\mathbf{x})^2 \quad \mathbf{x} \in \partial D_0,$$

for the Dirichlet data of the mean's second order correction term $\mathbb{E}[\delta^2 u]$ from (VI.1.11). Having thus access to the exact Dirichlet data, we can easily compute an overkill solution of $\mathbb{E}[\delta^2 u]$ as a reference solution by an additional refinement in h .

Figure VI.3 validates that we achieve the convergence rates predicted by Lemma VI.4.6 and Lemma VI.4.11 for both, $\text{Cor}[\delta u]$ and $\mathbb{E}[\delta^2 u]$. In fact, $\mathbb{E}[\delta^2 u]$ has even a higher convergence rate than predicted. Although not covered by our theory, the case $d = 1$, i.e., the case of piecewise constant boundary elements, seems to converge as well at a rate of h^2 . Notice that the computed quantities are independent of the amplitude of the deformation ε , as long as the compactum $K \Subset D_\varepsilon^{\cap\Omega}$ does not depend on ε .

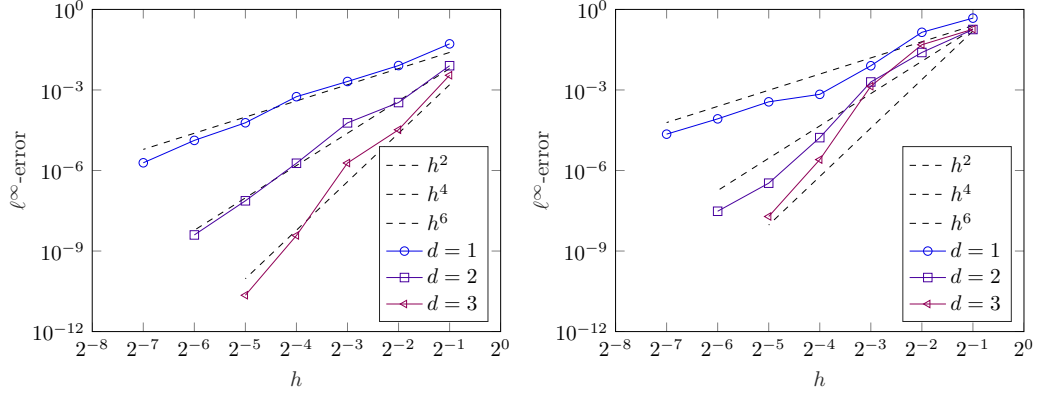


Figure VI.3: Convergence rates for $\text{Cor}[\delta u]_h$ (left) and $\mathbb{E}[\delta^2 u]_h$ (right) in h on the deformed unit ball for different orders d of the ansatz functions.

VI.6.2 Convergence in ε

As the mean, the correlation, and the covariance for the example on the unit ball are not analytically known, we have to use quadrature methods to compute a reference solution in order to study the asymptotics in ε . Choosing $\varepsilon = 0.05, 0.1, 0.15, 0.2, 0.25, 0.3$, we use a one-dimensional 15 point Gauss-Legendre quadrature rule to compute the stochastic integral of the mean, the correlation, and the covariance on a spatial discretization with continuous piecewise linear ansatz functions on a mesh with 24,576 elements. The PDE solves are accelerated by an OpenMP parallelization.

Since the probability distribution of $\langle \mathbf{V}, \mathbf{n} \rangle$ is symmetric around zero, we can expect fourth order accurate approximations to the mean, the correlation, and the covariance in ε . Thus, for the numerical solutions $\mathbb{E}[u_\varepsilon]_h$, $\text{Cor}[u_\varepsilon]_h$, and $\text{Cov}[u_\varepsilon]_h$ computed by our expansions, the errors should behave like

$$\begin{aligned} |\mathbb{E}[u_\varepsilon] - \mathbb{E}[u_\varepsilon]_h| &= \mathcal{O}(h^{2d} + \varepsilon^2 h^{2(d-1)} + \varepsilon^4) && \text{in } K, \\ |\text{Cor}[u_\varepsilon] - \text{Cor}[u_\varepsilon]_h| &= \mathcal{O}(h^{2d} + \varepsilon^2 h^{2d} + \varepsilon^4) && \text{in } K \times K, \\ |\text{Cov}[u_\varepsilon] - \text{Cov}[u_\varepsilon]_h| &= \mathcal{O}(\varepsilon^2 h^{2d} + \varepsilon^4) && \text{in } K \times K, \end{aligned}$$

as h and ε tend to zero. The fourth order asymptotic is then reached as soon as the mesh size is small enough.

Figures VI.4, VI.5, and VI.6 show the errors for the former second order (see [HSS08b] for the details) and the new fourth order accurate approximations. We indeed reach the fourth order accuracy already for relatively coarse mesh sizes. A comparison with the second order accurate approximation shows that the consideration of the correction terms for the fourth order approximation can improve the error by several orders of magnitude.

