Unterschrift (Betreuer)

DIPLOMARBEIT

Isogeometrische Randelementmethode für die Lamé-Gleichung

ausgeführt am

Institut für
Analysis und Scientific Computing
Fakultät für Mathematik und Geoinformation
TU Wien

unter der Anleitung von

Univ.Prof. Dipl.-Math. Dr.techn. Dirk Praetorius und Dipl.-Ing. Dr.techn. Gregor Gantner

durch

Juliana Kainz
Matrikelnummer: 1127254
Kaiserstraße 28
1070 Wien

Wien, am 11. November 2019

Unterschrift (Student)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TECHNISCHE
UNIVERSITÄT
WIEN

DIPLOMA THESIS

Isogeometric Boundary Element Method for the Lamé-equation

written at the

Institute for
Analysis and Scientific Computing
Faculty of Mathematics and Geoinformation
Vienna University of Technology

supervised by

Univ.Prof. Dipl.-Math. Dr.techn Dirk Praetorius and
Dipl.-Ing. Dr.techn. Gregor Gantner

by

Juliana Kainz
Matriculation number: 1127254
Kaiserstraße 28
1070 Wien

Vienna, November 11, 2019

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

In dieser Arbeit betrachten wir das Dirichlet Randwertproblem für die homogene Lamé Gleichung für lineare Elastizität in zwei Dimensionen. Die Lamé Gleichung kann äquivalent als System von Randintegralgleichungen formuliert werden. Der Fokus dieser Arbeit liegt in der numerischen Approximation der Lösung dieser Randintegralgleichungen mittels isogeometrischer Randelementmethode (BEM, boundary element method). Die zentrale Idee der isogeometrischen Analysis ist, die gleichen Ansatzfunktionen für die Approximation der Lösung der Randintegralgleichungen zu verwenden, die auch für die Darstellung der Geometrie in Computer Aided Design (CAD) verwendet werden. Wir nehmen daher an, dass die Diskretisierung des Randes in Form von NURBS Funktionen gegeben ist, welche in [dB86] beschrieben werden.

Zuerst beschäftigen wir uns mit der eindeutigen Lösbarkeit der Symm'schen Integralgleichung, welche äquivalent zum Dirichlet Problem ist. Zusätzlich betrachten wir die Hypersinguläre Integralgleichung, welche äquivalent zum Neumann Problem ist.

Wir beschäftigen uns mit der numerischen Approximation der Integraloperatoren, die in der Symm'schen Integralgleichung auftreten, also die Spur des Einfachschicht- und Doppelschichtpotentialoperators. Wir verfolgen dafür die Ansätze von [Gan14] für isogeometrische BEM für die Laplace Gleichung und wenden diese auf die Lamé Gleichung an. Insbesondere sind für die Spur des Doppelschichtpotentialoperators bestimmte Integraltransformationen notwendig, da seine Darstellung nur als Cauchyscher Hauptwert eines Randintegrals existiert und nicht wie im Falle der Laplace Gleichung uneigentlich integrierbar ist. Wir wenden uniforme und adaptive Netzverfeinerung an, wobei wir für einen adaptiven Algorithmus die Ansätze von [FGHP16] verfolgen. Als a posteriori Fehlerschätzer und Verfeinerungsidikator betrachten wir den h–$h/2$–Fehlerschätzer und seine lokalen Beiträge.

Um unsere Implementierung der Operatoren zu validieren und unsere theoretischen Ergebnisse zu unterstreichen, beschäfitgen wir uns mit verschiedenen Tests und präsentieren einige numerische Beispiele.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Bibliothek Your knowledge hub
TU WIEN
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

In this work, we deal with the Dirichlet boundary value problem for the homogeneous Lamé equation from linear elasticity in two dimensions. The equation can then be equivalently reformulated as boundary integral equations. The focus of this work lies on the numerical approximation of the solution to these boundary integral equations via isogeometric BEM (boundary element method). The central idea of isogeometric analysis is to use the same ansatz functions of the approximation of the solution of the boundary integral equation as for the representation of the geometry in computer aided design (CAD). Therefore, we assume that the discretization of the boundary is given in NURBS functions which are described in [dB86].

First, we deal with the unique solvability of the Symm's integral equation, which is equivalent to the Dirichlet problem. We also consider the hypersingular integral equation, which is equivalent to the Neumann boundary value problem.

Then, we focus on the numerical approximation of the relevant integral operators occurring in the Symm's integral equation, namely the trace of the single and double layer potential. We follow the approach and results given in [Gan14] for isogeometric BEM for the Laplace equation and adapt them for the Lamé equation. However, the approximation of the trace of the double layer potential has to be treated with specific integral transformations, as its representation only exists as Cauchy principal value of a surface integral and the integral is not improperly integrable as it is the case for the Laplace equation. For mesh refinement we use uniform and adaptive refinement, where for the adaptive algorithm we follow the ideas of [FGHP16]. As an a posteriori error estimator and refinement indicator we consider the h–$h/2$–estimator and its local contributions.

Finally we validate the implementation of the operators using different tests and also present some numerical examples to underline our theoretical results.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Bibliothek, Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung

Ich möchte vor allem Herrn Prof. Dirk Praetorius danken, der mich von Beginn meines Studiums an durch viele interessante Lehrveranstaltungen für die Numerik begeistern hat können. Vielen Dank auch an Herrn Dr. techn. Gregor Gantner, welcher diese Arbeit mitbetreut hat. Beide haben sich immer Zeit genommen, mir meine Fragen zu beantworten und mir mit hilfreichen Denkanstößen weiter zu helfen.

Weiters möchte ich mich bei allen Studienkollegen und Kolleginnen, die ich im Laufe meines Studiums kennen gelernt habe und von denen einige zu guten Freunden geworden sind, bedanken! Durch die vielen interessanten sowie unterhaltsamen Gespräche habe ich die Studienzeit auch sehr genossen. Die Zeit an der TU Wien werde ich immer in besonderer Erinnerung halten.

Zuletzt möchte ich mich auch bei meinen Eltern bedanken, die es mir ermöglicht haben, hier in Wien zu studieren und immer für mich da waren und mich stets unterstützt haben.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 11. November 2019

Juliana Kainz

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

1. Introduction **1**
1.1. Physical motivation: Lamé equation 2

2. Lamé Operators **7**
2.1. Sobolev spaces 7
2.2. Variational methods 11
2.3. Weak formulation of the Lamé equation 13
    2.3.1. Solvability of the weak formulation of the Lamé equation 15
    2.3.2. Dirichlet boundary value problem 16
    2.3.3. Neumann boundary value problem 16
2.4. Fundamental solutions 18
2.5. Integral operators 19
2.6. Boundary integral equations 24
    2.6.1. Dirichlet boundary value problem 24
    2.6.2. Neumann boundary value problem 25

3. Compact presentation of BEM **27**
3.1. Galerkin method 27
    3.1.1. Dirichlet problem 28
3.2. Discretization 28
    3.2.1. B-splines and NURBS 29
    3.2.2. Boundary discretization 30
    3.2.3. Discretization space 32
3.3. Mesh-refinement and adaptive algorithm 32
3.4. The $(h\text{–}h/2)$–estimator 34

4. Numerical computation of discrete integral operators **39**
4.1. Dirichlet Problem 40
    4.1.1. Numerical computation of V_h 42
    4.1.2. Numerical computation of F_h 49
4.2. Validation of code with numerical examples 58
    4.2.1. Indirect BEM 60
    4.2.2. Validation of K 60
    4.2.3. Validation of V 62
    4.2.4. Combined validation for V and K 65

5. Numerical examples **69**
5.1. Indirect BEM with constant Dirichlet boundary data 70
5.2. Direct BEM for analytic solution 72

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

A. Implementation **77**
A.1. `Vmatrix.h` and `Vmatrix.c` 77
A.2. `Fvector.h` and `Fvector.c` 86

Bibliography **95**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

1. Introduction

In this work, we deal with the analysis and numerical solution of an equation from the area of linear elasticity theory using the isogeometric boundary element method. The Lamé equation with Dirichlet boundary data reads

$$\begin{aligned} -\mu\Delta \boldsymbol{u} - (\lambda+\mu)\nabla \mathrm{div}\ \boldsymbol{u} &= 0 \quad \text{in } \Omega && (1.1a)\\ \boldsymbol{u} &= \boldsymbol{g} \quad \text{on } \Gamma \ := \partial\Omega, && (1.1b) \end{aligned}$$

for Lamé constants $\lambda, \mu \in \mathbb{R}$ and a bounded Lipschitz domain $\Omega \subseteq \mathbb{R}^2$. In Chapter 2, we derive the weak formulation of the above partial differential equation by generally following the analysis of [Ste08] and [ME14]. To this end, we introduce the conormal derivative $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} : \Gamma \to \mathbb{R}^2$, i.e.,

$$\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} := \sigma(\boldsymbol{u})\boldsymbol{n},$$

which is composed of the matrix valued stress tensor $\sigma(\boldsymbol{u})$ and the outer unit normal vector $\boldsymbol{n}$ on Γ. We collect several important results, i.e., Betti's first and second formula and Korn's first and second inequality. Furthermore, we conclude the unique solvability of the weak formulation of the Dirichlet boundary value problem. Besides the Dirichlet boundary value problem, which is the main focus of our work, we also consider the Neumann boundary value problem

$$\begin{aligned} -\mu\Delta \boldsymbol{u} - (\lambda+\mu)\nabla \mathrm{div}\ \boldsymbol{u} &= 0 \quad \text{in } \Omega\\ \mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} &= \boldsymbol{\phi} \quad \text{on } \Gamma. \end{aligned}$$

Next, we present the fundamental solution as stated in [McL00] and investigate how the conormal derivative of the fundamental solution $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{U}$ is to be understood, since the literature considered does not give a clear interpretation for it. We deal with integral operators, namely the single and double layer potential operator as well as their trace and conormal derivative. The conormal derivative of the fundamental solution does occur in the integral kernel of the double layer integral operator and is therefore of great importance to us. In addition, we derive the boundary integral equations for the Dirichlet and Neumann boundary value problem, namely Symm's integral equation

$$V\boldsymbol{\phi} = (K + 1/2)\boldsymbol{g}$$

and the hypersingular integral equation

$$W\boldsymbol{g} = (K' - 1/2)\boldsymbol{\phi}.$$

From Chapter 3 onwards, we focus entirely on the Dirichlet boundary value problem. In that chapter, we present the Galerkin method for approximating the weak solution of the

Dirichlet boundary value problem. Furthermore, we introduce the NURBS functions and their main properties as mentioned in [dB86]. The NURBS functions form the basis of the idea of isogeometric analysis, as we assume that the boundary Γ is parametrized by these types of functions. Therefore, we consider the same functions space as the ansatz space for the Galerkin method. In addition, we follow the idea of the adaptive mesh refinement algorithm presented in [FGHP16] and then consider the $(h$–$h/2)$–estimator and its local contributions as the refinement indicators.

In Chapter, 4 we present a detailed description of the computation of the discrete boundary integral equations. In particular, we focus on the operators V and K. We follow the approach given in [Gan14] for the Laplace equation. For the occurring double integrals with integration domain Γ, we consider a partition of Γ into boundary elements and then distinguish between three different cases: separated elements, neighbouring elements and identical (overlapping) elements. In particular, for the double layer boundary integral operator K, we consider integral transformations presented in [SS11] for three dimensions in order to deal with the Cauchy principal value occurring in the representation of K. Furthermore, we present several sketches to visualize how the integral domain is transformed. We also prove that our computation of the boundary integrals using Gauß quadrature presents a reliable approximation. In Section 4.2 we present some numerical examples in order to validate the implementation in C (via MATLAB's MEX-Interface) of the integral operators V and K.

In Chapter 5, we deal with some further numerical examples on curved boundary geometries in order to underline our theoretical results. In Appendix A, we present our extension for the Lamé equation of the MATLAB-C-Implementation for the Laplace equation from [Gan14].

Throughout this work, $|\cdot|$ denotes the absolute value of scalars, the Euclidean norm of vectors, and the Hausdorff measure of a set in $\mathbb{R}^n$ for $n \geq 1$. We use the notation $A \lesssim B$ as an abbreviation for $A \leq cB$ with a generic constant $c > 0$. Furthermore, we abbreviate $A \lesssim B \lesssim A$ with $A \simeq B$.

1.1. Physical motivation: Lamé equation

The following physical motivation of the Lamé equation relies on the motivation given in [ME14, Chapter 1.1]. However, we restate it here, as we think that it gives a good introduction into the physical background.

As a model problem, we consider the Dirichlet problem

$$-\mu\Delta \boldsymbol{u} - (\lambda+\mu)\nabla \mathrm{div}\, \boldsymbol{u} = 0 \quad \text{in } \Omega \tag{1.2a}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \Gamma \tag{1.2b}$$

for a domain $\Omega \subset \mathbb{R}^n$ with $\Gamma := \partial\Omega$. The partial differential equation (1.2a) is called *Lamé-equation.* In fact, we are dealing with a linear system of n equations: the unknown $\boldsymbol{u}$ is a vector valued function $\boldsymbol{u} : \Omega \to \mathbb{R}^n$. Equation (1.2a) is also known as *Navier–Cauchy equation* or *equation of linear elasticity.* The parameters $\mu, \lambda \in \mathbb{R}$ are known as *Lamé parameters* or *Lamé constants.*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek, WIEN, Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

We aim to give an idea of the importance of problem (1.2a)–(1.2b) in real life applications. The Lamé equation essentially describes the deformation of an elastic body under the influence of different forces. The domain $\Omega \subset \mathbb{R}^n$ represents the body in its "reference configuration", i.e.: before deformation. The vector valued function $\boldsymbol{u} = (u_1, \dots, u_n) : \Omega \to \mathbb{R}^n$ describes the so-called deplacement, i.e., the deformation in relation to the reference configuration. This means that a point $x \in \Omega$ will be displaced to $x + \boldsymbol{u}(x) \in \mathbb{R}^n$.

For an arbitrary part $\Omega^* \subset \Omega$, we now consider its surface $\partial\Omega^*$. The surface tension, which is caused by the deformation of $\partial\Omega^*$, can be described by the Cauchy stress tensor $\sigma(\boldsymbol{u}) = (\sigma_{ij}(\boldsymbol{u})) : \overline{\Omega} \to \mathbb{R}^{n\times n}$. The stress tensor represents a linear mapping which associates the normal vector $\boldsymbol{n}$ at a point $x \in \partial\Omega^*$ with the stress tensor $\sigma(\boldsymbol{u}) \cdot \boldsymbol{n}$. Let us assume that there is a volume force $\boldsymbol{f} : \Omega \to \mathbb{R}^n$ acting on the body. With Newton's second Law of Motion ("force equals mass times acceleration"), we obtain that

$$\int_{\Omega^*} \rho \frac{d^2\boldsymbol{u}}{dt^2} = \int_{\partial\Omega^*} \sigma(\boldsymbol{u}) \cdot \boldsymbol{n} + \int_{\Omega^*} \boldsymbol{f}.$$

The scalar valued function ρ represents the density of the body Ω. By applying the Gauss divergence theorem and by keeping in mind that $\Omega^* \subset \Omega$ is arbitrary, it follows that

$$\rho \frac{d^2\boldsymbol{u}}{dt^2} = \operatorname{div} \sigma(\boldsymbol{u}) + \boldsymbol{f} \quad \text{in } \Omega. \tag{1.3}$$

Here, the expression $\operatorname{div} \sigma(\boldsymbol{u})$ is understood componentwise as

$$(\operatorname{div} \sigma(\boldsymbol{u}))_k = \sum_{i=1}^{n} \partial_i \sigma_{ki}(\boldsymbol{u}) \quad \text{for all } k = 1, \dots, n.$$

In the case, that the body is not in motion, the time derivative disappears. Consequently, equation (1.3) simplifies to

$$-\operatorname{div} \sigma(\boldsymbol{u}) = \boldsymbol{f} \quad \text{in } \Omega.$$

In this case, we are talking about *elastostatics*. We will focus on the particular case $\boldsymbol{f} = 0$:

$$-\operatorname{div} \sigma(\boldsymbol{u}) = 0 \quad \text{in } \Omega. \tag{1.4}$$

Physical observations show that the stress tensor $\sigma(\boldsymbol{u}(x))$ at some point $x \in \mathbb{R}$ depends locally on the length distortions of the mapping $x \mapsto x + \boldsymbol{u}(x)$. In other words, $\sigma(\boldsymbol{u}(x))$ depends on the ratio of $v^T(I_n + \nabla\boldsymbol{u})^T(I_n + \nabla\boldsymbol{u})v$ to v^Tv for $v \in \mathbb{R}^n$, where $I_n \in \mathbb{R}^{n\times n}$ represents the identity matrix and $\nabla\boldsymbol{u} = (\partial_j u_i)_{i,j=1}^n$ is the Jacobian of $\boldsymbol{u}$. The matrix

$$\frac{1}{2}((I_n + \nabla\boldsymbol{u})^T(I_n + \nabla\boldsymbol{u}) - I_n) \tag{1.5}$$

is referred to as *strain tensor*. In most of the relevant applications, one can assume that $\nabla\boldsymbol{u}$ is comparably small. Hence, the term $\nabla\boldsymbol{u}^T\nabla\boldsymbol{u}$ in (1.5) is negligible. Therefore, we can derive the *(linearised) strain tensor*

$$\varepsilon(\boldsymbol{u}) := \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T). \tag{1.6}$$

According to Hooke's Law, there is a linear relation between the stress tensor $\sigma(\boldsymbol{u})$ and the strain tensor $\varepsilon(\boldsymbol{u})$, i.e.,

$$\sigma(\boldsymbol{u}) = A\varepsilon(\boldsymbol{u}). \tag{1.7}$$

The following theory is then referred to as linear elastostatics. In the above equation (1.7), the tensor $A = (a_{ij}^{k\ell}) \in \mathbb{R}^{n\times n\times n\times n}$ is a linear mapping $\mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, which is determined by particular properties of the material the body is made of. In general, A depends on $x \in \Omega$. However, if the material is homogeneous, then A is constant.