VI.6.3 Non-smooth Boundaries

In order to demonstrate that the perturbation approach is not necessarily limited to smooth surfaces, we consider the unit cube $D_0 = [0, 1]^3$ as the reference domain. We assume the perturbing vector field to deform the upper side of the cube. More precisely, given uniformly distributed random variables $X_{ij} \sim U[-1, 1]$, $i, j = 1, \dots, 4$, the perturbation field is given as

$$\mathbf{V}(\omega, \mathbf{x}) = \begin{cases} \sum_{i,j=1}^4 B_i(x_1) B_j(x_2) X_{ij}(\omega), & x_3 = 1, \\ 0, & \text{otherwise,} \end{cases}$$

where B_i , $i = 1, \dots, 4$ are fifth order B-splines on $[0, 1]$, which, as well as their derivatives, are zero at the interval boundaries. As a result, the shape derivatives are well defined and no

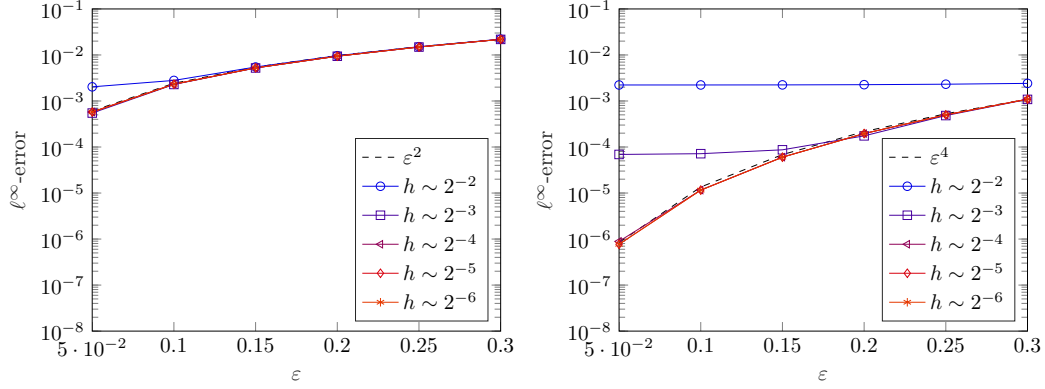


Figure VI.4: Asymptotics in ε for the numerical approximation of the second order (left) and fourth order (right) accurate expansions for the mean by different mesh sizes h .

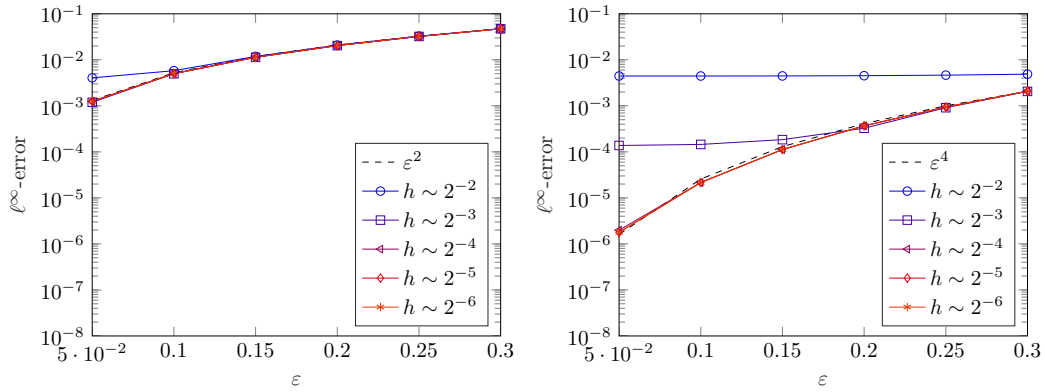


Figure VI.5: Asymptotics in ε for the numerical approximation of the second order (left) and fourth order (right) accurate expansions for the correlation by different mesh sizes h .

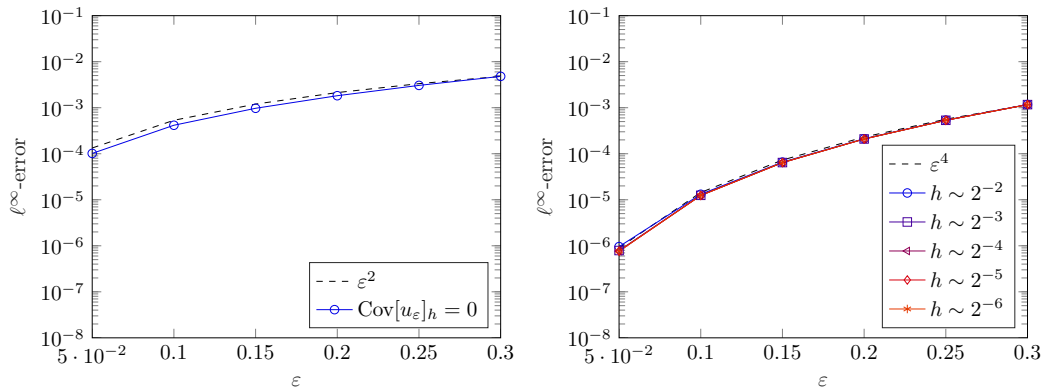


Figure VI.6: Asymptotics in ε for the numerical approximation of the second order (left) and fourth order (right) accurate expansions for the covariance by different mesh sizes h .