Let us assume that the material is also isotropic, meaning uniform in all orientations. Then, in the three dimensional case, we can derive that the $3^4 = 81$ matrix entries of $A = (a_{ij}^{k\ell})$ are fully determined by the two Lamé constants $\lambda, \mu \in \mathbb{R}$. More precisely, the Lamé constants are actually defined over two other constants (see [Ste08, p.6 (1.24)]), namely the Young modulus $E > 0$ and the Poisson ratio $\nu \in (0, 1/2)$

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}.$$

Using the Lamé constants, the relation (1.7) can be written as

$$\sigma(\boldsymbol{u}) = \lambda(\operatorname{div}\, \boldsymbol{u})I_n + 2\mu\varepsilon(\boldsymbol{u}). \tag{1.8}$$

For our purposes, (1.8) will be regarded as the definition of the stress tensor for $n \geq 2$. By using (1.8), we obtain that

$$\begin{aligned}
(\operatorname{div}\, \sigma(\boldsymbol{u}))_k &= \sum_{i=1}^{n} \partial_i \sigma_{ik}(\boldsymbol{u}) \\
&\overset{(1.8)}{=} \sum_{i=1}^{n} \partial_i(\lambda(\operatorname{div}\, \boldsymbol{u})\delta_{ik} + 2\mu\varepsilon_{ik}(\boldsymbol{u})) \\
&= \lambda\partial_k(\operatorname{div}\, \boldsymbol{u}) + \mu\sum_{i=1}^{n} \partial_i(\partial_i u_k + \partial_k u_i) \\
&= \mu\sum_{i=1}^{n} \partial_i^2 u_k + (\lambda+\mu)\partial_k \sum_{i=1}^{n} \partial_i u_i \\
&= \mu\Delta u_k + (\lambda+\mu)\partial_k(\operatorname{div}\, \boldsymbol{u})
\end{aligned} \tag{1.9}$$

for all $k = 1, \dots, n$. Together with (1.4), it holds that

$$-\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla\operatorname{div}\, \boldsymbol{u} = -\operatorname{div}\, \sigma(\boldsymbol{u}) = 0 \quad \text{in } \Omega. \tag{1.10}$$

Consequently, we derive the Lamé equation (1.2a).

In conclusion, the Lamé equation (1.2a) describes at least in $n = 3$ dimensions the behaviour of sufficiently small deformations $\boldsymbol{u} : \Omega \to \mathbb{R}^n$ of a stationary, elastic body made of homogeneous, isotropic material, which fulfils all the requirements of Hooke's Law. The two dimensional case of plane elasticity can then be derived from the three dimension case (cf. [Ste08, Chapter 1.2.1]).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Furthermore, one could also consider different boundary conditions. Instead of fixing the displacement $\boldsymbol{u}$ at the boundary, one could fix the surface tension $\sigma(\boldsymbol{u}) \cdot \boldsymbol{n}$ on Γ, which leads to a Neumann problem.

For further information concerning the theory of elasticity and modelling of the Lamé equation, we refer to [Ste08, Chapter 1.2], [NH80] and [LL59].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2. Lamé Operators

2.1. Sobolev spaces

Before we start, we have to make some fundamental definitions. We consider a Lipschitz domain $\Omega \subset \mathbb{R}^n$ with boundary $\Gamma := \partial\Omega$, which means that Γ can be locally represented by the graph of a Lipschitz continuous function (cf. [McL00, Definition 3.28]). Note that by this definition Ω is open and Γ is compact. However, we do not necessarily assume that Ω is bounded or connected. Moreover, we mention that $\Omega' := \mathbb{R}^n \backslash \overline{\Omega}$ is also a Lipschitz domain.

The L^2-scalar products on Ω and Γ will be written as

$$(u, v)_\Omega := \int_\Omega u(x)v(x)dx \quad \text{and} \quad (u, v)_\Gamma := \int_\Gamma u(x)v(x)dx, \tag{2.1}$$

where the integration in the second expression is understood with respect to the surface measure. The definition of the surface measure can, e.g., be found in [Gan14, Chapter 2]. As we will be dealing with vector valued functions, we introduce the notation

$$\boldsymbol{L}^2(X) := L^2(X)^n = \{\boldsymbol{u} = (u_1, \ldots, u_n)^T : u_i \in L^2(X) \text{ for all } i = 1, \ldots, n\}$$

for $X \in \{\Omega, \Gamma\}$. The canonical scalar product on the product space $\boldsymbol{L}^2(X)$ reads

$$(\boldsymbol{u}, \boldsymbol{v})_X = \sum_{i=1}^n (u_i, v_i)_X = \int_X \boldsymbol{u} \cdot \boldsymbol{v} \; dx$$

with induced Hilbert norm $\|\cdot\|^2_{\boldsymbol{L}^2(X)} := (\cdot, \cdot)_X$. Analogeously, we define $\boldsymbol{L}^\infty(X) := L^\infty(X)^n$.

Furthermore, we use the same notation for the scalar product for matrix valued functions $\boldsymbol{U}, \boldsymbol{V} : X \to \mathbb{R}^{n\times n}$, componentwise given as $\boldsymbol{U} = (U_{ij})^n_{i,j=1}$ respective $\boldsymbol{V} = (V_{ij})^n_{i,j=1}$, i.e.,

$$(\boldsymbol{U}, \boldsymbol{V})_X := \int_X \sum_{i,j=1}^n U_{ij}V_{ij} \; dx.$$

Let $\mathcal{D}(\Omega)$ be the space of all smooth test functions with compact support, i.e.

$$\mathcal{D}(\Omega) := \{u \in C^\infty(\Omega) : \operatorname{supp} u \subseteq K, \text{ for some compact set } K \subseteq \Omega\}.$$

we introduce the *set of all distributions* $\mathcal{D}^*(\Omega)$ as the topological dual space of $\mathcal{D}(\Omega)$. An element of $\mathcal{D}^*(\Omega)$ is then referred to as *distribution*.

For the definition of Sobolev Spaces, we would like to mention that all derivatives of L^2-functions shall be understood in the weak sense. According to [McL00, Chapter 3], this means that for a function $u \in L^2(\Omega)$ and a multi-index $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0^n$ with $|\alpha| = \alpha_1 + \ldots + \alpha_n$ the "derivative" $\partial^\alpha u$ is seen as a distribution on Ω. If there exists a function $g_\alpha \in L^2(\Omega)$ such that

$$\int_\Omega g_\alpha \phi \, dx = (-1)^{|\alpha|} \int_\Omega u \partial^\alpha \phi \, dx \quad \text{for all } \phi \in \mathcal{D}(\Omega),$$

then we call g_α the *weak partial derivative* of u and denote it by $\partial^\alpha u$.

For the definitions of $H^s(\Omega), \widetilde{H}^s(\Omega), s \in \mathbb{R}$ and $H^s(\Gamma), s \in [-1, 1]$, we refer to [McL00, Chapter 3] and [Ste08, Chapter 2]. In particular, for the definition of $H^s(\Gamma)$ we also refer to [ME14, Chapter 2], as the *weak surface gradient* ∇_Γ *(defined in [ME14, Chapter 2.1.2]) and related seminorms and spaces on* Γ *are examined more closely in this work.*

In the following, we list some important special cases, which we will need in this work.

Definition 2.1.

- *For* $s = 0$ *and* $X \in \{\Omega, \Gamma\}$ *we define* $H^0(X) := L^2(X)$.
- *For* $s = 1$*, we define*

  $$H^1(\Omega) := \{u \in L^2(\Omega) : \nabla u \in \boldsymbol{L}^2(\Omega)\}$$

  and

  $$H^1(\Gamma) := \{u \in L^2(\Gamma) : \nabla_\Gamma u \in \boldsymbol{L}^2(\Gamma)\}.$$

  The Sobolev spaces $H^1(X)$ *for* $X \in \{\Omega, \Gamma\}$ *are equipped with the norm*

  $$\|u\|_{H^1(X)}^2 := \|u\|_{L^2(X)}^2 + |u|_{H^1(X)}^2,$$

  where

  $$|u|_{H^1(\Omega)}^2 := \|\nabla u\|_{\boldsymbol{L}^2(\Omega)}^2 \quad \text{and} \quad |u|_{H^1(\Gamma)}^2 := \|\nabla_\Gamma u\|_{\boldsymbol{L}^2(\Gamma)}^2 .$$

- *Let* $s \in (0, 1)$, $X \in \{\Omega, \Gamma\}$ *and*

  $$k = \begin{cases} n & \text{if } X = \Omega \\ n - 1 & \text{if } X = \Gamma. \end{cases}$$

  Then, the Sobolev space $H^s(X)$ *is defined by*

  $$H^s(X) := \{u \in L^2(X) : |u|_{H^s(X)} < \infty\} \tag{2.2}$$

  with the Slobodjeckij seminorm *given by*

  $$|u|_{H^s(X)}^2 := \int_X \int_X \frac{|u(x) - u(y)|^2}{|x - y|^{k+2s}} \, dx \, dy. \tag{2.3}$$

  Again, for $s > 0$*, we equip the spaces with the norm*

  $$\|u\|_{H^s(X)}^2 := \|u\|_{L^2(X)}^2 + |u|_{H^s(X)}^2.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- *Furthermore, we define the special case of*

$$H_0^1(\Omega) := \overline{\mathcal{D}(\Omega)}^{\|\cdot\|_{H^1(\Omega)}}$$

as the closure of the subspace $\mathcal{D}(\Omega) \subset H^1(\Omega)$ *with respect to the* $\|\cdot\|_{H^1(\Omega)}$ *norm.*

Additionally, we define Sobolev spaces with negative index, which can be characterised as the topological dual spaces.

Definition 2.2. *Let*

$$\begin{aligned} \widetilde{H}^{-s}(\Omega) &:= H^s(\Omega)^*, \qquad && \textit{for } s > 0 \\ H^{-s}(\Gamma) &:= H^s(\Gamma)^*, \qquad && \textit{for } s \in [0,1]. \end{aligned}$$

Furthermore, we would like to use the dual pairing on $H^{-1/2}(\Gamma), H^{1/2}(\Gamma)$ as extension of the L^2 scalar product. In order to justify this, we recall Gelfand triples and some related results from [SS11, Chapter 2.1.2.4].

Definition 2.3. *If* $X \leq Y$ *are real Hilbert spaces such that the identity* $I : X \to Y$ *is a continuous and dense embedding, we call* (X, Y, X^*) *a* Gelfand triple.

Lemma 2.4. *Let* (X, Y, X^*) *be a Gelfand triple. Then,* X *and* Y *are also continuously and densely embedded into* X^**.*

With the following lemma (cf. [SS11, Proposition 2.5.2]), we can apply the theory of Gelfand triples to the Hilbert spaces $H^{1/2}(\Gamma)$ and $L^2(\Gamma)$.

Lemma 2.5. *The spaces* $X := H^{1/2}(\Gamma)$*,* $Y := L^2(\Gamma)$*, and* $H^{-1/2}(\Gamma)$ *form a Gelfand triple.*

We can now extend the notation of the $L^2(\Gamma)$ scalar product to the dual pairing in $H^{1/2}(\Gamma)$. More precisely, for $u \in H^{-1/2}(\Gamma)$ and $v \in H^{1/2}(\Gamma)$, in consistency with (2.1) we denote

$$(u, v)_\Gamma := (v, u)_\Gamma := u(v).$$

Moreover, we also introduce vector valued Sobolev spaces. For $s \in \mathbb{R}$ for Ω and $s \in [-1, 1]$ for Γ, we define

$$\boldsymbol{H}^s(X) := H^s(X)^n \quad \text{for } X \in \{\Omega, \Gamma\}.$$

Then, the spaces are equipped with the canonical product space norm

$$\|\boldsymbol{u}\|^2_{\boldsymbol{H}^s(X)} := \sum_{i=1}^{n} \|u_i\|^2_{H^s(X)}.$$

Analogeously, we define $\boldsymbol{H}_0^1(\Omega)$ and the Slobdjeckij-seminorm for vector valued functions $|\cdot|_{\boldsymbol{H}^s(X)}$.

Furthermore, we introduce the trace operator and underline the importance of the space $H^{1/2}(\Gamma)$ with the following theorem from [McL00, Theorem 3.37, Theorem 3.38 and Theorem 3.40].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Theorem 2.6 (Trace theorem)**.** *We define the trace operator through*

$$\begin{aligned}\gamma : C^\infty(\overline{\Omega}) &\to C^\infty(\Gamma)\\ u &\mapsto \gamma u := u|_\Gamma.\end{aligned}$$

Then, γ has a unique linear and continuous extension

$$\gamma : H^1(\Omega) \to H^{1/2}(\Gamma)$$

which is surjective and has a continuous right inverse. Furthermore, there holds $\ker \gamma = H^1_0(\Omega)$. ☐

For vector valued functions $\boldsymbol{u} = (u_1, \dots, u_n)$, the trace operator γ shall act componentwise

$$\gamma \boldsymbol{u} := (\gamma u_1, \dots, \gamma u_n).$$

In addition, for a Lipschitz domain Ω we can define the outer normal vector $\boldsymbol{n} = (n_i)_{i=1}^n \in \mathbb{R}^n$ almost everywhere on $\Gamma = \partial\Omega$. Hence, the Gauß divergence theorem holds and provides the integration by parts formula

$$(\partial_i u, v)_\Omega = -(u, \partial_i v)_\Omega + (n_i u, v)_\Gamma \quad \text{for all } i = 1, \dots, n \text{ and } u, v \in C^1(\overline{\Omega}). \tag{2.4}$$

Since the embedding $C^1(\overline{\Omega}) \subset H^1(\Omega)$ is dense, Theorem 2.6 implies that (2.4) holds for all $u, v \in H^1(\Omega)$. As we are dealing with vector valued functions, we can generalize (2.4) to higher dimensions:

$$(\partial_i \boldsymbol{u}, \boldsymbol{v})_\Omega = -(\boldsymbol{u}, \partial_i \boldsymbol{v})_\Omega + (n_i \boldsymbol{u}, \boldsymbol{v})_\Gamma \quad \text{for all } i = 1, \dots, n \text{ and } \boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{C}^1(\overline{\Omega}),$$

with $\boldsymbol{C}^1(\overline{\Omega}) := C^1(\overline{\Omega})^n$.

With the following proposition from [Pra17, Proposition 2.8], we can easily construct equivalent norms on $H^1(\Omega)$.

Proposition 2.7. *Let $|\cdot|_{H^1}$ be a continuous seminorm on $H^1(\Omega)$ which is definite on the constant functions, i.e., $|c|_{H^1} = 0$ implies $c = 0$ for all $c \in R$. Then, there are constants $C_1, C_2 > 0$ such that*

$$|v|_{H^1} \leq C_1 \|v\|_{H^1(\Omega)} \quad \textit{as well as} \quad \|v\|_{L^2(\Omega)} \leq C_2 \underbrace{\left(\|\nabla v\|_{L^2(\Omega)} + |v|_{H^1}\right)}_{=:|\!|\!|v|\!|\!|} \textit{ for all } v \in H^1(\Omega).$$

In particular, $|\!|\!|\cdot|\!|\!|$ defines an equivalent norm on $H^1(\Omega)$, i.e.,

$$(1 + C_1)^{-1} |\!|\!|v|\!|\!| \leq \|v\|_{H^1(\Omega)} \leq (1 + C_2) |\!|\!|v|\!|\!| \textit{ for all } v \in H^1(\Omega).$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.2. Variational methods

In the later part of this work, we aim to solve operator equations of the following type: Let X be a Hilbert space with the inner product $\langle \cdot, \cdot \rangle$, $f \in X^*$ and $A : X \to X^*$ a bounded linear operator, i.e. for $C > 0$ it holds that

$$\|Av\|_{X^*} \leq C \|v\|_X \quad \text{for all } v \in X.$$

We assume that A is self-adjoint with respect to the inner product $\langle \cdot, \cdot \rangle$. For a Hilbert space X and a given $f \in X^*$, we aim to find the solution of the operator equation

$$Au = f. \tag{2.5}$$

The above equation can be formulated equivalently as a variational problem: Find $u \in X$ such that

$$\langle Au, v \rangle = \langle f, v \rangle \quad \text{for all } v \in X. \tag{2.6}$$

Every solution of the operator equation (2.5) is also a solution to the variational problem (2.6) and vice versa.

The operator $A : X \to X^*$ induces a bilinear form

$$\begin{aligned} \alpha : X \times X &\to \mathbb{R} \\ (u, v) &\mapsto \langle Au, v \rangle. \end{aligned} \tag{2.7}$$

Conversely, according to [Ste08, Lemma 3.1], also each bounded bilinear form (2.7) defines a bounded operator.

Lemma 2.8. *Let $\alpha(\cdot, \cdot) : X \times X \to \mathbb{R}$ be a bounded linear form, i.e.*

$$|\alpha(u, v)| \leq C_1 \|u\|_X \|v\|_X \quad \textit{for all } u, v \in X,$$

for some constant $C_1 > 0$. For any $u \in X$, define $Au \in X^*$ by

$$\langle Au, v \rangle = \alpha(u, v) \quad \textit{for all } v \in X.$$

Then, the induced operator $A : X \to X^*$ is linear and bounded, i.e.

$$\|Au\|_{X^*} \leq C_1 \|u\|_X \quad \textit{for all } u \in X.$$

□

With the above understanding, results proven for operators can be applied to bilinear forms and vice versa.