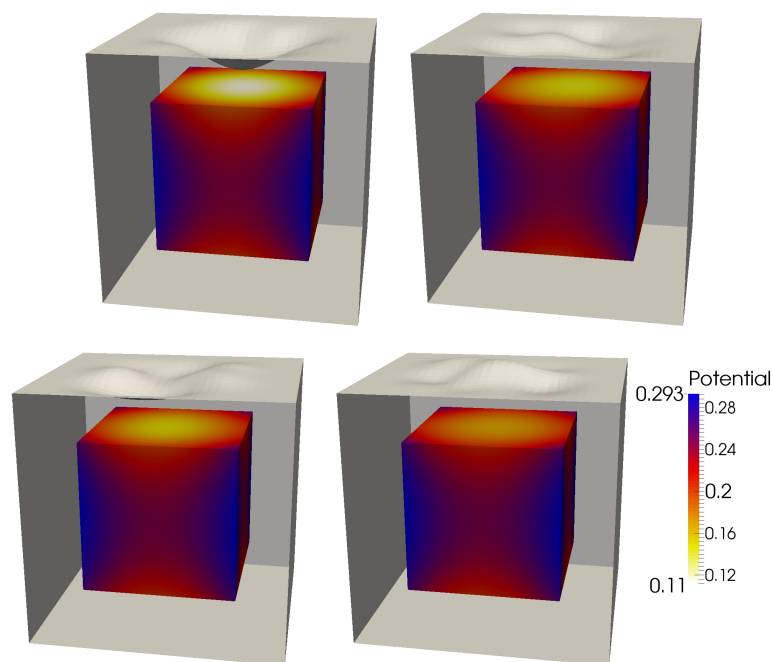


Figure VI.7: Realizations of the perturbed cube with its solution visualized on the inscribed compactum K for $\varepsilon = 0.05$.

singularities occur in the boundary values of the PDEs for the correction terms, which means that the expansions for the mean, the correlation, and the covariance are well defined.

A visualization of different realizations for $\varepsilon = 0.05$ for the Dirichlet boundary values

$$g(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$$

and the used compactum K with the corresponding solution of (VI.1.2) is found in Figure VI.7.

Since the stochastic domain is truly 16-dimensional, we use a Monte-Carlo simulation with 10,000 samples for a visual comparison with the asymptotic expansions for the choice $\varepsilon = 0.05$. All computations are performed by using continuous piecewise linear ansatz functions on a mesh with 24,576 elements, while we accelerate the Monte-Carlo simulation by an MPI parallelization and by computing only the changed matrix entries for each sample. The computation time on all 24 logical cores takes for all quantities 67,241 seconds using the perturbation approach and 496,582 seconds using the Monte-Carlo simulation. However, due to the slow convergence rate of the Monte-Carlo simulation and the high-dimensionality of the problem, we only aim at a qualitative comparison of the mean and the diagonal of the correlation, which is presented in Figure VI.8 and Figure VI.9.

VI.7 Conclusion

In this chapter, we considered the approximate computation of the mean, the correlation, and the covariance of the solution of PDEs on random domains by the perturbation approach. Additionally to existing third order accurate expansions of the covariance discussed in [HSS08b], we derived third order accurate expansions of the mean and the correlation and discussed their numerical computation. These expansions become even fourth order accurate for specific types of boundary variations. While the solution on the unperturbed domain yields a second order accurate solution, the correction terms for the more accurate expansions are given by correlation equations.

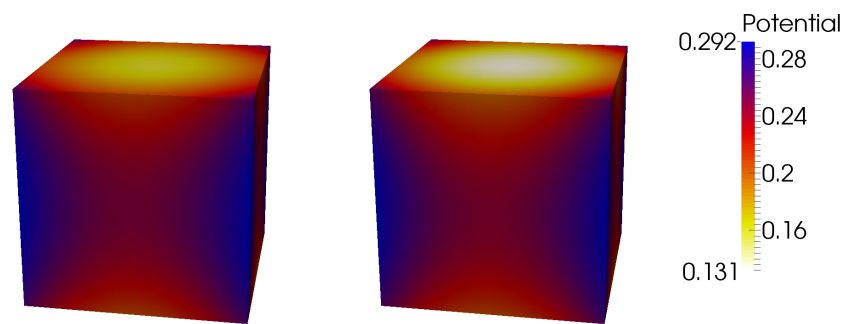


Figure VI.8: Comparison between the perturbation approach (left) and a Monte-Carlo simulation (right) with 10,000 samples for $\varepsilon = 0.05$ for the mean.

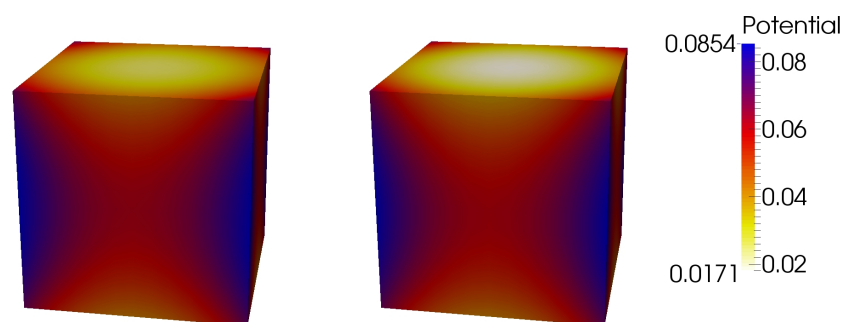


Figure VI.9: Comparison between the perturbation approach (left) and a Monte-Carlo simulation (right) with 10,000 samples for $\varepsilon = 0.05$ for the diagonal of the correlation.

Omitting the meshing of the computational domain, we employ the boundary element method to discretize the correlation equations with ansatz spaces of piecewise polynomials of at least second order. To obtain computational efficiency, we solve the corresponding matrix equations in almost linear complexity by the \mathcal{H} -matrix approach. The numerical experiments in three dimensions validate the asymptotic expansions and show the convergence of the proposed method for piecewise linear and piecewise quadratic boundary elements. Piecewise constant boundary elements, although not covered by the theory, seem to work as well. Finally, we provided numerical experiments which demonstrate the perturbation approach could be applied on nonsmooth domains, too.

Chapter VII

Final Remarks

In this thesis, we presented a new approach for the solution of correlation equations

$$(A \otimes A) \text{Cor}[u] = \text{Cor}[f],$$

which is robust to “rough” correlations $\text{Cor}[f]$, that is, $\text{Cor}[f]$ has low Sobolev smoothness or a high concentration of measure around the origin. The resulting matrix equations

$$\mathbf{A}\mathbf{C}_u\mathbf{A}^\top = \mathbf{C}_f$$

occur from the discretization by the finite element method or the boundary element method. They are, since, in general, \mathbf{C}_f and \mathbf{C}_u are densely populated, prohibitively expensive to solve with naive approaches. As common low-rank and sparse tensor product techniques are known to struggle with rough correlations, we proposed a different solution strategy. To achieve computational efficiency, we represented the correlation matrices as well as the finite element and boundary element matrices with hierarchical matrices. Using the hierarchical matrix arithmetic, we were able to solve the correlation equations in almost linear time, i.e., linear up to (poly-)logarithmic factors, with respect to the dimension of the used finite element spaces. The representation of the unknown as a hierarchical matrix is theoretically justified, at least on sufficiently smooth domains.

We discussed several examples from the uncertainty quantification of PDEs with random input data where such correlation equations occur. In particular, we discussed random loads, random boundary values, and random domains. While the first two examples led to linear operator equations with random right-hand side, the dependence becomes non-linear on random domains. Therefore, we employed an approach based on perturbation theory and asymptotic expansions. Additional to the existing third order accurate expansion of the correlation, we also provided a third order accurate expansion of the mean and a full convergence analysis. Both expansions become fourth order accurate under certain conditions.

The convergence analysis shows that we need higher order boundary elements for the Galerkin scheme of the perturbation approach for random domains. Therefore, we developed an efficient fast multipole method for higher order ansatz functions on parametric surfaces. Such surfaces arise for example in isogeometric analysis, which aims at bridging the gap between computer aided design and finite element methods and boundary element methods. The developed multipole method fits into the framework of hierarchical matrices as well as in its improved \mathcal{H}^2 -variant. The numerical experiments showed the benefit of the \mathcal{H}^2 -setting for the matrix-vector multiplication and that the higher order approach asymptotically reaches more accurate solutions in shorter time, even on non-smooth and complex geometries.

Possible future work could be the treatment of PDEs with random diffusion coefficients by the perturbation approach. Corresponding asymptotic expansions can for example be found in [HPS13]. One may use the finite element techniques for hierarchical matrices similar to Chapter III to compute the correction terms.

Concerning hierarchical matrix techniques, one may consider the use of \mathcal{H}^2 -matrices instead of hierarchical matrices for the correlation equations. Using the arithmetic proposed in [Bör06], one could possibly solve the correlation equations in linear complexity with significantly lower constants. If finite elements are used, one may also directly consider the matrix Galerkin scheme proposed in [BB13, Boy15] to solve the matrix equation. However, both of these techniques are known to be more involved than hierarchical matrices and their application requires careful investigation.

Bibliography

- [Гал15] Б. Г. Галеркин. Прутки и плиты. Серия, посвященная различным вопросам упругого равновесия стержней и пластин. *Вестник инженера*, 19:897–908, 1915.
- [AF03] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*. Academic Press, Oxford, 2003.
- [Alt02] H. W. Alt. *Lineare Funktionalanalysis: Eine Anwendungsorientierte Einführung*. Springer, Berlin, 2002.
- [Aro55] N. Aronszajn. Boundary values of functions with finite Dirichlet integral. *Studies in Eigenvalue Problems*, 14, 1955.
- [AS64] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and mathematical tables*. National Bureau of Standards Applied Mathematics Series. Dover Publications, N. Chemsford, MA, 1964.
- [BB13] D. Boysen and S. Börm. A Galerkin approach for solving matrix equations with hierarchical matrices. In L. Cvetković, T. Atanacković, and V. Kostić, editors, *84th Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM)*, volume 13 of *Proceedings in Applied Mathematics and Mechanics*, pages 405–406. Wiley, Weinheim, 2013.
- [BCR91] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991.
- [BdMK67] L. Boutet de Monvel and P. Krée. Pseudo-differential operators and Gevrey classes. *Annales de l'Institut Fourier (Grenoble)*, 17(fasc. 1):295–323, 1967.
- [Beb00] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.
- [Beb05] M. Bebendorf. Efficient inversion of the Galerkin matrix of general second-order elliptic operators with nonsmooth coefficients. *Mathematics of Computation*, 74(251):1179–1199, 2005.
- [Beb07] M. Bebendorf. Why finite element discretizations can be factored by triangular hierarchical matrices. *SIAM Journal on Numerical Analysis*, 45(4):1472–1494, 2007.
- [Beb08] M. Bebendorf. *Hierarchical Matrices*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin-Heidelberg, 2008.
- [BG04a] S. Börm and L. Grasedyck. Low-rank approximation of integral operators by interpolation. *Computing*, 72(3-4):325–332, 2004.
- [BG04b] H. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