Furthermore, another important result that we want to mention is the well-known *Lax-Milgram lemma* [SS11, Lemma 2.1.51].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Lemma 2.9 (Lax-Milgram)**.** *Let X be a Hilbert space and $a(\cdot,\cdot)$ be a bounded and elliptic bilinear form on X, i.e.,*

$$a(u,v) \leq L \left\|u\right\|_X \left\|v\right\|_X \quad \textit{and} \quad a(u,u) \geq M \left\|u\right\|_X^2 \quad \textit{for all } u,v \in X,$$

for given $L, M > 0$. Then, for all $f \in X^*$, there exists a unique $u \in X$ such that

$$a(u,v) = f(u) \quad \textit{for all } v \in X.$$

Furthermore, it holds that

$$\left\|u\right\|_X \leq \frac{1}{M} \left\|f\right\|_{X^*}.$$

□

In the later part of this work we aim to solve an operator equation with an additional constraint. We therefore regard the following setting: Let X, Y be Hilbert spaces, $f \in X^*$ and $g \in Y^*$. For operators $A : X \to X^*$ and $B : X \to Y^*$, we want to find a solution $u \in X$ of

$$\begin{aligned} Au &= f \quad \text{in } X^* \\ Bu &= g \quad \text{in } Y^*, \end{aligned} \tag{2.8}$$

where $Au = f$ is the main equation we want to solve under the constraint $Bu = g$. Of course, in order for a solution to exist, f, g have to satisfy $g \in \text{range}(B)$ and $f \in \text{range}(A|_{V_g})$, where the manifold V_g is given through

$$V_g := \{v \in X : Bv = g\}. \tag{2.9}$$

In particular, it holds that $V_0 = \ker B$. The problem (2.8) can then be rewritten in the variational form: Find $u \in V_g$ such that

$$\langle Au, v\rangle = \langle f, v\rangle \quad \text{for all } v \in V_0.$$

The unique solvability now follows from the following result (cf. [Ste08, Theorem 3.8]).

Theorem 2.10. *Let X, Y be Hilbert spaces. Let $A : X \to X^*$ be bounded and V_0-elliptic, i.e.*

$$\langle Av, v\rangle \geq C_A \left\|v\right\|_X^2 \quad \textit{for all } v \in V_0 := \ker B,$$

where $B : X \to Y^*$. Then, for $f \in$ range$(A|_{V_g})$ and $g \in$ range(B) there exists a unique solution $u \in X$ of the operator equation (2.8).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.3. Weak formulation of the Lamé equation

Let us consider the *Lamé operator* from Section 1.1

$$\mathcal{L}\boldsymbol{u} = -\mathrm{div}\ \sigma(\boldsymbol{u}) = -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla\mathrm{div}\ \boldsymbol{u}, \tag{2.10}$$

which is a linear differential operator of second order. We use the strain tensor from (1.6)

$$\varepsilon(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T), \tag{2.11}$$

componentwise given as

$$\varepsilon_{ij}(\boldsymbol{u}) = \frac{1}{2}\left(\partial_i u_j + \partial_j u_i\right) \quad \text{for } i,j = 1,\dots,n, \tag{2.12}$$

and the stress tensor from (1.8)

$$\sigma(\boldsymbol{u}) = \lambda(\mathrm{div}\ \boldsymbol{u})I_n + 2\mu\varepsilon(\boldsymbol{u}), \tag{2.13}$$

componentwise given as

$$\sigma_{ij}(\boldsymbol{u}) = \lambda\delta_{ij}\sum_{k=1}^{n}\partial_k u_k + 2\mu\varepsilon_{ij}(\boldsymbol{u}) \quad \text{for } i,j = 1,\dots,n. \tag{2.14}$$

We can then write (2.10) componentwise as

$$(\mathcal{L}\boldsymbol{u})_k = -\mathrm{div}\ (\sigma(\boldsymbol{u}))_k = -\sum_{i=1}^{n}\partial_i\sigma_{ki}(\boldsymbol{u}). \tag{2.15}$$

In this work, we are looking at the following equation for $\boldsymbol{u} \in \boldsymbol{H}^2(\Omega)$

$$\mathcal{L}\boldsymbol{u} = \boldsymbol{0}. \tag{2.16}$$

By using the componentwise representation of $\mathcal{L}$ from (2.15), multiplying with a test function $\boldsymbol{v} = (v_1,\dots,v_n)^T \in \boldsymbol{H}^2(\Omega)$, integrating over Ω and applying integration by parts, we obtain that

$$\begin{aligned} 0 &= -\int_\Omega \sum_{j=1}^{n}\partial_j\sigma_{ij}(\boldsymbol{u})v_i\ dx \\ &= \int_\Omega \sum_{j=1}^{n}\sigma_{ij}(\boldsymbol{u})\partial_j v_i\ dx - \int_\Gamma \sum_{j=1}^{n} n_j\sigma_{ij}(\boldsymbol{u})v_i\ dx \end{aligned} \tag{2.17}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

for $i = 1, \dots, n$, where the last integral over Γ occurs with respect to the surface measure. We now define the so-called *induced bilinear form* of $\mathcal{L}$

$$\begin{aligned}\alpha(\boldsymbol{u}, \boldsymbol{v}) &:= \int_\Omega \sum_{i,j=1}^n \sigma_{ij}(\boldsymbol{u}) \partial_j v_i \; dx \\ &= \int_\Omega \sum_{i,j=1}^n \sigma_{ij}(\boldsymbol{u}) \frac{1}{2} \left(\partial_j v_i + \partial_i v_j\right) \; dx \\ &= \int_\Omega \sum_{i,j=1}^n \sigma_{ij}(\boldsymbol{u}) \varepsilon_{ij}(\boldsymbol{v}) \; dx \\ &= (\sigma(\boldsymbol{u}), \varepsilon(\boldsymbol{v}))_\Omega,\end{aligned}$$

where the second equality holds due to symmetry of $\sigma(\boldsymbol{u})$. Using the componentwise representation (2.14) of $\sigma(\boldsymbol{u})$, we can rewrite $\alpha(\boldsymbol{u}, \boldsymbol{v})$ as

$$\begin{aligned}\alpha(\boldsymbol{u}, \boldsymbol{v}) &= \lambda \int_\Omega \sum_{i,j=1}^n \delta_{ij} \varepsilon_{ij}(\boldsymbol{v}) \sum_{k=1}^n \partial_k u_k \; dx + 2\mu \int_\Omega \sum_{i,j=1}^n \varepsilon_{ij}(\boldsymbol{u}) \varepsilon_{ij}(\boldsymbol{v}) \; dx \\ &= \lambda \int_\Omega \operatorname{div} \boldsymbol{v} \operatorname{div} \boldsymbol{u} \; dx + 2\mu \int_\Omega \sum_{i,j=1}^n \varepsilon_{ij}(\boldsymbol{u}) \varepsilon_{ij}(\boldsymbol{v}) \; dx \\ &= \lambda (\operatorname{div} \boldsymbol{u}, \operatorname{div} \boldsymbol{v})_\Omega + 2\mu (\varepsilon(\boldsymbol{u}), \varepsilon(\boldsymbol{v}))_\Omega.\end{aligned}$$

From the above representation, we can now conclude the symmetry of the bilinear form $\alpha(\cdot, \cdot)$.

In addition, we introduce the *conormal derivative* $\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u} : \Gamma \to \mathbb{R}^n$ as

$$(\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})_i := \sum_{j=1}^n \sigma_{ij}(\boldsymbol{u}) n_j \quad \text{for } i = 1, \dots, n. \tag{2.18}$$

If we now sum over all components in (2.17), we obtain that

$$-\int_\Omega \sum_{i,j=1}^n \partial_j \sigma_{ij}(\boldsymbol{u}) v_i \; dx = -(\operatorname{div} \sigma(\boldsymbol{u}), \boldsymbol{v})_\Omega = (\mathcal{L}\boldsymbol{u}, \boldsymbol{v})_\Omega = \alpha(\boldsymbol{u}, \boldsymbol{v}) - (\gamma \boldsymbol{v}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})_\Gamma, \tag{2.19}$$

which is called *Betti's first formula* in [Ste08, Chapter 1.2]. Betti's first formula is a special case of the first Green identity (see for instance in [ME14, Lemma 3.1, (3.7)]) for the Lamé equation.

Due to the fact that $\alpha(\boldsymbol{u}, \boldsymbol{v}) = \alpha(\boldsymbol{v}, \boldsymbol{u})$ and by using Betti's first formula from (2.19), we obtain *Betti's second formula*

$$-(\operatorname{div} \sigma(\boldsymbol{u}), \boldsymbol{v})_\Omega + (\gamma \boldsymbol{v}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})_\Gamma = -(\operatorname{div} \sigma(\boldsymbol{v}), \boldsymbol{u})_\Omega + (\gamma \boldsymbol{u}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{v})_\Gamma. \tag{2.20}$$

Again, Betti's second formula is a special case of the *second Green identity* (see for instance [ME14, Lemma 3.2]).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

2.3.1. Solvability of the weak formulation of the Lamé equation

First, we rewrite Betti's first formula to

$$\alpha(\boldsymbol{u},\boldsymbol{v}) = (\mathcal{L}\boldsymbol{u},\boldsymbol{v})_\Omega + (\gamma\boldsymbol{v},\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u})_\Gamma.$$

As we want to apply the theory of Section 2.2 in order to get unique solvability, we need to show that the bilinear form $\alpha(\cdot,\cdot)$ is bounded and elliptic. It is shown in [Ste08, Lemma 4.13], that $\alpha(\cdot,\cdot)$ is bounded.

Lemma 2.11. *For all* $\boldsymbol{u},\boldsymbol{v} \in \boldsymbol{H}^1(\Omega)$*, it holds that*

$$|\alpha(\boldsymbol{u},\boldsymbol{v})| \leq L|\boldsymbol{u}|_{\boldsymbol{H}^1(\Omega)}|\boldsymbol{v}|_{\boldsymbol{H}^1(\Omega)},$$

where the constant $L > 0$ *depends only on* $E, \nu > 0$ *from section 1.1.* ☐

Note that the above lemma also holds with the seminorm $|\cdot|_{\boldsymbol{H}^1(\Omega)}$ replaced by the full norm $\|\cdot\|_{\boldsymbol{H}^1(\Omega)}$.

In order to show the $\boldsymbol{H}_0^1(\Omega)$-ellipticity of the bilinear form $\alpha(\cdot,\cdot)$, we require several steps from [Ste08, Chpater 4.2].

Lemma 2.12. *For* $\boldsymbol{v} \in \boldsymbol{H}^1(\Omega)$ *we have*

$$\alpha(\boldsymbol{v},\boldsymbol{v}) \geq C(\varepsilon(\boldsymbol{v}),\varepsilon(\boldsymbol{v}))_\Omega,$$

where the constant $C > 0$ *depends only on* $E, \nu > 0$ *from Section 1.1.* ☐

Next, we can formulate *Korn's first inequality.*

Lemma 2.13 (Korn's first inequality)**.** *For all* $\boldsymbol{v} \in \boldsymbol{H}_0^1(\Omega)$*, it holds that*

$$(\varepsilon(\boldsymbol{v}),\varepsilon(\boldsymbol{v}))_\Omega \geq \frac{1}{2}|\boldsymbol{v}|^2_{\boldsymbol{H}^1(\Omega)}.$$

☐

We can now conclude the $\boldsymbol{H}_0^1(\Omega)$-ellipticity of the bilinear form $\alpha(\cdot,\cdot)$. The following result is proven by using equivalent norms on $\boldsymbol{H}^1(\Omega)$.

Corollary 2.14. *For all* $\boldsymbol{v} \in \boldsymbol{H}_0^1(\Omega)$*, it holds that*

$$\alpha(\boldsymbol{v},\boldsymbol{v}) \geq C\,\|\boldsymbol{v}\|_{\boldsymbol{H}^1(\Omega)}\,,$$

where the constant $C > 0$ *depends only on* E, ν *from section 1.1 and a norm equivalence constant.* ☐

According [Ste08, Chpater 4.2], we can extend the bilinear form $\alpha(\cdot,\cdot)$ by some L^2 norm to obtain an equivalent norm in $\boldsymbol{H}^1(\Omega)$. This is a direct consequence of *Korn's second inequality*, or sometimes just simply referred to as *Korn's inequality* in literature.

Theorem 2.15 (Korn's second inequality)**.** *Let* $\Omega \subseteq \mathbb{R}^n$ *be a bounded domain with piecewise smooth boundary* $\Gamma = \partial\Omega$*. Then we have*

$$(\varepsilon(\boldsymbol{v}),\varepsilon(\boldsymbol{v}))_\Omega + \|\boldsymbol{v}\|^2_{\boldsymbol{L}^2(\Omega)} \geq C\,\|\boldsymbol{v}\|^2_{\boldsymbol{H}^1(\Omega)} \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^1(\Omega).$$

for a constant $C > 0$. ☐

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.3.2. Dirichlet boundary value problem

For mechanical applications, the boundary Γ is often divided into parts, where different boundary conditions hold, i.e., $\Gamma = \overline{\Gamma}_D \cup \overline{\Gamma}_N$ for Dirichlet and Neumann boundary conditions respectively. Given $\boldsymbol{g} \in \boldsymbol{H}^{1/2}(\Gamma)$, we consider the homogeneous Dirichlet boundary value problem

$$\begin{aligned} -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla\text{div } \boldsymbol{u} &= \boldsymbol{0} \quad \text{in } \Omega, \\ \gamma\boldsymbol{u} &= \boldsymbol{g} \quad \text{on } \Gamma. \end{aligned} \tag{2.21}$$

Note that the first equation holds in $\boldsymbol{H}^{-1}(\Omega)$. We define a solution manifold in $\boldsymbol{H}^1(\Omega)$

$$V_g := \left\{\boldsymbol{v} \in \boldsymbol{H}^1(\Omega) : \gamma\boldsymbol{v} = \boldsymbol{g}\right\},$$

where $V_0 = \boldsymbol{H}_0^1(\Omega)$. We have to find $\boldsymbol{u} \in V_g$ such that

$$\alpha(\boldsymbol{u}, \boldsymbol{v}) = 0 \quad \text{for all } \boldsymbol{v} \in V_0. \tag{2.22}$$

Since the bilinear form $\alpha(\cdot,\cdot)$ is bounded (cf. Lemma 2.11) and $\boldsymbol{H}_0^1(\Omega)$-elliptic (cf. Corollary 2.14) we conclude the unique solvability of (2.22) by applying Theorem 2.10. In addition, the unique solution of (2.22) satisfies that

$$\|\boldsymbol{u}\|_{\boldsymbol{H}^1(\Omega)} \leq C\,\|\boldsymbol{g}\|_{H^{1/2}(\Gamma)},$$

for some $C > 0$.

2.3.3. Neumann boundary value problem

We first consider the case of the homogeneous Neumann boundary value problem with homogeneous boundary conditions

$$\begin{aligned} -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla\text{div } \boldsymbol{u} &= \boldsymbol{0} \quad \text{in } \Omega, \\ \mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} &= \boldsymbol{0} \quad \text{on } \Gamma. \end{aligned} \tag{2.23}$$

The non-trivial solutions of the above boundary value problem are given by the *rigid body motions* $\boldsymbol{r} \in \mathcal{R}$, where

$$\mathcal{R} = \text{span}\left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}-x_2\\x_1\end{pmatrix}\right\} \tag{2.24}$$

for $n = 2$. We can see that $\varepsilon(\boldsymbol{r}) = \boldsymbol{0}$ and $\Delta\boldsymbol{r} = 0 = \text{div } \boldsymbol{r}$ for all $\boldsymbol{r} \in \mathcal{R}$. Consequently also $\sigma(\boldsymbol{r}) = \boldsymbol{0}$ and therefore also $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{r} = 0$ for all $\boldsymbol{r} \in \mathcal{R}$. As a conclusion we see that the rigid body motions do indeed solve the homogeneous Neumann boundary value problem.

In order to justify the inverse statement, namely that all solutions of (2.23) are given by the rigid body motions, we have to add the condition that Ω is connected. The proof for the following lemma is found in [McL00, Lemma 10.5] for $n = 3$ but can analogously be done for $n = 2$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Lemma 2.16. *Let $n = 2$ and $\Omega \subset \mathbb{R}^n$ be open and connected. Then, for all distributions $\boldsymbol{r} \in \mathcal{D}^*(\Omega)$, $\varepsilon(\boldsymbol{r}) = 0$ in Ω already implies that $\boldsymbol{r} \in \mathcal{R}$.* ☐

If we now insert the rigid body motions $\boldsymbol{r} \in \mathcal{R}$ into Betti's second formula (2.20), we get the following orthogonality

$$-(\operatorname{div} \sigma(\boldsymbol{u}), \boldsymbol{r})_\Omega + (\gamma \boldsymbol{r}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})_\Gamma = 0 \tag{2.25}$$

for all $\boldsymbol{r} \in \mathcal{R}$. Next, we consider a homogeneous Neumann boundary value problem with inhomogeneous boundary conditions

$$\begin{aligned} -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla\operatorname{div}\boldsymbol{u} &= \boldsymbol{0} \quad \text{in } \Omega, \\ \mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} &= \boldsymbol{\phi} \quad \text{on } \Gamma, \end{aligned} \tag{2.26}$$

with $\boldsymbol{\phi} \in \boldsymbol{H}^{1/2}(\Gamma)$. Due to the orthogonality (2.25) we have to assume the *solvability condition*

$$(\gamma\boldsymbol{r}, \boldsymbol{\phi})_\Gamma = 0 \quad \text{for all } \boldsymbol{r} \in \mathcal{R}, \tag{2.27}$$

in order for a solution to exist. Note the solution of the Neumann boundary value problem is only unique up to the rigid body motions according to [Ste08, p.8]. In order to fix the rigid body motions, we can formulate appropriate scaling conditions. To this end, we define

$$\boldsymbol{H}^1_*(\Omega) = \left\{\boldsymbol{v} \in \boldsymbol{H}^1(\Omega) : (\boldsymbol{r}, \boldsymbol{v})_\Omega = 0 \quad \text{for all } \boldsymbol{r} \in \mathcal{R}\right\}.$$

Then, the weak formulation of the Neumann boundary value problem (2.26) reads: Find $\boldsymbol{u} \in \boldsymbol{H}^1_*(\Omega)$ such that

$$\alpha(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{\phi}, \gamma\boldsymbol{v})_\Gamma \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^1_*(\Omega). \tag{2.28}$$

Using Theorem 2.7, we can introduce an equivalent norm in $\boldsymbol{H}^1(\Omega)$ as stated in the following corollary from [Ste08, Corollary 4.18].