- [BGH03] S. Börm, L. Grasedyck, and W. Hackbusch. Hierarchical matrices. Technical Report 21, Max Planck Institute for Mathematics in the Sciences, 2003.
- [BH03] M. Bebendorf and W. Hackbusch. Existence of \mathcal{H} -matrix approximants to the inverse FE-matrix of elliptic operators with L^∞ -coefficients. *Numerische Mathematik*, 95(1):1–28, 2003.
- [BH15] A Behzadan and M Holst. Multiplication in Sobolev spaces, revisited. *Preprint*, arXiv:1512.07379, 2015.
- [BNT07] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.
- [Boc33] S. Bochner. Integration von Funktionen, deren Werte die Elemente eines Vektorraumes sind. *Fundamenta Mathematicae*, 20(1):262–176, 1933.
- [Bör06] S. Börm. \mathcal{H}^2 -matrix arithmetics in linear complexity. *Computing*, 77(1):1–28, 2006.
- [Bör10] S. Börm. *Efficient Numerical Methods for Non-local Operators*, volume 14 of *EMS Tracts in Mathematics*. European Mathematical Society (EMS), Zürich, 2010.
- [Bör17] S. Börm. Hierarchical matrix arithmetic with accumulated updates. *Preprint*, arXiv:1703.09085, 2017.
- [Boy15] D. Boysen. *Das Matrix-Galerkin-Verfahren*. PhD thesis, Christian-Albrechts-Universität zu Kiel, 2015.
- [BP02] I. Babuška and C. Panagiotis. On solving elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 191(37–38):4093–4122, 2002.
- [Bra13] D. Braess. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer, Berlin, fifth edition, 2013.
- [BSZ11] A. Barth, C. Schwab, and N. Zollinger. Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numerische Mathematik*, 119(1):123–161, 2011.
- [BT02] I. Brainman and S. Toledo. Nested-dissection orderings for sparse LU with partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 23(4):998–1012, 2002.
- [BTZ04] I. Babuška, R. Tempone, and G. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004.
- [Соб38] С. Л. Соболев. Об одной теореме функционального анализа. *Математический сборник*, 4(3):471–497, 1938.
- [Сло58] Л. Н. Слободецкий. Обобщенные пространства С. Л. Соболева и их приложения к краевым задачам в частных производных. *Ученые записки Ленинградского педагогического института имени А. И. Герцена*, 197:54–112, 1958.
- [Caf98] R. Caffisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.

-
- [CCNT16] J. E. Castrillón-Candás, F. Nobile, and R. F. Tempone. Analytic regularity and collocation approximation for elliptic PDEs with random domain deformations. *Computers & Mathematics with Applications*, 71(6):1173–1197, 2016.
- [CCNT17] J. E. Castrillón-Candás, F. Nobile, and R. F. Tempone. Hybrid collocation perturbation for PDEs with random domains. *Preprint*, arXiv:1703.10040, 2017.
- [CHB09] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, Hoboken, NJ, first edition, 2009.
- [CHR07] J. A. Cottrell, T. J. R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.
- [CK12] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*, volume 93 of *Applied Mathematical Sciences*. Springer Science & Business Media, New York, 2012.
- [CK13] D. Colton and R. Kress. *Integral Equation Methods in Scattering Theory*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2013.
- [CPT15] A. Chernov, D. Pham, and T. Tran. A shape calculus based method for a transmission problem with a random interface. *Computers & Mathematics with Applications*, 70(7):1401–1424, 2015.
- [DB78] C. De Boor. *A Practical Guide to Splines*. Springer, New York, 1978.
- [DBO01] M. Deb, I. Babuška, and J. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001.
- [DDH15] M. Dambrine, C. Dapogny, and H. Harbrecht. Shape optimization for quadratic functionals and states with random right-hand sides. *SIAM Journal on Control and Optimization*, 53(5):3081–3103, 2015.
- [DH17] J. Dölz and H. Harbrecht. Hierarchical matrix approximation for the uncertainty quantification of potentials on random domains. *Preprint 2017-05, Fachbereich Mathematik, Universität Basel, Switzerland*, 2017.
- [DHP15] J. Dölz, H. Harbrecht, and M. Peters. \mathcal{H} -matrix accelerated second moment analysis for potentials with rough correlation. *Journal of Scientific Computing*, 65(1):387–410, 2015.
- [DHP16] J. Dölz, H. Harbrecht, and M. Peters. An interpolation-based fast multipole method for higher-order boundary elements on parametric surfaces. *International Journal for Numerical Methods in Engineering*, 108(13):1705–1728, 2016.
- [DHP17] J. Dölz, H. Harbrecht, and M. Peters. \mathcal{H} -matrix based second moment analysis for rough random fields and finite element discretizations. *SIAM Journal on Scientific Computing*, 39(4):B618–B639, 2017.
- [DHS06] W. Dahmen, H. Harbrecht, and R. Schneider. Compression techniques for boundary integral equations. Asymptotically optimal complexity estimates. *SIAM Journal on Numerical Analysis*, 43(6):2251–2271, 2006.
- [DHS17] J. Dölz, H. Harbrecht, and C. Schwab. Covariance regularity and \mathcal{H} -matrix approximation for rough random fields. *Numerische Mathematik*, 135(4):1045–1071, 2017.
-