Corollary 2.17. *For the space of all rigid body motions $\mathcal{R} = span\{\boldsymbol{r}_k\}_{k=1}^{\dim\mathcal{R}}$ with the basis $(\boldsymbol{r}_k)_{k=1}^{\dim\mathcal{R}}$ as in* (2.24)*,*

$$\|\boldsymbol{v}\|_{\boldsymbol{H}^1(\Omega),\Gamma} := \left(\sum_{k=1}^{\dim\mathcal{R}} (\boldsymbol{r}_k, \boldsymbol{v})_\Omega^2 + (\varepsilon(\boldsymbol{v}), \varepsilon(\boldsymbol{v}))_\Omega\right)^{1/2}.$$

defines an equivalent norm on $\boldsymbol{H}^1(\Omega)$. ☐

With the above Corollary 2.17 and Lemma 2.12 we can establish the $\boldsymbol{H}^1_*(\Omega)$-ellipticity of the bilinear form $\alpha(\cdot,\cdot)$ and hence get unique solvability of the variational problem (2.28).

In addition, for a weak solution $\boldsymbol{u} \in \boldsymbol{H}^1_*(\Omega)$ we can define other solutions to the Neumann boundary problem just by adding a linear combination of rigid body motions

$$\widetilde{\boldsymbol{u}} := \boldsymbol{u} + \sum_{k=1}^{\dim\mathcal{R}} a_k \boldsymbol{r}_k \in \boldsymbol{H}^1(\Omega),$$

for $a_k \in \mathbb{R}$ and $\boldsymbol{r}_k$ as in Corollary 2.17.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.4. Fundamental solutions

Next, we consider the Lamé operator $\boldsymbol{\mathcal{L}u}$ from (2.10) and Betti's second formula (2.20). Since this work considers only $\boldsymbol{\mathcal{L}u} = \boldsymbol{0}$, Betti's second formula simplifies to

$$(\boldsymbol{\mathcal{L}v}, \boldsymbol{u})_\Omega = -(\operatorname{div} \sigma(\boldsymbol{v}), \boldsymbol{u})_\Omega = (\gamma \boldsymbol{v}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})_\Gamma - (\gamma \boldsymbol{u}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{v})_\Gamma .$$

Let us assume that for every $x \in \Omega$ and for every component $k = 1, 2$ there exists a function $\boldsymbol{v}(\cdot) := \boldsymbol{U}_k^*(x, \cdot)$ which satisfies

$$((\boldsymbol{\mathcal{L}} \boldsymbol{U}_k^*)(x, \cdot), \boldsymbol{u})_\Omega = u_k(x).$$

Then, the solution $\boldsymbol{u} = (u_1, \dots, u_n)^T$ of $\boldsymbol{\mathcal{L}u} = \boldsymbol{0}$ is given by the *representation formula* for $x \in \Omega$

$$\begin{aligned} u_k(x) &= (\boldsymbol{U}_k^*(x, \cdot), \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})_\Gamma - (\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{U}_k^*(x, \cdot), \gamma \boldsymbol{u})_\Gamma \\ &= \int_\Gamma \boldsymbol{U}_k^*(x, y) \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u}(y) \; dy - \int_\Gamma \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{U}_k^*(x, y) \gamma \boldsymbol{u}(y) \; dy, \end{aligned} \tag{2.29}$$

for $k = 1, 2$, where the integration occurs with respect to the surface measure. Therefore, it is enough to know the *Cauchy data* $(\gamma \boldsymbol{u}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})$ on Γ in order to compute a solution to the equation $\boldsymbol{\mathcal{L}u} = \boldsymbol{0}$. Since

$$u_k(x) = \int_\Omega \delta_0(x - y) u_k(y) \; dy \quad \text{for } x \in \Omega,$$

where δ_0 is the Dirac delta, we have to solve the partial differential equation for $j, k = 1, 2$

$$\begin{aligned} (\mathcal{L}_y \boldsymbol{U}_k^*)_j(x, y) &= \delta_0(y - x), \qquad && \text{for } j = k \\ (\mathcal{L}_y \boldsymbol{U}_k^*)_j(x, y) &= 0, && \text{for } j \neq k \end{aligned} \tag{2.30}$$

for $x, y \in \mathbb{R}^2$ in the distributional sense. Note that with the notation $\mathcal{L}_y$, we want to imply that the operator acts only on the y component of the function. A solution $\boldsymbol{U}^* = (\boldsymbol{U}_1^*, \boldsymbol{U}_2^*)$ to the equation (2.30) is called *fundamental solution*. In order to compute a solution $\boldsymbol{u}$ to the Lamé equation via the representation formula (2.29), we need the existence of a fundamental solution $\boldsymbol{U}^*(x, y)$. Using the representation formula, we can formulate appropriate boundary integral equations to find the complete Cauchy data on the boundary. According to [Ste08, p. 90], the existence of a fundamental solution is ensured for the Lamé equation.

In [McL00, Theorem 10.4] the fundamental solution for the Lamé equation in two dimensions is given.

Theorem 2.18 (Fundamental solution of the Lamé equation)**.** *If* $\mu \neq 0$ *and* $2\mu + \lambda \neq 0$, *then a fundamental solution* $\boldsymbol{U}^*(x, y) := \boldsymbol{U}^*(x - y)$ *for the Lamé operator* (2.10) *in* $n = 2$ *dimensions is given by*

$$\boldsymbol{U}^*(z) := \frac{1}{4\pi\mu(2\mu + \lambda)} \left(-(3\mu + \lambda) \log |z| I_2 + (\mu + \lambda) \frac{z z^T}{|z|^2} \right),$$

where $I_2 \in \mathbb{R}^{2\times 2}$ *is the identity matrix. Note that* $\boldsymbol{U}^* = (U_{ij}^*)_{i,j=1}^2$ *is a matrix valued function and consists of two separate vector valued functions* $\boldsymbol{U}^* = (\boldsymbol{U}_1^*, \boldsymbol{U}_2^*)$. ☐

For further details on the computation of the fundamental solution we refer to [ME14, section 3.2] and [Ste08, section 5.2].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.5. Integral operators

As we will be considering the Dirichlet boundary value problem (2.21) and the Neumann boundary value problem (2.26), we have to derive appropriate boundary integral equations in order to find the complete Cauchy data $(\gamma \boldsymbol{u}, \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})$ on Γ. By inserting the Cauchy data in the representation formula (2.29), we then have a representation of the solution. We consider the surface potentials that occur in the representation formula including their mapping properties. The following approach is called *direct approach* in literature.

For $x \in \Omega \cup \Omega^C$, the *single layer potential*

$$\mathcal{V} : \boldsymbol{H}^{-1/2}(\Gamma) \to \boldsymbol{H}^1(\Omega).$$

for a function $\boldsymbol{w} \in \boldsymbol{H}^{-1/2}(\Gamma)$ is given by

$$(\mathcal{V}\boldsymbol{w})_k(x) := \int_\Gamma (\boldsymbol{U}^*(x,y)\boldsymbol{w}(y))_k \, dy$$

for $k = 1, \dots, n$, where $\boldsymbol{U}^*(x,y)\boldsymbol{w}(y)$ is a matrix vector multiplication and we integrate the k-th entry of the resulting vector over Γ.

Remark 2.19. *Note that the single layer potential $\mathcal{V}\boldsymbol{w}$ for any $\boldsymbol{w} \in H^{-1/2}(\Gamma)$ is a solution to $\mathcal{L}(\mathcal{V}\boldsymbol{w}) = \boldsymbol{0}$ according to [SS11, Proposition 3.4.1].*

Before we introduce the double layer potential, we will take a closer look at the conormal derivative of the fundamental solution. We will denote derivatives ∂_k with respect to y by $\partial_{k,y}$. Similarly, $\mathfrak{d}_{\boldsymbol{n},y}$ means that the occurring derivatives act only on the variable y and the normal vector $\boldsymbol{n} = (n_1, n_2)^T$ is in point y. Since $\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u}$ is defined in (2.18) only for vector valued functions $\boldsymbol{u}$, we will explain how $\mathfrak{d}_{\boldsymbol{n},y} \boldsymbol{U}^*$ is to be interpreted. By comparing with [Ste08, Section 5.2, (5.11)], we can see that the fundamental solution is to be understood row-wise and therefore, for $\boldsymbol{U}^* = \left(U^*_{ij}\right)_{i,j=1}^2$ there holds

$$\mathfrak{d}_{\boldsymbol{n},y} \boldsymbol{U}^* = \begin{pmatrix} \mathfrak{d}_{\boldsymbol{n},y} (U^*_{11}, U^*_{12})^T \\ \mathfrak{d}_{\boldsymbol{n},y} (U^*_{21}, U^*_{22})^T \end{pmatrix},$$

where $\mathfrak{d}_{\boldsymbol{n},y}(U^*_{i1}, U^*_{i2})$ is a column vector for $i = 1, 2$. Using (2.14) and (2.18), we have for vector valued $\boldsymbol{u} = (u_1, u_2)^T$

$$(\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u})_i = \sum_{j=1}^2 \sigma_{ij}(\boldsymbol{u}) n_j = \sum_{j=1}^2 \left(\lambda \delta_{ij} \left(\sum_{k=1}^2 \partial_k u_k \right) + \mu(\partial_j u_i + \partial_i u_j) \right) n_j, \quad \text{for } i = 1, 2.$$

For the fundamental solution $\boldsymbol{U}^*$, we thus see that

$$\begin{aligned} (\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{U}^*)_{kp} &= \sum_{q=1}^2 \sigma_{kpq}(\boldsymbol{U}^*) n_q \\ &:= \sum_{q=1}^2 \left(\lambda \delta_{pq} \left(\sum_{r=1}^2 \partial_{r,y} U^*_{kr} \right) + \mu(\partial_{p,y} U^*_{kq} + \partial_{q,y} U^*_{kp}) \right) n_q, \end{aligned} \tag{2.31}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

for $k, p = 1, 2$. Note that $\sigma_{kpq}(\boldsymbol{U}^*)$ is a 3-dimensional tensor and that $\mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(x,y)$ is a matrix. We write $(\mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(x,y)\boldsymbol{v}(y))_k$ for the k-th entry of the vector $\mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(x,y)\boldsymbol{v}(y)$.

Then, for $x \in \Omega \cup \Omega^C$, the *double layer potential*

$$\mathcal{K} : \boldsymbol{H}^{1/2}(\Gamma) \to \boldsymbol{H}^1(\Omega)$$

of function $\boldsymbol{v} \in \boldsymbol{H}^{1/2}(\Gamma)$ is given by

$$(\mathcal{K}\boldsymbol{v})_k(x) := \int_\Gamma (\mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(x,y)\boldsymbol{v}(y))_k \, dy \quad \text{for } k = 1, \dots, n.$$

Using the introduced surface integral potentials, the representation formula (2.29) can be written as

$$\boldsymbol{u} = \mathcal{V}(\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u}) - \mathcal{K}(\gamma\boldsymbol{u}). \tag{2.32}$$

According to [ME14, Corollary 3.12] and [McL00, Theorem 6.11] the operators $\mathcal{V}$ and $\mathcal{K}$ are continuous for bounded Ω.

In order to obtain the complete Cauchy data, we have to consider the trace γ and the conormal derivative $\mathfrak{d}_{\boldsymbol{n}}$ of the operators $\mathcal{V}$ and $\mathcal{K}$. First, we consider the single layer potential. According to [Ste08, Section 6.7], the trace of $\mathcal{V}$ defines a bounded linear operator

$$V := \gamma\mathcal{V} : \boldsymbol{H}^{-1/2}(\Gamma) \to \boldsymbol{H}^{1/2}(\Gamma).$$

The operator V has a representation as a weakly singular surface integral for $\boldsymbol{w} \in \boldsymbol{L}^\infty(\Gamma)$

$$(V\boldsymbol{w})_i(x) = \int_\Gamma \sum_{j=1}^n U^*_{ij}(x,y) w_j(y) \, dy \quad \text{for } x \in \Gamma \text{ and } i = 1, \dots, n, \tag{2.33}$$

as stated in [Ste08, (6.58)] and [SS11, (3.32)]. As in [SS11, Remark 5.1.7], this means that (2.33) exists as an improper integral. According to [Ste08, p.158/Section 6.7] we can assume for simplicity that $x \in \Gamma$ is on a smooth part of the boundary. In particular, for $n = 2$ this means that we exclude the cases, where $x \in \Gamma$ is a corner point.

Next, we consider the trace of the operator $\mathcal{K}$. To this end, we define

$$K := \left(\frac{1}{2}I + \gamma\mathcal{K}\right) : \boldsymbol{H}^{1/2}(\Gamma) \to \boldsymbol{H}^{1/2}(\Gamma),$$

where I is the identity operator. Then, it holds that

$$\gamma\mathcal{K} = \left(K - \frac{1}{2}I\right) : \boldsymbol{H}^{1/2}(\Gamma) \to \boldsymbol{H}^{1/2}(\Gamma).$$

Furthermore, K is linear and has the following representation

$$(K\boldsymbol{v})_i(x) := \lim_{\varepsilon\to 0} \int_{y\in\Gamma : |y-x| \geq \varepsilon} \sum_{j=1}^n (\mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(x,y))_{ij} v_j(y) \, dy \quad \text{for } v_j \in \boldsymbol{L}^\infty(\Gamma), \tag{2.34}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

for $i = 1, \dots, n$. Note that the above integral is a Cauchy principal value and not necessarily improperly integrable. The above representation holds for all points $x \in \Gamma$, where Γ is differentiable (cf. [McL00, Theorem 7.4]). According to [McL00, Theorem 6.11] the operator $K : \boldsymbol{H}^{1/2}(\Gamma) \to \boldsymbol{H}^{1/2}(\Gamma)$ is also bounded.

Later, we will also need the adjoint operator K' of K,

$$K' : \boldsymbol{H}^{-1/2}(\Gamma) \to \boldsymbol{H}^{-1/2}(\Gamma).$$

According to [Ste08, Section 6.7], the operator K' has a representation as a Cauchy principal value of a surface integral

$$(K'\boldsymbol{w})_k(x) = \lim_{\varepsilon \to 0} \int_{y \in \Gamma : |y-x| \geq \varepsilon} \sum_{j=1}^{n} \sum_{\ell=1}^{n} \sigma_{k\ell}(\boldsymbol{U}_j^*(\cdot, y))(x) n_\ell(x) w_j(y) \, dy \quad \text{for } k = 1, \dots, n.$$

Next, we consider the conormal derivative (2.18), or alternatively referred to as *interior boundary stress operator*, of the operators $\mathcal{V}$ and $\mathcal{K}$. The operator

$$\mathfrak{d}_{\boldsymbol{n}}\mathcal{V} : \boldsymbol{H}^{-1/2}(\Gamma) \to \boldsymbol{H}^{-1/2}(\Gamma)$$

is linear and bounded (cf. [Ste08, Section 6.7]) and can be represented by

$$(\mathfrak{d}_{\boldsymbol{n}}\mathcal{V}\boldsymbol{w})_i(x) = \lim_{\Omega \ni \widetilde{x} \to x \in \Gamma} \sum_{j=1}^{n} \sigma_{ij}(\mathcal{V}\boldsymbol{w})(\widetilde{x}) n_j(\widetilde{x}) \quad \text{for } i = 1, \dots, n.$$

An alternative representation is given by

$$\mathfrak{d}_{\boldsymbol{n}}\mathcal{V} = \frac{1}{2} I + K'.$$

By applying the interior boundary stress operator $\mathfrak{d}_{\boldsymbol{n}}$ on the double layer potential $\mathcal{K}$ we obtain a bounded linear operator

$$\mathfrak{d}_{\boldsymbol{n}}\mathcal{K} : \boldsymbol{H}^{1/2}(\Gamma) \to \boldsymbol{H}^{-1/2}(\Gamma),$$

according to [Ste08, Section 6.7, p. 163]. We define the *hypersingular boundary integral operator* by $W := -\mathfrak{d}_{\boldsymbol{n}}\mathcal{K}$.