- [DKS13] J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.
- [DNP⁺04] B. J. Deusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, and O. P. Le Maître. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal on Scientific Computing*, 26(2):698–719, 2004.
- [DNPV12] E. Di Nezza, G. Palatucci, and E. Valdinoci. Hitchhiker’s guide to the fractional Sobolev spaces. *Bulletin des Sciences Mathématiques*, 136(5):521–573, 2012.
- [Duf82] M. G. Duffy. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM Journal on Numerical Analysis*, 19(6):1260–1262, 1982.
- [DZ01] M. Delfour and J. Zolésio. *Shapes and Geometries*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2001.
- [Epp00] K. Eppler. Boundary integral representations of second derivatives in shape optimization. *Discussiones Mathematicae, Differential Inclusions, Control and Optimization*, 20(1):63–78, 2000.
- [Fau15] M. Faustmann. *Approximation inverser Finite Elemente- und Randelementematrizen mittels hierarchischer Matrizen*. PhD Thesis, Technische Universität Wien, 2015.
- [FMP15] M. Faustmann, J. M. Melenk, and D. Praetorius. \mathcal{H} -matrix approximability of the inverses of FEM matrices. *Numerische Mathematik*, 131(4):615–642, 2015.
- [FST05] P. Frauenfelder, C. Schwab, and R. A. Todor. Finite elements for elliptic problems with stochastic coefficients. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5):205–228, 2005.
- [Gag58] E. Gagliardo. Proprieta di alcune classi di funzioni in piu variabili. *Ricerche di Matematica*, 7(1):102–137, 1958.
- [Geo73] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.
- [Gev18] M. Gevrey. Sur la nature analytique des solutions des équations aux dérivées partielles. Premier mémoire. *Annales Scientifiques de l’École Normale Supérieure. Troisième Série*, 35:129–190, 1918.
- [GH03] L. Grasedyck and W. Hackbusch. Construction and arithmetics of \mathcal{H} -matrices. *Computing*, 70(4):295–334, 2003.
- [GH17] M. Griebel and H. Harbrecht. Singular value decomposition versus sparse grids. Refined complexity estimates. *Preprint 2017–08, Fachbereich Mathematik, Universität Basel, Switzerland.*, 2017.
- [Gie01] K. Giebermann. Multilevel approximation of boundary integral operators. *Computing*, 67(3):183–207, 2001.
- [Gil15] M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259, 2015.
- [GK65] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, 2(2):205–224, 1965.
- [GKLB09] L. Grasedyck, R. Kriemann, and S. Le Borne. Domain decomposition based \mathcal{H} -LU preconditioning. *Numerische Mathematik*, 112(4):565–600, 2009.

-
- [GR87] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.
- [GR97] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.
- [Gra01] L. Grasedyck. *Theorie und Anwendungen Hierarchischer Matrizen*. PhD thesis, Christian-Albrechts-Universität zu Kiel, 2001.
- [Gra06] A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, Boca Raton, FL, 2006.
- [GS03] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Dover Publications, Mineola, New York, 2003.
- [GVL12] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, fourth edition, 2012.
- [GW12] M. J. Gander and G. Wanner. From Euler, Ritz, and Galerkin to modern computing. *SIAM Review*, 54(4):627–666, 2012.
- [Hac95] W. Hackbusch. *Integral Equations*, volume 120 of *International Series of Numerical Mathematics*. Birkhäuser, Basel, 1995.
- [Hac99] W. Hackbusch. A sparse matrix arithmetic based on \mathcal{H} -matrices part I: Introduction to \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
- [Hac15] W. Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*. Springer, Heidelberg, 2015.
- [Har10a] H. Harbrecht. A finite element method for elliptic problems with stochastic input data. *Applied Numerical Mathematics*, 60(3):227–244, 2010.
- [Har10b] H. Harbrecht. On output functionals of boundary value problems on stochastic domains. *Mathematical Methods in the Applied Sciences*, 33(1):91–102, 2010.
- [HB02] W. Hackbusch and S. Börm. \mathcal{H}^2 -matrix approximation of integral operators by interpolation. *Applied Numerical Mathematics*, 43(1–2):129–143, 2002.
- [HCB05] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.
- [HK00a] W. Hackbusch and B. N. Khoromskij. A sparse \mathcal{H} -matrix arithmetic. General complexity estimates. *Journal of Computational and Applied Mathematics*, 125:479–501, 2000.
- [HK00b] W. Hackbusch and B. N. Khoromskij. A sparse \mathcal{H} -matrix arithmetic. II. Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.
- [HKK04] W. Hackbusch, B. N. Khoromskij, and R. Kriemann. Hierarchical matrices based on a weak admissibility criterion. *Computing*, 73(3):207–243, 2004.
- [HKT08] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtysnikov. Approximate iterations for structured matrices. *Numerische Mathematik*, 109(3):365–383, 2008.
- [HL89] J. Hoschek and D. Lasser. *Grundlagen der Geometrischen Datenverarbeitung*. Teubner, Stuttgart, 1989.
- [HL13] H. Harbrecht and J. Li. First order second moment analysis for stochastic interface problems based on low-rank approximation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(5):1533–1552, 2013.
-