Remark 2.20 (Symmetry of W). *For E resp. ν being the Young modulus and Poisson ratio from Section 1.1, let*

$$G_{ij}(x,y) := \frac{1}{4\pi} \frac{E}{1-\nu^2} \left(-\log|x-y| \, \delta_{ij} + \frac{(x_i - y_i)(x_j - y_j)}{|x-y|^2} \right) \quad \textit{for } i,j = 1,2.$$

According to [Ste08, p.163] for $\boldsymbol{C}(\Gamma) := C(\Gamma) \times C(\Gamma)$ and $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{H}^{1/2}(\Gamma) \cap \boldsymbol{C}(\Gamma)$, we obtain for the hypersingular boundary integral operator that

$$(W\boldsymbol{u}, \boldsymbol{v})_\Gamma = \sum_{i,j=1}^{2} \int_\Gamma d_x v_j(x) \int_\Gamma G_{ij}(x,y) d_y u_i(y) \, dy \, dx,$$

where d_x, d_y denotes the derivative with respect to the arclength. Since $\boldsymbol{C}(\Gamma) \cap \boldsymbol{H}^{1/2}(\Gamma)$ is dense in $\boldsymbol{H}^{1/2}(\Gamma)$, we therefore obtain that W is symmetric on $\boldsymbol{H}^{1/2}(\Gamma)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

In order to gain a better overview, we summarize the relations between the potential $\mathcal{V}$ and $\mathcal{K}$ and the different boundary integrals V, K, K', W. There holds

$$\begin{aligned} \gamma\mathcal{V} &= V, & \gamma\mathcal{K} &= K - \frac{1}{2}I, \\ \mathfrak{d}_{\boldsymbol{n}}\mathcal{V} &= K' + \frac{1}{2}I, & \mathfrak{d}_{\boldsymbol{n}}\mathcal{K} &= -W. \end{aligned} \tag{2.35}$$

When considering the representation formula (2.29) on Γ, we can separately apply γ and $\mathfrak{d}_{\boldsymbol{n}}$. Using the identities from (2.35), we then obtain that

$$\begin{pmatrix} \gamma\boldsymbol{u} \\ \mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}I - K & V \\ W & \frac{1}{2}I + K' \end{pmatrix} \begin{pmatrix} \gamma\boldsymbol{u} \\ \mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} \end{pmatrix}, \tag{2.36}$$

where the involved operator matrix

$$\mathcal{C} := \begin{pmatrix} \frac{1}{2}I - K & V \\ W & \frac{1}{2}I + K' \end{pmatrix}$$

is called *Calderón projector*. According to [Ste08, p. 163] the Calderón projector defines a projector fulfilling $\mathcal{C} = \mathcal{C}^2$. When rearranging the first row of equation (2.36), we obtain the following boundary integral equation

$$V\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} = \left(\frac{1}{2}I + K\right)\gamma\boldsymbol{u} \quad \text{in } \boldsymbol{H}^{1/2}(\Gamma), \tag{2.37}$$

which is called *Symm's integral equation*. The second row of equation (2.36) simplifies to

$$W\gamma\boldsymbol{u} = \left(\frac{1}{2}I - K'\right)\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} \quad \text{in } \boldsymbol{H}^{-1/2}(\Gamma) \tag{2.38}$$

and is called the *hypersingular integral equation*.

In addition, we consider the rigid body motions from (2.24). Since $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{r} = 0$ for all $\boldsymbol{r} \in \mathcal{R}$, inserting $\boldsymbol{r} \in \mathcal{R}$ into the representation formula from (2.32), we get that

$$\boldsymbol{r} = -\mathcal{K}\boldsymbol{r} \quad \text{on } \Omega.$$

Moreover, considering equation (2.37) and (2.38) gives

$$\left(\frac{1}{2}I + K\right)\gamma\boldsymbol{r} = \boldsymbol{0} \tag{2.39}$$

and

$$W\gamma\boldsymbol{r} = \boldsymbol{0}. \tag{2.40}$$

As we later want to solve (2.37) for $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u}$, we need the ellipticity of the operator V. We define for $n = 2$

$$\boldsymbol{H}_{+}^{-1/2}(\Gamma) := \left\{\boldsymbol{w} \in \boldsymbol{H}^{-1/2}(\Gamma) : (w_i, 1)_\Gamma = 0 \quad \text{for } i = 1, 2\right\}.$$

According to [Ste08, Theorem 6.36], we then get ellipticity in $\boldsymbol{H}_{+}^{-1/2}(\Gamma)$ for $n = 2$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Theorem 2.21. *It holds that Then we have*

$$(V\boldsymbol{w},\boldsymbol{w})_\Gamma \geq C_V \|\boldsymbol{w}\|^2_{\boldsymbol{H}^{-1/2}(\Gamma)}, \quad \textit{for all } \boldsymbol{w} \in \boldsymbol{H}_+^{-1/2}(\Gamma).$$

where $C_V > 0$ depends only on Γ. □

In [Ste08, Section 6.7], the ellipticity of V in $\boldsymbol{H}^{-1/2}(\Gamma)$ is shown, which requires some more steps and suitable scaling of the domain Ω. In the following theorem from [ME14, Theorem 3.18] a shorter explanation is given.

Theorem 2.22. *For a Lipschitz domain $\Omega \subset \mathbb{R}^2$, we may either consider $h\Omega$ for a sufficiently small scaling parameter $h > 0$, or, equivalently, we may consider Ω and the scaled fundamental solution $\boldsymbol{U}_c(z) = \boldsymbol{U}(z) + cI$ with sufficiently large $c > 0$. Then, we obtain the ellipticity of V in $\boldsymbol{H}^{-1/2}(\Gamma)$, i.e.,*

$$(V\boldsymbol{w},\boldsymbol{w})_\Gamma \geq \widetilde{C}_V \|\boldsymbol{w}\|^2_{\boldsymbol{H}^{-1/2}(\Gamma)} \quad \textit{for all } \boldsymbol{w} \in \boldsymbol{H}^{-1/2}(\Gamma).$$

In particular, V is an isomorphism with $\|V^{-1}\| \leq \widetilde{C}_V^{-1}$ and

$$(\boldsymbol{w},\boldsymbol{v})_V := (V\boldsymbol{w},\boldsymbol{v})_\Gamma \quad \textit{for all } \boldsymbol{v},\boldsymbol{w} \in \boldsymbol{H}^{-1/2}(\Gamma)$$

define an equivalent scalar product on $\boldsymbol{H}^{-1/2}(\Gamma)$.

By applying the Lax-Milgram lemma (Lemma 2.9) we then get existence of the inverse operator $V^{-1} : \boldsymbol{H}^{1/2}(\Gamma) \to \boldsymbol{H}^{-1/2}(\Gamma)$. We define

$$\boldsymbol{H}_*^{-1/2}(\Gamma) := \left\{\boldsymbol{w} \in \boldsymbol{H}^{-1/2}(\Gamma) : (\boldsymbol{w},\boldsymbol{r})_\Gamma = 0 \quad \text{for all } \boldsymbol{r} \in \mathcal{R}\right\},$$

and

$$\boldsymbol{H}_*^{1/2}(\Gamma) := \left\{\boldsymbol{v} \in \boldsymbol{H}^{1/2}(\Gamma) : (V^{-1}\boldsymbol{v},\boldsymbol{r})_\Gamma = 0 \quad \text{for all } \boldsymbol{r} \in \mathcal{R}\right\}.$$

Then, $V : \boldsymbol{H}_*^{-1/2}(\Gamma) \to \boldsymbol{H}_*^{1/2}(\Gamma)$ is an isomorphism.

Furthermore, the single layer operator V is symmetric as stated in the following corollary.

Corollary 2.23. *The single layer operator $V \in L(\boldsymbol{H}^{-1/2}(\Gamma); \boldsymbol{H}^{1/2}(\Gamma)$ is a symmetric operator, i.e. $(V\boldsymbol{\phi},\boldsymbol{\psi})_\Gamma = (\boldsymbol{\phi}, V\boldsymbol{\psi})_\Gamma$ for all $\boldsymbol{\phi},\boldsymbol{\psi} \in \boldsymbol{H}^{-1/2}(\Gamma)$.*

For the Laplace operator the above corollary is proven in [Pra07, Corollary 4.24]. The proof for Corollary 2.23 can be done in the same way for the Lamé operator, since the fundamental solution for the Lamé equation $\boldsymbol{U}^*(x-y) = \boldsymbol{U}^*(y-x)$ from Theorem 2.18 is symmetric.

As for V, we also need an ellipticity result for the hypersingular boundary integral operator W. According to [Ste08, p. 164], W is $\boldsymbol{H}_*^{1/2}(\Gamma)$-ellipticity, i.e.,

$$(W\boldsymbol{v},\boldsymbol{v})_\Gamma \geq C_W \|\boldsymbol{v}\|^2_{\boldsymbol{H}^{1/2}(\Gamma)} \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}_*^{1/2}(\Gamma),$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

for some constant $C_W > 0$ which only depends on Γ. Moreover, we introduce

$$\boldsymbol{H}^{1/2}_{**}(\Gamma) := \Big\{ \boldsymbol{v} \in \boldsymbol{H}^{1/2}(\Gamma) : (\boldsymbol{v}, \boldsymbol{r})_\Gamma = 0 \quad \text{for all } \boldsymbol{r} \in \mathcal{R} \Big\},$$

which essentially equips the space $\boldsymbol{H}^{1/2}(\Gamma)$ with the solvability condition (2.27). Then, there exits a constant $\widetilde{C}_W > 0$, which depends only on Γ, such that

$$(W\boldsymbol{v}, \boldsymbol{v})_\Gamma \geq \widetilde{C}_W \, \|\boldsymbol{v}\|^2_{\boldsymbol{H}^{1/2}(\Gamma)} \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^{1/2}_{**}(\Gamma); \tag{2.41}$$

see [Ste08, p. 165].

2.6. Boundary integral equations

In this section, we consider boundary value problems for $\mathcal{L}\boldsymbol{u} = \boldsymbol{0}$ for a bounded, simple connected domain Ω with Lipschitz boundary $\Gamma := \partial\Omega$.

2.6.1. Dirichlet boundary value problem

First, we consider the Dirichlet boundary value problem. For given $\boldsymbol{g} \in \boldsymbol{H}^{1/2}(\Gamma)$, it reads

$$\begin{aligned} -\mu\Delta\boldsymbol{u} - (\lambda + \mu)\nabla \mathrm{div}\, \boldsymbol{u} &= \boldsymbol{0} \quad \text{in } \Omega, \\ \gamma\boldsymbol{u} &= \boldsymbol{g} \quad \text{on } \Gamma. \end{aligned}$$

When looking at the representation formula (2.32), we still have to find the unknown Neumann data $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} \in \boldsymbol{H}^{-1/2}(\Gamma)$. By inserting into Symm's integral equation (2.37), we obtain

$$V\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} = \left(\frac{1}{2}I + K\right)\boldsymbol{g}. \tag{2.42}$$

The above equation is in fact equivalent to the Dirichlet problem, because for each solution $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u}$ we can construct a solution $\boldsymbol{u} \in \boldsymbol{H}^1(\Omega)$ to the Dirichlet problem via the representation formula (2.32). On the other hand, for a solution $\boldsymbol{u} \in \boldsymbol{H}^1(\Omega)$, the trace $\gamma\boldsymbol{u} = \boldsymbol{g}$ and conormal derivative $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u}$ fulfil equation (2.42).

According to [Ste08, p.173] we may also consider (2.42) in the equivalent variational formulation: Find $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} \in \boldsymbol{H}^{-1/2}(\Gamma)$ such that

$$(V\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u}, \boldsymbol{v})_\Gamma = ((\frac{1}{2}I + K)\boldsymbol{g}, \boldsymbol{v})_\Gamma \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^{-1/2}(\Gamma). \tag{2.43}$$

As V is bounded and $\boldsymbol{H}^{-1/2}(\Gamma)$-elliptic according to Theorem 2.22 when assuming a suitable scaling of domain Ω, we conclude the unique solveability of the boundary integral equation (2.43) by applying the Lax-Milgram lemma (Lemma 2.9). Moreover, since V, V^{-1}, K are bounded, it holds that

$$\|\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u}\|_{\boldsymbol{H}^{-1/2}(\Gamma)} \leq \frac{1}{C_V} \left\| \left(\frac{1}{2}I + K\right)\boldsymbol{g} \right\|_{\boldsymbol{H}^{1/2}(\Gamma)} \leq \frac{C_K}{C_V} \|\boldsymbol{g}\|_{\boldsymbol{H}^{1/2}(\Gamma)},$$

where $C_V, C_K > 0$ are the boundedness constants for V and K. The above inequalities are stated in [Ste08, p. 172] for the Laplace equation, but still hold for the Lamé equation as the operators remain bounded.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.6.2. Neumann boundary value problem

Next, we consider the Neumann boundary value problem. For given $\boldsymbol{\phi} \in \boldsymbol{H}^{-1/2}(\Gamma)$, it reads

$$\begin{aligned} -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla \operatorname{div} \boldsymbol{u} &= \boldsymbol{0} \quad \text{in } \Omega, \\ \mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} &= \boldsymbol{\phi} \quad \text{on } \Gamma. \end{aligned}$$

In order for a solution to exist, we have to assume the solvability condition (2.27)

$$(\gamma\boldsymbol{r}, \boldsymbol{\phi})_\Gamma = 0 \quad \text{for all } \boldsymbol{r} \in \mathcal{R}.$$

Note that the solution of the Neumann boundary value problem is only unique up to the rigid body motions. When considering the representation formula (2.32), we still have to find the unknown Dirichlet data $\gamma\boldsymbol{u} \in \boldsymbol{H}^{1/2}(\Gamma)$. By inserting into the hypersingular integral equation (2.38), we obtain

$$W\gamma\boldsymbol{u} = \left(\frac{1}{2}I - K'\right)\boldsymbol{\phi}. \tag{2.44}$$

The above equation is equivalent to the Neumann problem, since for each solution $\gamma\boldsymbol{u}$ of (2.44), we can construct a solution $\boldsymbol{u} \in \boldsymbol{H}^1(\Omega)$ to the Neumann problem via the represenation formula (2.32). On the other hand, for a solution $\boldsymbol{u} \in \boldsymbol{H}^1(\Omega)$ the trace $\gamma\boldsymbol{u}$ and conormal derivative $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u} = \boldsymbol{\phi}$ fulfil equation (2.44).

As $W\gamma\boldsymbol{r} = \boldsymbol{0}$ for all rigid body motions $\boldsymbol{r} \in \mathcal{R}$, we have $\ker W = \gamma\mathcal{R}$. In order to ensure the solvability of (2.44), similar as done in [Ste08, Section 7.2] for the Laplace equation, we need to assume the following solvability condition

$$(\frac{1}{2}I - K')\boldsymbol{\phi} \in \operatorname{range}(W) = (\ker W)^0, \tag{2.45}$$

where

$$(\ker W)^0 := \{f \in H^{-1/2}(\Gamma) : \langle f, v\rangle = 0 \text{ for all } v \in \ker W\} \subseteq H^{-1/2}(\Gamma)$$

and the last equality in (2.45) results from the closed range theorem (cf. [Pra17, Theorem 5.7]) and Remark 2.20. Instead of (2.44), we can consider the equivalent variational formulation: find $\gamma\boldsymbol{u} \in \boldsymbol{H}^{1/2}(\Gamma)$

$$(W\gamma\boldsymbol{u}, \boldsymbol{v})_\Gamma = ((\frac{1}{2}I - K')\boldsymbol{\phi}, \boldsymbol{v})_\Gamma \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^{1/2}(\Gamma). \tag{2.46}$$

As the operator W is bounded and $\boldsymbol{H}^{1/2}_{**}(\Gamma)$-elliptic due to (2.41) we obtain that the variational formulation

$$(W\gamma\boldsymbol{u}, \boldsymbol{v})_\Gamma = ((\frac{1}{2}I - K')\boldsymbol{\phi}, \boldsymbol{v})_\Gamma \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^{1/2}_{**}(\Gamma), \tag{2.47}$$

has a unique solution $\gamma\boldsymbol{u} \in \boldsymbol{H}^{1/2}_{**}(\Gamma)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Remark 2.24. *Since we cannot prove the ellipticity of operator W in $H^{1/2}(\Gamma)$ we may consider a modified formulation of the problem: find $\gamma\boldsymbol{u} \in \boldsymbol{H}^{1/2}(\Gamma)$ such that*

$$(\hat{W}\gamma\boldsymbol{u}, \boldsymbol{v})_\Gamma = ((\frac{1}{2}I - K')\boldsymbol{\phi}, \boldsymbol{v})_\Gamma \quad \textit{for all } \boldsymbol{v} \in \boldsymbol{H}^{1/2}(\Gamma), \tag{2.48}$$

where $(\boldsymbol{r}_i)_{i=1}^{\dim\mathcal{R}}$ is a basis of $\mathcal{R}$. The modified hypersingular integral operator $\hat{W} : \boldsymbol{H}^{1/2}(\Gamma) \to \boldsymbol{H}^{-1/2}(\Gamma)$ is given by

$$(\hat{W}\boldsymbol{w}, \boldsymbol{v})_\Gamma := (W\boldsymbol{w}, \boldsymbol{v})_\Gamma + \sum_{i=1}^{\dim\mathcal{R}} (\boldsymbol{w}, \gamma\boldsymbol{r}_i)_\Gamma (\boldsymbol{v}, \gamma\boldsymbol{r}_i)_\Gamma \quad \textit{for } \boldsymbol{v}, \boldsymbol{w} \in \boldsymbol{H}^{1/2}(\Gamma).$$

As W is bounded, also $\hat{W}$ is bounded.

A proof that the problem formulations (2.47) and (2.48) are equivalent and that the operator $\hat{W}$ is $\boldsymbol{H}^{1/2}(\Gamma)$–elliptic, remains open.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3. Compact presentation of BEM

As the boundary integral equations of the previous section can in general not be solved analytically, in this chapter we will present the Galerkin boundary element method and related theory in order to solve these boundary integral equations numerically. From this chapter on, we will only consider the Dirichlet problem.

3.1. Galerkin method

We consider a Hilbert space H with norm $\|\cdot\|_H$ and a continuous and elliptic bilinear form $\langle\!\langle\cdot,\cdot\rangle\!\rangle$ on H. For a given $F \in H^*$, we then get existence and uniqueness of the solution $u \in H$ of

$$\langle\!\langle u, v\rangle\!\rangle = F(v) \quad \text{for all } v \in H, \tag{3.1}$$

by applying the Lax-Milgram lemma (Lemma 2.9). The Galerkin scheme then consists of replacing the continuous space H by a finite dimensional and hence closed subspace X_ℓ of H. Since the Lax-Milgram lemma applies as well to the subspace X_ℓ, there also exists a unique Galerkin solution $u_\ell \in X_\ell$ of

$$\langle\!\langle u_\ell, v_\ell\rangle\!\rangle = F(v_\ell) \quad \text{for all } v_\ell \in X_\ell.$$

The solution fulfils an important property called the *Galerkin orthogonality*

$$\langle\!\langle u - u_\ell, v_\ell\rangle\!\rangle = 0 \quad \text{for all } v_\ell \in X_\ell. \tag{3.2}$$

The projection $\mathbb{G}_\ell : H \to X_\ell$ characterised by

$$\langle\!\langle \mathbb{G}_\ell u, v_\ell\rangle\!\rangle = \langle\!\langle u, v_\ell\rangle\!\rangle \quad \text{for all } v_\ell \in X_\ell,$$

is called *Galerkin projection*. Due to the Galerkin orthogonality (3.2) the Galerkin projection is an orthogonal projection. If $\langle\!\langle\cdot,\cdot\rangle\!\rangle$ is additionally symmetric, then it is a scalar product. Hence, we call the scalar product $\langle\!\langle\cdot,\cdot\rangle\!\rangle$ and the induced norm $|\!|\!|\cdot|\!|\!| := \langle\!\langle\cdot,\cdot\rangle\!\rangle^{1/2}$ the *energy scalar product* and *energy norm* respectively.

With the Pythagoras theorem, we have that

$$|\!|\!|u - \mathbb{G}_\ell u|\!|\!|^2 + |\!|\!|\mathbb{G}_\ell u - v_\ell|\!|\!|^2 = |\!|\!|u - v_\ell|\!|\!|^2 \quad \text{for all } v_\ell \in X_\ell.$$

This implies that $\mathbb{G}_\ell u$ is the best approximation of u in X_ℓ with respect to the energy norm $|\!|\!|\cdot|\!|\!|$, i.e.

$$|\!|\!|u - \mathbb{G}_\ell u|\!|\!| \le |\!|\!|u - v_\ell|\!|\!| \quad \text{for all } v_\ell \in X_\ell.$$

By switching to an equivalent norm in H, we obtain the *Céa lemma* [Pra07, Lemma 5.3]

Lemma 3.1 (Céa-Lemma)**.** *The Galerkin error is quasi-optimal, i.e.*

$$\|u - \mathbb{G}_\ell u\|_H \leq C_{Céa} \min_{v_\ell \in X_\ell} \|u - v_\ell\|_H \quad \textit{for all } u \in H,$$

where the constant $C_{Céa} > 0$ *depends only on the ellipticity and continuity of* $\langle\!\langle \cdot, \cdot \rangle\!\rangle$. *If* $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ *is additionally symmetric, it holds that*

$$|\!|\!| u - \mathbb{G}_\ell u |\!|\!| = \min_{v_\ell \in X_\ell} |\!|\!| u - v_\ell |\!|\!| \quad \textit{for all } u \in H.$$

The Galerkin method then consists of constructing a sequence of finite dimensional subspaces X_ℓ of H with $X_\ell \subseteq X_{\ell+1}$.

3.1.1. Dirichlet problem

We consider the Dirichlet problem (2.21) with the variational formulation of Symm's integral equation from (2.43):

$$(V \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u}, \boldsymbol{v})_\Gamma = ((\frac{1}{2} I + K)\boldsymbol{g}, \boldsymbol{v})_\Gamma \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^{-1/2}(\Gamma), \tag{3.3}$$

where we seek $\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u} \in \boldsymbol{H}^{-1/2}(\Gamma)$ for given Dirichlet data $\boldsymbol{g} \in \boldsymbol{H}^{1/2}(\Gamma)$. For simplicity we set $\boldsymbol{f} := (\frac{1}{2} I + K)\boldsymbol{g} \in \boldsymbol{H}^{1/2}(\Gamma)$, $\boldsymbol{\phi} := \mathfrak{d}_{\boldsymbol{n}} \boldsymbol{u}$ and $(\boldsymbol{\phi}, \boldsymbol{v})_V := (V\boldsymbol{\phi}, \boldsymbol{v})_\Gamma$. The above formulation (3.3) can then be rewritten as: find $\boldsymbol{\phi} \in \boldsymbol{H}^{-1/2}(\Gamma)$ such that

$$(\boldsymbol{\phi}, \boldsymbol{v})_V = (\boldsymbol{f}, \boldsymbol{v})_\Gamma \quad \text{for all } \boldsymbol{v} \in \boldsymbol{H}^{-1/2}(\Gamma).$$

The right–hand side defines a continuous linear form for $\boldsymbol{v}$. The left hand side $(\boldsymbol{\phi}, \boldsymbol{v})_V$ defines a symmetric bounded bilinear form, which is under certain assumptions on the scaling of the domain Ω also elliptic (see Theorem 2.22). Hence, $(\cdot, \cdot)_V$ defines an equivalent scalar product on $\boldsymbol{H}^{-1/2}(\Gamma)$. We can therefore apply the Galerkin method and replace $\boldsymbol{H}^{-1/2}(\Gamma)$ by a suitable finite dimensional subspace X_ℓ. Then, our problem (3.3) reads: find $\Phi_\ell \in X_\ell$, such that

$$(\Phi_\ell, \Psi_\ell)_V = (\boldsymbol{f}, \Psi_\ell)_\Gamma \quad \text{for all } \Psi_\ell \in X_\ell. \tag{3.4}$$

The seized solution Φ_ℓ is then the Galerkin projection onto X_ℓ of the exact solution $\boldsymbol{\phi} \in \boldsymbol{H}^{-1/2}(\Gamma)$ with respect to the energy scalar product $\langle\!\langle \cdot, \cdot \rangle\!\rangle := (\cdot, \cdot)_V$. Hence there holds the Galerkin orthogonality and we get that Φ_ℓ is the best approximation of $\boldsymbol{\phi}$ in X_ℓ.

3.2. Discretization

In this section, we discuss the discretization of the boundary Γ as well as the ansatz spaces X_ℓ. Before we give the boundary discretization, we introduce the so–called NURBS functions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.2.1. B-splines and NURBS

In this section we aim to introduce B-splines as well as NURBS and mention some basic properties.

We consider a sequence of *knots* $\check{\mathcal{K}} := (t_i)_{i\in\mathbb{Z}}$ on $\mathbb{R}$ with $t_{i-1} \leq t_i$ for $i \in \mathbb{Z}$ and $\lim_{i\to\pm\infty} t_i = \pm\infty$. For the multiplicity of any knot t_i we write $\#t_i := j \in \mathbb{Z} : t_i = t_j$. We denote the corresponding set of *nodes* $\check{\mathcal{N}} := \{t_i : i \in \mathbb{Z}\} = \{\check{z}_j : j \in \mathbb{Z}\}$ with $\check{z}_{j-1} < \check{z}_j$ for $j \in \mathbb{Z}$. For $i \in \mathbb{Z}$, the i-th *B-spline* of degree p is defined inductively by

$$\begin{aligned} B_{i,0} &:= \chi_{[t_{i-1},t_i)}, \\ B_{i,p} &:= \beta_{i-1,p} B_{i,p-1} + (1-\beta_{i,p}) B_{i+1,p-1} \quad \text{for } p \in \mathbb{N}, \end{aligned}$$

where, for $t \in \mathbb{R}$,

$$\beta_{i,p}(t) := \begin{cases} \frac{t-t_i}{t_{i+p}-t_i} & \text{if } t_i \neq t_{i+p}, \\ 0 & \text{if } t_i = t_{i+p}. \end{cases}$$

We may also use the notations $B^{\check{\mathcal{K}}}_{i,p} := B_{i,p}$ and $\beta^{\check{\mathcal{K}}}_{i,p} := \beta_{i,p}$ in order to stress the dependence on the knots $\check{\mathcal{K}}$.

The following properties of B-splines are taken from [dB86]. We refer to [dB86, Section 2-4,6] for the proof.

Lemma 3.2. *For $p \in \mathbb{N}_0$ and a finite interval $I = [a,b)$, the following assertions hold:*

(i) *For $i \in \mathbb{Z}$ and $\ell \in \mathbb{Z}$, $B_{i,p}|_{[t_{\ell-1},t_\ell)}$ is a polynomial of degree p.*

(ii) *For $i \in \mathbb{Z}$, $B_{i,p}$ vanishes outside the interval $[t_{i-1}, t_{i+p})$ and is positive on the open interval (t_{i-1}, t_{i+p}).*

(iii) *For $i \in \mathbb{Z}$, $B_{i,p}$ is completely determined by the $p+2$ knots $t_{i-1}, \dots, t_{i+p}$.*

(iv) *The B-splines of degree p form a (locally finite) partition of unity, i.e.*

$$\sum_{i\in\mathbb{Z}} B_{i,p} = 1 \quad \text{on } \mathbb{R}.$$

(v) *The set $\{B_{i,p}|_I : i \in \mathbb{Z} \text{ with } B_{i,p}|_I \neq 0\}$ is a basis for the space of all right-continuous $\check{\mathcal{N}}$-piecewise polynomials of degree lower or equal to p on I with break points $\check{\mathcal{N}} \cap (a,b)$ and which are, at each break point t_i, $p - \#t_i$ times continuously differentiable if $p - \#t_i \geq 0$.* □

In addition to the knots $\check{\mathcal{K}}$, we consider a sequence of fixed positive weights $\mathcal{W} := (w_i)_{i\in\mathbb{Z}}$ with $w_i > 0$. Then, we define the corresponding NURBS functions.

Definition 3.3. *For $i \in \mathbb{Z}$ and $p \in \mathbb{N}_0$, we define the i-th* non-uniform rational B-spline *of degree p, or shortly* NURBS, *as*

$$R_{i,p} := \frac{w_i B_{i,p}}{\sum_{\ell\in\mathbb{Z}} w_\ell B_{\ell,p}}. \tag{3.5}$$

Note that the denominator is never zero due to Lemma 3.2 (ii) *and* (iv)*. We also use the notation $R^{\check{\mathcal{K}},\mathcal{W}}_{i,p} := R_{i,p}$.*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

Furthermore, we define for $p \in \mathbb{N}_0$ the B-spline space

$$\mathscr{S}^p(\check{\mathcal{K}}) := \left\{ \sum_{i\in\mathbb{Z}} a_i B_{i,p}^{\check{\mathcal{K}}} : a_i \in \mathbb{R} \right\}$$

as well as the NURBS space

$$\mathscr{N}^p(\check{\mathcal{K}}, \mathcal{W}) := \left\{ \sum_{i\in\mathbb{Z}} a_i R_{i,p}^{\check{\mathcal{K}},\mathcal{W}} : a_i \in \mathbb{R} \right\} = \frac{\mathscr{S}^p(\check{\mathcal{K}})}{\sum_{\ell\in\mathbb{Z}} w_\ell B_{\ell,p}^{\check{\mathcal{K}},\mathcal{W}}}.$$

We also define the NURBS space on Γ

$$\hat{\mathscr{N}}^p(\check{\mathcal{K}}, \mathcal{W}) := \mathscr{N}^p(\check{\mathcal{K}}, \mathcal{W})|_{[a,b)} \circ \gamma|_{[a,b)}^{-1}.$$

Note that with Lemma 2.5, it holds that

$$\hat{\mathscr{N}}^p(\check{\mathcal{K}}, \mathcal{W}) \subset L^2(\Gamma) \subset H^{-1/2}(\Gamma). \tag{3.6}$$

The following result is from [dB86, corollary 2] and gives us nestedness of the B-spline spaces, if we refine the knots.

Corollary 3.4. *Let $p \in \mathbb{N}_0$. For a refinement $\check{\mathcal{K}}'$ of $\check{\mathcal{K}}$, i.e., $\check{\mathcal{K}} = (t_i)_{i\in\mathbb{Z}}$ is a subsequence of $\check{\mathcal{K}}' = (t_i)'_{i\in\mathbb{Z}}$, it holds that*

$$\mathscr{S}^p(\check{\mathcal{K}}) \subseteq \mathscr{S}^p(\check{\mathcal{K}}').$$

3.2.2. Boundary discretization

We assume that $\Omega \subseteq \mathbb{R}^2$ is a bounded Lipschitz domain whose boundary $\Gamma := \partial\Omega$ can be parametrised by a fixed regular closed curve $\gamma : [a, b] \to \Gamma$. Moreover, we demand that γ is continuous, piecewise continuously differentiable and that $\gamma|_{[a,b)}$ is bijective. Furthermore, we make the assumption that γ is positively orientated. We also assume that for the left and right derivative of γ, there holds $\gamma'^{\ell}(t) \neq 0$ and $\gamma'^{r}(t) \neq 0$ for $t \in [a, b)$. Furthermore, we demand that

$$\gamma'^{\ell}(t) + c\gamma'^{r}(t) \neq 0 \quad \text{for all } c > 0 \text{ and } t \in [a, b].$$

For the discretization, we introduce some further notation.

- **Nodes**
  Let $\check{\mathcal{N}}_\star := \{\check{z}_j \in [a, b] : j = 0, \ldots, n\}$ be a set of nodes with $a = \check{z}_0 < \check{z}_1 < \ldots < \check{z}_n = b$ and such that $\gamma|_{[\check{z}_{j-1},\check{z}_j]} \in C^1([\check{z}_{j-1}, \check{z}_j])$. The corresponding nodes on Γ are then given by $\mathcal{N}_\star := \{z_j := \gamma(\check{z}_j) : j = 1, \ldots, n\}$, where $z_0 := z_n$.

- **Multiplicity and knots**
  Let $p \in \mathbb{N}_0$ be some fixed polynomial order. Each node $z_j \in \mathcal{N}_\star$ has a fixed multiplicity $\#z_j \in \{1, 2, \ldots, p+1\}$ with $\#z_0 = \#z_n = p+1$ and $\#z_j \leq p+1$ for $j = 1, \ldots, n-1$. This induces knots

  $$\mathcal{K}_\star = (\underbrace{z_1, \ldots, z_1}_{\#z_1-\text{times}}, \ldots, \underbrace{z_n, \ldots, z_n}_{\#z_n-\text{times}}),$$

  with corresponding knots $\check{\mathcal{K}}_\star := \gamma|_{(a,b]}^{-1}(\mathcal{K}_\star)$ on the parameter domain $(a, b]$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- **Elements and partitions**
  Let $\mathcal{T}_\star = \{T_1, \dots, T_n\}$ be a partition of Γ into compact and connected segments $T_j = \gamma(\check{T}_j)$ with $\check{T}_j = [\check{z}_{j-1}, \check{z}_j]$. Then, we define

$$[\mathcal{T}_\star] := \{[T] : T \in \mathcal{T}_\star\} \quad \text{with} \quad [T] := (T, \#z_{T,1}, \#z_{T,2}),$$

  where $z_{T,1} = z_{j-1}$ and $z_{T,2} = z_j$ are the two nodes of $T = T_j$. We will refer to $\mathcal{T}_\star$ as mesh and denote the set of all meshes of Γ by $\mathbb{T}$.

- **Local mesh-sizes**
  Let $h_{\star,T}$ denote the arclength of each element $T \in \mathcal{T}_\star$. Then, we can define the local mesh-width function $h_\star \in L^\infty(\Gamma)$ by $h_\star|_T = h_{\star,T}$.
  In addition, for each element $T \in \mathcal{T}_\star$, we define its length by $\check{h}_{\star,T} := |\gamma^{-1}(T)|$ with respect to the parameter domain $[a,b]$. Again, we can define a global function $\check{h}_\star \in L^\infty(\Gamma)$ with $\check{h}_\star|_T := \check{h}_{\star,T}$.
  Note that the lengths of $h_{\star,T}$ and $\check{h}_{\star,T}$ of an element T are equivalent and the equivalence constants depend only on γ.

- **Local mesh-ratio (shape regularity constant)**
  The *shape regularity constants* of the mesh on $[a,b]$ and Γ are given by

$$\begin{aligned}\check{\kappa}(\mathcal{T}_\star) &:= \max\{\check{h}_{\star,T}/\check{h}_{\star,T'} : T, T' \in \mathcal{T}_\star \text{ with } T \cap T' \neq \emptyset\},\\ \kappa(\mathcal{T}_\star) &:= \max\{h_{\star,T}/h_{\star,T'} : T, T' \in \mathcal{T}_\star \text{ with } T \cap T' \neq \emptyset\}.\end{aligned}$$

  Note that $\kappa(\mathcal{T}_\star) \simeq \check{\kappa}(\mathcal{T}_\star)$, where the hidden constants depend only on the parametrization γ.