- [HN89] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54(4):463–491, 1989.
- [Hör03] L. Hörmander. *The Analysis of Linear Partial Differential Operators I*. Classics in Mathematics. Springer, Berlin, 2003. Distribution theory and Fourier analysis, Reprint of the second (1990) edition.
- [Hör07] L. Hörmander. *The Analysis of Linear Partial Differential Operators III*. Classics in Mathematics. Springer, Berlin, 2007. Pseudo-differential operators, Reprint of the 1994 edition.
- [HP13] H. Harbrecht and M. Peters. Comparison of fast boundary element methods on parametric surfaces. *Computer Methods in Applied Mechanics and Engineering*, 261–262:39–55, 2013.
- [HP15] H. Harbrecht and M. Peters. The second order perturbation approach for PDEs on random domains. *Preprint 2015–40, Fachbereich Mathematik, Universität Basel, Switzerland.*, 2015.
- [HPS12a] H. Harbrecht, M. Peters, and R. Schneider. On the low-rank approximation by the pivoted Cholesky decomposition. *Applied Numerical Mathematics*, 62:28–440, 2012.
- [HPS12b] H. Harbrecht, M. Peters, and M. Siebenmorgen. On multilevel quadrature for elliptic stochastic partial differential equations. In J. Garcke and M. Griebel, editors, *Sparse Grids and Applications*, pages 161–179. Springer, Berlin-Heidelberg, 2012.
- [HPS13] H. Harbrecht, M. Peters, and M. Siebenmorgen. Combination technique based k -th moment analysis of elliptic problems with random diffusion. *Journal of Computational Physics*, 252:128–141, 2013.
- [HPS15] H. Harbrecht, M. Peters, and M. Siebenmorgen. Efficient approximation of random fields for numerical applications. *Numerical Linear Algebra with Applications*, 22(4):596–617, 2015.
- [HPS16a] H. Harbrecht, M. Peters, and M. Siebenmorgen. Analysis of the domain mapping method for elliptic diffusion problems on random domains. *Numerische Mathematik*, 134(4):823–856, 2016.
- [HPS16b] H. Harbrecht, M. Peters, and M. Siebenmorgen. Multilevel accelerated quadrature for PDEs with log-normally distributed diffusion coefficient. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):520–551, 2016.
- [HR98] B. Hendrickson and E. Rothberg. Improving the run time and quality of nested dissection ordering. *SIAM Journal on Scientific Computing*, 20(2):468–489, 1998.
- [HR09] H. Harbrecht and M. Randrianarivony. Wavelet BEM on molecular surfaces: Parametrization and implementation. *Computing*, 86(1):1–22, 2009.
- [HR10] H. Harbrecht and M. Randrianarivony. From computer aided design to wavelet BEM. *Computing and Visualization in Science*, 13(2):69–82, 2010.
- [HR11] H. Harbrecht and M. Randrianarivony. Wavelet BEM on molecular surfaces: Solvent excluded surfaces. *Computing*, 92(4):335–364, 2011.
- [HSS08a] H. Harbrecht, R. Schneider, and C. Schwab. Multilevel frames for sparse tensor product spaces. *Numerische Mathematik*, 110(2):199–220, 2008.

-
- [HSS08b] H. Harbrecht, R. Schneider, and C. Schwab. Sparse second moment analysis for elliptic problems in stochastic domains. *Numerische Mathematik*, 109(3):385–414, 2008.
- [HW08] G. C. Hsiao and W. L. Wendland. *Boundary Integral Equations*, volume 164 of *Applied Mathematical Sciences*. Springer, Berlin, 2008.
- [Int14] International Organization for Standardization. *ISO 10303-242:2014, Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 242: Application Protocol: Managed Model-based 3D Engineering*, 2014.
- [Iza12] M. Izadi. Parallel \mathcal{H} -matrix arithmetic on distributed-memory systems. *Computing and Visualization in Science*, 15(2):87–97, 2012.
- [KLM09] B. N. Khoromskij, A. Litvinenko, and H. G. Matthies. Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing*, 84(1-2):49–67, 2009.
- [KR83] R. V. Kadison and J. R. Ringrose. *Fundamentals of the Theory of Operator Algebras Volume I: Elementary Theory*. Academic Press, New York, 1983.
- [KR88] B. W. Kernighan and D. M. Ritchie. *The C Programming Language*. Prentice-Hall, Upper Saddle River, NJ, 1988.
- [Kr 69] P. Kr e. Les noyaux des op rateurs pseudo-diff rentiels classiques (OPDC). *Annales de l’Institut Fourier (Grenoble)*, 19(fasc. 1):179–194, 1969.
- [Kri05] R. Kriemann. Parallel \mathcal{H} -matrix arithmetics on shared memory systems. *Computing*, 74(3):273–297, 2005.
- [Kri13] R. Kriemann. \mathcal{H} -LU factorization on many-core systems. *Computing and Visualization in Science*, 16(3):105–117, 2013.
- [KS11] B. N. Khoromskij and C. Schwab. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM Journal on Scientific Computing*, 33(1):364–385, 2011.
- [Lan61] C. Lanczos. *Linear Differential Operators*. D. Van Nostrand Co. Ltd., London-Toronto-New York-Princeton, 1961.
- [LC85] W. A. Light and E. W. Cheney. *Approximation Theory in Tensor Product Spaces*. Lecture Notes in Mathematics 1169. Springer, Berlin-Heidelberg, first edition, 1985.
- [LRT79] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16(2):346–358, 1979.
- [LSY98] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *Arpack User’s Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods (Software, Environments, Tools)*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1998.
- [Mat60] B. Mat rn. Spatial Variation. *Meddelanden fr n Statens Skogsforskningsinstitut*, 49(5), 1960.
- [McL01] W. McLean. *Strongly Elliptic Systems and Boundary Integral Equations*. Cambridge University Press, Cambridge, 2001.
- [MK05] H. Matthies and A. Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 194(12-16):1295–1331, 2005.
-