- **Patches**
  For each set $\Gamma_0 \subseteq \Gamma$ and $m \in \mathbb{N}_0$, we define the patch inductively

$$\omega_\star^m(\Gamma_0) := \begin{cases} \Gamma_0 & \text{if } m = 0,\\ \omega_\star(\Gamma_0) := \bigcup\{T \in \mathcal{T}_\star : T \cap \Gamma_0 \neq \emptyset\} & \text{if } m = 1,\\ \omega_\star(\omega_\star^{m-1}(\Gamma_0)) & \text{if } m > 1.\end{cases}$$

  For nodes $z \in \Gamma$, we abbreviate $\omega_\star(z) := \omega_\star(\{z\})$ for the node patch. Analogously, for each set $\mathcal{E} \subseteq [\mathcal{T}_\star]$ and $m \in \mathbb{N}_0$, we define inductively

$$[\omega_\star^m](\mathcal{E}) := \begin{cases} \mathcal{E} & \text{if } m = 0,\\ [\omega_\star](\mathcal{E}) := \{[T] \in [\mathcal{T}_\star] : \exists [T'] \in \mathcal{E}, T \cap T' \neq \emptyset\} & \text{if } m = 1,\\ [\omega_\star]([\omega_\star^{m-1}](\mathcal{E})) & \text{if } m > 1.\end{cases}$$

  We also need

$$\bigcup \mathcal{E} := \bigcup\{T \in \mathcal{T}_\star : [T] \in \mathcal{E}\} \subseteq \Gamma$$

  and

$$\omega_\star^m(\mathcal{E}) := \omega_\star^m\left(\bigcup \mathcal{E}\right).$$

3.2.3. Discretization space

We consider a mesh $\mathcal{T}_h$ of Γ as defined in Section 3.2.2 with corresponding nodes $\mathcal{N}_h$ and $\check{\mathcal{N}}_h$ and knots $\mathcal{K}_h$ and $\check{\mathcal{K}}_h$ such that for a given $p \in \mathbb{N}_0$ and each node $\check{z}_j \in \check{\mathcal{N}}_h$, it holds that $\#\check{z}_j \leq p+1$ for $j = 1, \dots, n-1$ and $\#\check{z}_0 = \#\check{z}_n = p+1$. Let the given partition of the boundary Γ be denoted with $\mathcal{T}_0$ with corresponding knots $\mathcal{K}_0$. We assume that the mesh $\mathcal{T}_h$ is a refinement of $\mathcal{T}_0$, i.e., that $\mathcal{K}_0 \subset \mathcal{K}_h$.

In addition, we assume that for $N = \#\mathcal{K}_h$ we have weights $\mathcal{W}_h = (\omega_i)_{i=1-p}^{N-p}$ given. Then, we define

$$\mathcal{S}(\mathcal{T}_h) := \hat{\mathscr{N}}^p(\check{\mathcal{K}}_h, \mathcal{W}_h) := \mathscr{N}^p(\check{\mathcal{K}}_h, \mathcal{W}_h) \circ \gamma|_{[a,b)}^{-1}.$$

A basis of $\mathcal{S}(\mathcal{T}_h)$ is given by

$$\{R_{i,p}|_{[a,b)} : i = 1-p, \dots, N - \#b + 1\} \circ \gamma|_{[a,b)}^{-1}.$$

The approximation space we use for the ansatz functions will be $\boldsymbol{\mathcal{S}}(\mathcal{T}_h) := \mathcal{S}(\mathcal{T}_h) \times \mathcal{S}(\mathcal{T}_h) \subset \boldsymbol{H}^{-1/2}(\Gamma)$ (cf. (3.6)). The canonical basis of $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ is

$$\{\boldsymbol{R}_i^k := R_{i,p}|_{[a,b)} \boldsymbol{e}_k : i = 1-p, \dots, N - \#b + 1, k = 1, 2\} \circ \gamma|_{[a,b)}^{-1}, \tag{3.7}$$

where $\boldsymbol{e}_k \in \mathbb{R}^2$ denotes the k-th identity vector.

A special case of the above described definition is to consider weights $\mathcal{W}_h = (1)_{i=1-p}^{N-p}$ and knots $\mathcal{K}_h$ resp. $\check{\mathcal{K}}_h$ with full multiplicity $\#z_j = p+1$ for all $j = 0, \dots, n$. The corresponding space of all B-splines $\mathscr{S}^p(\check{\mathcal{K}}_h)$ of degree p is then the space of all piecewise polynomials of degree p. Note that these polynomials generally are discontinuous at the nodes z_j for $j = 1, \dots, n$. We will denote this space of piecewise polynomials of order p by $\mathcal{P}^p(\mathcal{T}_h)$. The B-splines

$$\{B_{i,p}^{\check{\mathcal{K}}_h}|_{[a,b)} : i = 1-p, \dots, N-p\} \circ \gamma|_{[a,b)}^{-1},$$

form a basis of $\mathcal{P}^p(\mathcal{T}_h)$ as can be seen with Lemma 3.2. The two-dimensional space of piecewise polynomials $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h) := \mathcal{P}^p(\mathcal{T}_h) \times \mathcal{P}^p(\mathcal{T}_h)$ then has a basis given by

$$\{\boldsymbol{B}_i^k := B_{i,p}^{\check{\mathcal{K}}_h}|_{[a,b)} \cdot \boldsymbol{e}_k : i = 1-p, \dots, N-p \text{ and } k = 1, 2\} \circ \gamma|_{[a,b)}^{-1}.$$

3.3. Mesh-refinement and adaptive algorithm

In this subsection, we describe an adaptive algorithm for the mesh refinement, which also considers increasing the multiplicity of the knots.

To this end, we fix a polynomial order $p \in \mathbb{N}_0$, an adaptivity constant $0 < \theta \leq 1$ and a bound for the shape regularity constant $\kappa_{\max} > 0$. We denote the surface measure of Γ with μ_Γ (for Definition see [Gan14, Chapter 2]). With the index $\ell \in \mathbb{N}_0$, we count the number of steps. For $\ell := 0$ we start with an initial mesh $[\mathcal{T}_0]$. This includes nodes

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

$\check{\mathcal{N}}_0 = \{\check{z}_j^{[0]} : j = 1, \dots, n_0\}$ and knots $\check{\mathcal{K}}_0$ with $N_0 := |\check{\mathcal{K}}_0|$. Furthermore, we have initial weights $\mathcal{W}_0 = \{w_i^{[0]} : i = 1-p, \dots, N_0 - \#b + 1\}$. We then make the following assumptions

$$\begin{aligned} \kappa(\mathcal{T}_0) &\leq \kappa_{\max} \\ h_0 &\leq \mu_\Gamma(\Gamma)/4 \\ n_0 &\geq 4 \\ p + 1 &\leq N_0. \end{aligned}$$

For the mesh refinement, we will consider a node–based error estimator $\eta_\ell := \sum_{z \in \mathcal{N}_\ell} \eta_\ell(z)^2$ with local contributions $\eta_\ell(z)$ for $z \in \mathcal{N}_\ell$. In the next Section 3.4, we will present an error estimator of this form.

With a specific marking strategy which we will explain below, we then obtain a set of marked nodes and elements which will be refined by additionally considering knot insertion. The following algorithm is found in [FGHP16, Algorithm 2.2].

Algorithm 3.5. INPUT : *initial mesh* $[\mathcal{T}_0]$, *initial knots* $\mathcal{K}_0$, *initial weights* $\mathcal{W}_0$, *polynomial degree* $p \in \mathbb{N}_0$, *adaptivity parameter* $0 < \theta \leq 1$, *counter* $\ell := 0$.

1. *Compute discrete approximation* Φ_ℓ.
2. *Compute the refinement indicators* $\eta_\ell(z)$ *for all* $z \in \mathcal{N}_\ell$.
3. *Stop, if the error estimator* η_ℓ *is sufficiently small.*
4. *Determine a set of marked nodes* $\mathcal{M}_\ell \subseteq \mathcal{N}_\ell$ *of minimal cardinality such that*
$$\theta \eta_\ell \leq \sum_{z \in \mathcal{M}_\ell} \eta_\ell(z)^2.$$
5. *If both nodes of an element* $T \in \mathcal{T}_\ell$ *are marked,* T *will be marked.*
6. *For all other nodes* $z \in \mathcal{M}_\ell$ *the multiplicity will be increased, if it is smaller than* $p+1$, *otherwise, if the multiplicity is already* $p+1$, *the elements which contain the node* z *will be marked.*
7. *Refine all marked elements* $T \in \mathcal{T}_\ell$ *by bisection (insertion of a node with multiplicity one) of the corresponding* $\check{T} \in \check{\mathcal{T}}_\ell$. *Use further bisections to guarantee that the new partition* $\mathcal{T}_{\ell+1}$ *satisfies*
$$\check{\kappa}(\mathcal{T}_{\ell+1}) \leq 2\check{\kappa}(\mathcal{T}_0).$$
8. *For the obtained knots* $\check{\mathcal{K}}_{\ell+1}$, *we need new weights* $\mathcal{W}_{\ell+1}$, *which are uniquely chosen such that the denominator of the NURBS functions does not change, i.e.,*
$$\sum_{i=1-p}^{N_\ell - \#b + 1} w_i^{[\ell]} B_{i,p}^{\check{\mathcal{K}}_\ell} = \sum_{i=1-p}^{N_{\ell+1} - \#b + 1} w_i^{[\ell+1]} B_{i,p}^{\check{\mathcal{K}}_{\ell+1}}, \tag{3.8}$$
where N_ℓ *resp.* $N_{\ell+1}$ *denote* $\#\check{\mathcal{K}}_\ell$ *resp.* $\#\check{\mathcal{K}}_{\ell+1}$. *As the new weights are convex combinations of the initial weights* $\mathcal{W}_0$, *it holds that* $\min \mathcal{W}_0 \leq \min \mathcal{W}_{\ell+1} \leq \max \mathcal{W}_{\ell+1} \leq \max \mathcal{W}_0$ *for details see [FGP15, Section 4.2].*

TU Bibliothek, WIEN, Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

9. Set $\ell \leftarrow \ell + 1$ and go to 1.

OUTPUT : *We obtain approximate solutions Φ_ℓ and error estimators η_ℓ for all $\ell \in \mathbb{N}$.*

Note that for $\theta = 1$ the above algorithm leads to uniform refinement. Furthermore, with Corollary 3.4 and (3.8) we also obtain nestedness of

$$\hat{\mathcal{N}}^p(\check{\mathcal{K}}_\ell, \mathcal{W}_\ell) \subseteq \hat{\mathcal{N}}^p(\check{\mathcal{K}}_{\ell+1}, \mathcal{W}_{\ell+1}),$$

which also implies $\boldsymbol{\mathcal{S}}(\mathcal{T}_\ell) \subseteq \boldsymbol{\mathcal{S}}(\mathcal{T}_{\ell+1})$. For further details we refer to [FGHP16, Section 2.9].

3.4. The $(h$–$h/2)$–estimator

In order to refine a mesh adaptively, we need certain indicators to determine where the error is largest. To this end, we introduce the $(h$–$h/2)$–estimator and later also the local $(h$–$h/2)$–estimator for the Dirichlet problem. For a mesh $\mathcal{T}_h$ of Γ we consider the uniformly refined mesh $\mathcal{T}_{h/2}$. The corresponding nodes are given by $\mathcal{N}_h$ resp. $\mathcal{N}_{h/2}$. The corresponding approximation spaces are $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ and $\boldsymbol{\mathcal{S}}(\mathcal{T}_{h/2})$. Let Φ_h resp. $\Phi_{h/2}$ be the Galerkin solution of the Dirichlet problem (3.4) for $X_h = \boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ resp. $X_{h/2} = \boldsymbol{\mathcal{S}}(\mathcal{T}_{h/2})$. As $\boldsymbol{\mathcal{S}}(\mathcal{T}_h) \subseteq \boldsymbol{\mathcal{S}}(\mathcal{T}_{h/2})$ (cf. Corollary 3.4), we get with the Galerkin orthogonality (3.2) that

$$\left|\!\left|\!\left| \phi - \Phi_{h/2} \right|\!\right|\!\right|^2 + \left|\!\left|\!\left| \Phi_{h/2} - \Phi_h \right|\!\right|\!\right|^2 = \left|\!\left|\!\left| \phi - \Phi_h \right|\!\right|\!\right|^2.$$

Note that as an immediate consequence there also holds that

$$\left|\!\left|\!\left| \phi - \Phi_{h/2} \right|\!\right|\!\right| \leq \left|\!\left|\!\left| \phi - \Phi_h \right|\!\right|\!\right|.$$

Therefore, we define the $(h$–$h/2)$*–estimator* as follows

$$\mu(\mathcal{T}_h) := \left|\!\left|\!\left| \Phi_{h/2} - \Phi_h \right|\!\right|\!\right| \leq \left|\!\left|\!\left| \phi - \Phi_h \right|\!\right|\!\right|.$$

Due to the above estimate $\mu(\mathcal{T}_h)$ is an efficient error estimator. However, it does not contain information about local contributions, which we need for an adaptive algorithm. Therefore, we define the *local* $(h$–$h/2)$*–estimator* for node contributions as

$$\widetilde{\mu}(z) := \left\| h^{1/2}(\Phi_{h/2} - \Phi_h) \right\|_{\boldsymbol{L}^2(\omega(z))}, \quad \text{for all } z \in \mathcal{N}_h$$

and the corresponding global estimator

$$\widetilde{\mu}(\mathcal{T}_h) = \left\| h^{1/2}(\Phi_{h/2} - \Phi_h) \right\|_{\boldsymbol{L}^2(\Gamma)},$$

where $h \in L^\infty(\Gamma)$ denotes the local mesh-width function. Note that there holds

$$\widetilde{\mu}(\mathcal{T}_h)^2 \leq \sum_{z \in \mathcal{N}_h} \widetilde{\mu}(z)^2 \leq 2\widetilde{\mu}(\mathcal{T}_h)^2,$$

since the node patch contains two elements.

We will prove the equivalence between the $(h$–$h/2)$–estimator $\mu(\mathcal{T}_h)$ and the local $(h$–$h/2)$–estimator $\widetilde{\mu}(\mathcal{T}_h)$ for the space of elementwise piecewise polnomials $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ of order p. For that, we need appropriate results, i.e., approximation estimates and inverse-type estimates. The following lemma from [CP06, Theorem 4.1] gives an appoximation property of the L^2-orthogonal projection onto $\mathcal{P}^p(\mathcal{T}_h)$.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Lemma 3.6. *Let* $\Pi_h : L^2(\Gamma) \to \mathcal{P}^p(\mathcal{T}_h)$ *be the orthogonal projection onto* $\mathcal{P}^p(\mathcal{T}_h)$*. Then, there exists a constant* $C_{apx} > 0$ *such that*

$$\|\psi - \Pi_h \psi\|_{H^{-1/2}(\Gamma)} \leq C_{apx} \left\| h^{1/2}(\psi - \Pi_h\psi) \right\|_{L^2(\Gamma)} \quad \textit{for all } \psi \in L^2(\Gamma).$$

The constant C_{apx} *depends only on the boundary* Γ*.*

The following proposition is a special case of [FGHP17, Proposition 4.1] and gives us an inverse-type estimate for NURBS.

Proposition 3.7. *For a triangulation* $\mathcal{T}$ *as defined in Section 3.2.2, it holds that*

$$\left\| h^{1/2}\Psi \right\|_{L^2(\Gamma)} \leq C_{inv} \|\Psi\|_{H^{-1/2}(\Gamma)} \quad \textit{for all } \Psi \in \mathcal{S}(\mathcal{T}),$$

where $C_{inv} > 0$ *depends only on* $\check{\kappa}(\mathcal{T}), p, \gamma, \min(\mathcal{W}), \max(\mathcal{W})$ *for weights* $\mathcal{W}$ *corresponding to function* Ψ*.*

The above result is formulated for NURBS functions $\mathcal{S}(\mathcal{T})$, however we can apply it to the space of all piecewise polynomials $\mathcal{P}^p(\mathcal{T})$ when choosing the multiplicity of the knots as $p+1$ at each node. Although the above results are only formulated for scalar spaces, they clearly hold for the vector–valued spaces. We can now prove the equivalence of the two different $(h$–$h/2)$–estimators.