- [Mol67] C. B. Moler. Iterative Refinement in Floating Point. *Journal of the ACM*, 1967.
- [MS64] N. G. Meyers and J. Serrin. $H = W$. *Proceedings of the National Academy of Sciences*, 51(6):1055–1056, 1964.
- [MZBF15] B. Marussig, J. Zechner, G. Beer, and T. Fries. Fast isogeometric boundary element method based on independent field approximation. *Computer Methods in Applied Mechanics and Engineering*, 284(0):458–488, 2015.
- [Néd82] J. C. Nédélec. Integral equations with non integrable kernels. *Integral Equations and Operator Theory*, 5(1):562–572, 1982.
- [NTW08] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2411–2442, 2008.
- [Pro95] P. Protter. *Stochastic Integration and Differential Equations: A New Approach*. Springer, Berlin-Heidelberg, 1995.
- [Rod93] L. Rodino. *Linear Partial Differential Operators in Gevrey Spaces*. World Scientific Publishing Co., Inc., River Edge, NJ, 1993.
- [RS80] M. Reed and B. Simon. *Methods of Modern Mathematical Physics, Volume 1: Functional Analysis*. Academic Press, San Diego, 1980.
- [RW05] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, 2005.
- [Saa03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, second edition, 2003.
- [Sch50] R. Schatten. *A Theory of Cross-Spaces*. Annals of Mathematics Studies 26. Princeton University Press, Princeton, 1950.
- [Si15] H. Si. TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2), 2015.
- [SS97] S. A. Sauter and C. Schwab. Quadrature for hp -Galerkin BEM in \mathbb{R}^3 . *Numerische Mathematik*, 78(2):211–258, 1997.
- [SS11] S. A. Sauter and C. Schwab. *Boundary Element Methods*. Springer, Berlin-Heidelberg, 2011.
- [ST03] C. Schwab and R. A. Todor. Sparse finite elements for elliptic problems with stochastic loading. *Numerische Mathematik*, 95(4):707–734, 2003.
- [ST06] C. Schwab and R. A. Todor. Karhunen-Loève approximation of random fields by generalized fast multipole methods. *Journal of Computational Physics*, 217(1):100–122, 2006.
- [Ste08] O. Steinbach. *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer Science & Business, New York, 2008.
- [SZ92] J. Sokolowski and J. P. Zolésio. *Introduction to Shape Optimization. Shape Sensitivity Analysis*, volume 16 of *Springer Series in Computational Mathematics*. Springer, Berlin-Heidelberg, 1992.
- [Tay81] M. E. Taylor. *Pseudodifferential Operators*, volume 34 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 1981.
- [Tyr96] E. Tyrtshnikov. Mosaic skeleton approximation. *Calcolo*, 33:47–57, 1996.

- [U.S96] U.S. Product Data Association. *Initial Graphics Exchange Specification. IGES 5.3*, 1996.
- [vPS06] T. von Petersdorff and C. Schwab. Sparse finite element methods for operator equations with stochastic data. *Applications of Mathematics*, 51(2):145–180, 2006.
- [Wil63] J. H. Wilkinson. *Rounding Errors in Algebraic Processes*. Prentice-Hall, Englewood Cliffs, 1963.
- [Wlo87] J. Wloka. *Partial Differential Equations*. Cambridge University Press, Cambridge, 1987.
- [XK02] D. Xiu and G. E. Karniadakis. Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos. *Computer Methods in Applied Mechanics and Engineering*, 191(43):4927–4948, 2002.
- [XT06] D. Xiu and D. M. Tartakovsky. Numerical methods for differential equations in random domains. *SIAM Journal on Scientific Computing*, 28(3):1167–1185, 2006.
- [ZMBF14] J. Zechner, B. Marussig, G. Beer, and T. Fries. Isogeometric boundary element method with hierarchical matrices. *Preprint*, arXiv:1406.2817, 2014.
- [Zol77] J. P. Zolésio. Multiplication dans les espaces de Besov. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 78(1-2):113–117, 1977.

Curriculum Vitae

Personal Data

Name	Jürgen Dölz
Date of birth	13 February 1990
Place of birth	Basel
Nationality	Swiss

Education

Sep 2013 – Aug 2017	PhD student in Mathematics <i>University of Basel, Basel</i>
Aug 2011 – Aug 2013	MSc student in Mathematics <i>University of Basel, Basel</i>
Aug 2008 – Jul 2011	BSc student in Mathematics <i>University of Basel, Basel</i>
Aug 2003 – Jul 2008	High School <i>Gymnasium Bäumlihof, Basel</i>
Aug 2000 – Jul 2003	Secondary School <i>Orientierungsschule Wasserstelzen, Riehen</i>
Aug 1996 – Jul 2000	Primary School <i>Primarschule Erlensträsschen, Riehen</i>