Lemma 3.8. *The* $(h$–$h/2)$*–estimator is equivalent to the local* $(h$–$h/2)$*–estimator for the ansatz space* $X_\ell = \boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$*, i.e., it holds that*

$$C_L^{-1}\widetilde{\mu}(\mathcal{T}_h) \leq \mu(\mathcal{T}_h) \leq C_H \widetilde{\mu}(\mathcal{T}_h).$$

The constant $C_H > 0$ *depends only on* Γ *and on the ellipticity and continuity of the energy scalar product, while* $C_L > 0$ *depends only on* $\check{\kappa}(\mathcal{T}_h), p, \gamma$*.*

Proof. We first prove $\mu(\mathcal{T}_h) \leq C_H \widetilde{\mu}(\mathcal{T}_h)$. Since the energy norm $|||\cdot|||$ is equivalent to $\|\cdot\|_{\boldsymbol{H}^{-1/2}}$ on $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_{h/2})$, i.e.,

$$|||\Phi_{h/2} - \Phi_h||| \simeq \left\|\Phi_{h/2} - \Phi_h\right\|_{\boldsymbol{H}^{-1/2}(\Gamma)},$$

we can switch to the $\|\cdot\|_{\boldsymbol{H}^{-1/2}}$ norm. As Φ_h is also the Galerkin projection of $\Phi_{h/2}$ onto $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$, we can apply the Céa-Lemma 3.1 to obtain that

$$\begin{aligned}\left\|\Phi_{h/2} - \Phi_h\right\|_{\boldsymbol{H}^{-1/2}(\Gamma)} &\leq C_{\text{Céa}} \min_{\Psi_h \in \boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)} \left\|\Phi_{h/2} - \Psi_h\right\|_{\boldsymbol{H}^{-1/2}(\Gamma)} \\ &\leq C_{\text{Céa}} \left\|\Phi_{h/2} - \Pi_h\Phi_{h/2}\right\|_{\boldsymbol{H}^{-1/2}(\Gamma)},\end{aligned}$$

with $\Pi_h : \boldsymbol{L}^2(\Gamma) \to \boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ being the $\boldsymbol{L}^2(\Gamma)$–orthogonal projection onto $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$. Then we can apply Lemma 3.6 and use the fact that Π_h is a projection and that it acts elementwise

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

on $T \in \mathcal{T}_h$ to obtain that

$$\begin{aligned}
\left\|\Phi_{h/2} - \Pi_h \Phi_{h/2}\right\|_{\boldsymbol{H}^{-1/2}(\Gamma)} &\leq C_{apx} \left\| h^{1/2}(\Phi_{h/2} - \Pi_h \Phi_{h/2}) \right\|_{\boldsymbol{L}^2(\Gamma)} \\
&= C_{apx} \left\| h^{1/2}(1 - \Pi_h)(\Phi_{h/2} - \Phi_h) \right\|_{\boldsymbol{L}^2(\Gamma)} \\
&\leq \frac{1}{\sqrt{2}} C_{apx} \left(\sum_{z \in \mathcal{N}_h} \left\| h^{1/2}(1 - \Pi_h)(\Phi_{h/2} - \Phi_h) \right\|^2_{\boldsymbol{L}^2(\omega(z))} \right)^{1/2} \\
&= \frac{1}{\sqrt{2}} C_{apx} \left(\sum_{z \in \mathcal{N}_h} \sum_{\{T \in \mathcal{T}_h : T \subseteq \omega(z)\}} h_T \left\| (1 - \Pi_h)(\Phi_{h/2} - \Phi_h) \right\|^2_{\boldsymbol{L}^2(T)} \right)^{1/2} \\
&\leq \frac{1}{\sqrt{2}} C_{apx} \left(\sum_{z \in \mathcal{N}_h} \sum_{\{T \in \mathcal{T}_h : T \subseteq \omega(z)\}} h_T \left\| \Phi_{h/2} - \Phi_h \right\|^2_{\boldsymbol{L}^2(T)} \right)^{1/2} \\
&\leq C_{apx} \left(\sum_{T \in \mathcal{T}_h} h_T \left\| \Phi_{h/2} - \Phi_h \right\|^2_{\boldsymbol{L}^2(T)} \right)^{1/2} \\
&= C_{apx} \left\| h^{1/2}(\Phi_{h/2} - \Phi_h) \right\|_{\boldsymbol{L}^2(\Gamma)} .
\end{aligned}$$

In the above inequalities we used the fact that the node patch consists of two elements.

Next, we consider $\widetilde{\mu}(\mathcal{T}_h) \leq C_L \mu(\mathcal{T}_h)$. This follows from Proposition 3.7 applied to $\Psi := \Phi_{h/2} - \Phi_h$. With the equivalence of $\|\cdot\|_{\boldsymbol{H}^{-1/2}(\Gamma)}$ and the energy norm $|||\cdot|||$, we obtain

$$\left\| h^{1/2}(\Phi_{h/2} - \Phi_h) \right\|_{\boldsymbol{L}^2(\Gamma)} \leq C_{inv} \left|\left|\left| \Phi_{h/2} - \Phi_h \right|\right|\right| .$$

This concludes the proof. ☐

Remark 3.9. *When trying to prove a similar result as Lemma 3.8 for an ansatz space* $X_\ell = \boldsymbol{\mathcal{S}}(\mathcal{T}_h)$*, we face several problems. The inverse equality from Proposition 3.7 holds for NURBS and therefore the estimate* $\widetilde{\mu}(\mathcal{T}_h) \leq C_L \mu(\mathcal{T}_h)$ *would still hold when using* $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ *instead of* $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$*.*
It is however not trivial to find an appropriate approximation property to prove the converse estimate. In the proof of Lemma 3.8 we used that the L^2*-orthogonal-projection onto* $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ *acts elementwise. If we were to replace* $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ *with* $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$*, the orthogonal projection does not necessarily act elementwise any more. The problem is that* $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ *might also require continuity at certain nodes.*
Another approach would be to use a Scott–Zhang type operator instead of the L^2*–orthogonal projection. A definition for an operator* $J_\star : L^2(\Gamma) \to \mathcal{S}(\mathcal{T}_h)$ *of this type can be found in [FGHP17, (5.9)]. In [FGHP17, Lemma 5.3] it is shown that* $J_\star$ *has a local projection property and that there holds local* L^2*–stability, i.e., for* $\psi \in L^2(\Gamma)$ *and* $T \in \mathcal{T}_h$ *there holds*

$$\|J_\star \psi\|_{L^2(T)} \leq C \|\psi\|_{L^2(\omega^p(T))} ,$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

where $C = C(p, \gamma, \max(\mathcal{W}_h), \kappa_{\max})$. We would however amongst other estimates still need an appropriate first–order approximation property for the Scott–Zhang type estimator with respect to the energy norm $|||\cdot|||$ or equivalently the the $\boldsymbol{H}^{-1/2}$ norm $\|\cdot\|_{\boldsymbol{H}^{-1/2}(\Gamma)}$. A proof for Lemma 3.8 for $X_\ell = \boldsymbol{S}(\mathcal{T}_h)$ therefore remains open.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4. Numerical computation of discrete integral operators

In this section, we deal with the computation of the occurring boundary operators and the corresponding matrices. This section strongly relies on [Gan14, Chapter 5], as the matrices for the Lamé equation are built up in a similar way compared to the Laplace equation.

We assume that $\Omega \subset \mathbb{R}^2$ and the boundary parametrization γ is defined as in Section 3.2.2. In addition, we assume that γ is two times piecewise differentiable. More precisely, let $a = \check{x}_0^\gamma < \ldots < \check{x}_{n_\gamma}^\gamma = b$ such that $\gamma|_{[\check{x}_{j-1}^\gamma, \check{x}_j^\gamma]}$ is two times continuously differentiable for $j = 1, \ldots, n_\gamma$. Furthermore, we write $x_j^\gamma := \gamma(\check{x}_j^\gamma)$. In addition, we assume that $\operatorname{diam}(\Omega)$ is sufficiently scaled such that V is an elliptic operator. The fact that γ is positively orientated means that the outer normal vector ν at any point $x = \gamma(t) = (\gamma_1(t), \gamma_2(t))^T \in \Gamma \backslash \{\gamma(\check{x}_1^\gamma, \ldots, \check{x}_{n_\gamma}^\gamma)\}$ is given by

$$\nu(x) = \frac{1}{|\gamma'(t)|} \begin{pmatrix} \gamma_2'(t) \\ -\gamma_1'(t) \end{pmatrix}. \tag{4.1}$$

The boundary Γ is parametrised by a NURBS curve γ, which means that the parametrisation has the special form

$$\gamma(t) = \sum_{i=1-p}^{N_\gamma - \#b + 1} C_i R_{i,p_\gamma}^{\check{\mathcal{K}}_\gamma, \mathcal{W}_\gamma}(t) \tag{4.2}$$

for all $t \in [a, b]$ and $\check{\mathcal{K}}_\gamma$ and $\mathcal{W}_\gamma$ being the knots and weights of the initial mesh defined on Γ. The polynomial degree is $p_\gamma \in \mathbb{N}_0$ and $(C_i)_{i=1-p}^{N_\gamma - \#b + 1}$ are *control points* in $\mathbb{R}^2$.
Moreover, we will use the discretization spaces described in Section 3.2.3.
In the following section we will also need the fundamental solution $\boldsymbol{U}^* = (U_{ij}^*(x,y))_{i,j=1}^2$ from Theorem 2.18, where

$$U_{ij}^*(x,y) = \frac{1}{4\pi\mu(2\mu+\lambda)} \left(-(3\mu+\lambda) \log|x-y| \, \delta_{ij} + (\mu+\lambda) \frac{(x_i - y_i)(x_j - y_j)}{|x-y|^2} \right)$$

and its ℓ-th partial derivative with respect to the second coordinate y is given by

$$\begin{aligned} \partial_{\ell,y} U_{ij}^*(x,y) = \frac{3\mu+\lambda}{4\pi\mu(2\mu+\lambda)} \frac{x_\ell - y_\ell}{|x-y|^2} \delta_{ij} + \\ \frac{\mu+\lambda}{4\pi\mu(2\mu+\lambda)} \frac{2(x_\ell - y_\ell)(x_i - y_i)(x_j - y_j) - (\delta_{\ell i}(x_j - y_j) + \delta_{\ell j}(x_i - y_i)) \, |x-y|^2}{|x-y|^4}. \end{aligned} \tag{4.3}$$

4.1. Dirichlet Problem

In this section, we aim to construct the setting for numerically solving problem (3.4): find $\Phi_h \in X_h$ such that

$$(\Phi_h, \Psi_h)_V = (\boldsymbol{f}, \Psi_h)_\Gamma \quad \text{for all } \Psi_h \in X_h. \tag{4.4}$$

For the approximation space X_h we chose $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ from Section 3.2.3 for given knots $\check{\mathcal{K}}_h$ and weights $\mathcal{W}_h$. We can then rewrite the above problem formulation with the basis function $\boldsymbol{R}_i^k$ from (3.7) for $i = 1-p, \dots, n - \#b + 1$ and $k = 1, 2$. For abbreviation, we set $\widehat{\boldsymbol{R}}_i^k := (R_{i,p}|_{[a,b)} \circ \gamma|_{[a,b)}^{-1}) \cdot \boldsymbol{e}_k$. Then, we define the symmetric positive definite matrix

$$V_h := \Big(\underbrace{\Big(\underbrace{(V\widehat{\boldsymbol{R}}_j^m, \widehat{\boldsymbol{R}}_i^k)_\Gamma}_{=:V_{h,ij}^{km}}\Big)_{i,j=1-p}^{N-\#b+1}}_{=:V_h^{km}}\Big)_{k,m=1}^2 \tag{4.5}$$

and the right-hand side vector

$$F_h := \Big(\underbrace{\Big(\underbrace{(\boldsymbol{f}, \widehat{\boldsymbol{R}}_i^k)_\Gamma}_{=:F_{h,i}^k}\Big)_{i=1-p}^{N-\#b+1}}_{=:F_h^k}\Big)_{k=1}^2. \tag{4.6}$$

Then, there exists a unique vector

$$\boldsymbol{c}_h := (\underbrace{c_{h,1-p}^1, \dots, c_{h,N-\#b+1}^1}_{=:c_h^1}, \underbrace{c_{h,1-p}^2, \dots, c_{h,n-\#b+1}^2}_{=:c_h^2}) \tag{4.7}$$

so that

$$V_h c_h = F_h \quad \text{and} \quad \Phi_h = \sum_{j=1-p}^{N-\#b+1} c_{h,j}^1 \widehat{\boldsymbol{R}}_j^1 + c_{h,j}^2 \widehat{\boldsymbol{R}}_j^2, \tag{4.8}$$

i.e., (4.8) is the algebraic system equivalent to the discrete variational formulation (4.4). As the matrix V_h and the vectors F_h and c_h are made up of sub-matrices and sub-vectors from the definitions (4.5), (4.6) and (4.7), we can write (4.8) as

$$\begin{pmatrix} V_h^{11} & V_h^{12} \\ V_h^{21} & V_h^{22} \end{pmatrix} \begin{pmatrix} c_h^1 \\ c_h^2 \end{pmatrix} = \begin{pmatrix} F_h^1 \\ F_h^2 \end{pmatrix}.$$

In order to calculate Φ_h we only have to solve the system of linear equations (4.8). In the following sections, we will prepare the entries of the matrix V_h and the vector F_h so that we can use tensor-Gauss quadrature to numerically approximate the integrals. For fixed positive weight functions $\theta_1, \theta_2 \in L^1([0,1])$ and integrands $f \in C([0,1]^2)$, we will have to approximate integrals of the form

$$Qf := \int_{[0,1]} \int_{[0,1]} f(s,t)\theta_1(s)\theta_2(t) \, dt \, ds.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

For $n_1, n_2 \in \mathbb{N}$ and nodes ξ_{1,q_1} resp. ξ_{2,q_2} and weights ω_{1,q_1} resp. ω_{2,q_2} for the Gauß quadrature on $[0,1]$ with weight function θ_1 resp. θ_2 (cf. [Pra10, Chapter 6.3]), we define the tensor Gauß quadrature as

$$Q_{n_1,n_2} f := \sum_{q_1=1}^{n_1} \sum_{q_2=1}^{n_2} f(\xi_{1,q_1}, \xi_{2,q_2}) \omega_{1,q_1} \omega_{2,q_2}.$$

We define the quadrature error as

$$E_{n_1,n_2} := Q - Q_{n_1,n_2}.$$

For $\ell = 1, 2$, we define linear functionals for $f \in C([0,1])$ by

$$\begin{aligned} Q^\ell_{n_\ell} f &= \sum_{q_\ell=1}^{n_\ell} f(\xi_{\ell,q_\ell}) \omega_{\ell,q_\ell}, \\ Q^\ell f &= \int_{[0,1]} f(r)\theta_\ell dr, \\ E^\ell_{n_\ell} &= Q^\ell - Q^\ell_{n_\ell}. \end{aligned}$$

In [Gan14, Theorem 5.1] the following error estimate is shown.

Theorem 4.1. *There holds the error estimate*

$$|E_{n_1,n_2}(f)| \leq \|\theta_2\|_{L^1([0,1])} \max_{s\in[0,1]} \left|E^1_{n_1} f(s,\cdot)\right| + \|\theta_1\|_{L^1([0,1])} \max_{t\in[0,1]} \left|E^1_{n_1} f(\cdot,t)\right|$$

for arbitrary $f \in C([0,1]^2)$, where the right hand side converges to zero for $n_1 \to \infty$ and $n_2 \to \infty$. ☐

Before we continue with the details for the computation of V_h and F_h, we first show, that the integral in the definition of the operator V defined in (2.33)

$$(V\boldsymbol{w})_i(x) = \int_\Gamma \sum_{j=1}^{n} \boldsymbol{U}^*_{ij}(x,y) w_j(y)\, dy, \quad \text{for } i = 1, 2,$$

does exist for $\boldsymbol{w} \in \boldsymbol{L}^2(\Omega)$. Consider a fixed $x = \gamma(s) \in \Gamma \backslash \{x^\gamma_1, \dots, x^\gamma_{n_\gamma}\}$. As the fundamental solution is a sum of terms of the type $\log|x-y|$ and $((x_i - y_i)(x_j - y_j))/|x-y|^2$ for $i, j = 1, 2$, we will consider those terms separately. First, we look at the log-term which also occurs for the Laplace equation. In [Gan14, Chapter 5, p. 52], it is shown for $w \in L^2(\Gamma)$ that

$$\begin{aligned} &\int_\Gamma |\log(|x-y|) w(y)|\ dy \\ &\qquad \leq \|w\|_{L^2(\Gamma)} \left(\int_{[a,b]} \left(\log\left(\frac{|\gamma(s) - \gamma(t)|}{|s-t|} \right) + \log(|s-t|) \right)^2 |\gamma'(t)|\, dt \right)^{1/2} \end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

and that the first integral does exist. For the rational part, it holds that

$$\begin{aligned}\int_\Gamma \left|\frac{(x_i-y_i)(x_j-y_j)}{|x-y|^2} w(y)\right| dy &\le \|w\|_{L^2(\Gamma)} \left(\int_\Gamma \left|\frac{(x_i-y_i)(x_j-y_j)}{|x-y|^2}\right|^2 dy\right)^{1/2} \\ &\le \|w\|_{L^2(\Gamma)} \int_\Gamma 1\ dy.\end{aligned}$$

Consequently, the integral in the definition of V exists. Not that for the operator K, we only have existence as a Cauchy principal value.

4.1.1. Numerical computation of V_h

In this section we will use several abbreviations

$$\begin{aligned}&R_i := R_{i,p}|_{[a,b)},\\ &\widehat{R}_i := R_{i,p}|_{[a,b)} \circ \gamma|_{[a,b)}^{-1},\\ &\check{U}^*_{km}(s,t) := U^*_{km}(\gamma(s),\gamma(t)),\\ &H_\ell := t_\ell - t_{\ell-1},\\ &\check{U}^*_{km,\ell_1,\ell_2}(s,t) := \check{U}^*_{km}(t_{\ell_1-1}H_{\ell_1}s, t_{\ell_2-1}+H_{\ell_2}t),\\ &\widetilde{R}_i(s) := R_i(s)\left|\gamma'(s)\right|,\\ &\widetilde{R}_{i,\ell}(s) := \widetilde{R}_i(t_{\ell-1}+H_\ell s).\end{aligned}$$

The aim of this section is to calculate the approximation of the left hand side of (3.4)

$$(\Phi_h,\Psi_h)_V = (V\Phi_h,\Psi_h)_\Gamma = \int_\Gamma\int_\Gamma \sum_{p=1}^2\sum_{q=1}^2 U^*_{pq}(x,y)\Phi_{h,q}(y)\Psi_{h,p}(x)\ dy\ dx,$$

where $\Phi_h,\Psi_h \in \boldsymbol{\mathcal{S}}(\mathcal{T}_h) \le \boldsymbol{H}^{-1/2}(\Gamma)$ with given knots $\check{\mathcal{K}}_h$ and weights $\mathcal{W}_h$ for $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$. To this end, we build up the matrix V_h. For $i,j = 1-p,\dots,N-\#b+1$ we then obtain

$$\begin{aligned}V^{km}_{h,ij} &= (V\widehat{\boldsymbol{R}}^m_j, \widehat{\boldsymbol{R}}^k_i)_\Gamma\\ &= \int_\Gamma\int_\Gamma \sum_{r=1}^2\sum_{q=1}^2 U^*_{rq}(x,y)\widehat{\boldsymbol{R}}^m_{j,q}(y)\widehat{\boldsymbol{R}}^k_{i,r}(x)\ dy\ dx.\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Since $\widehat{\boldsymbol{R}}_i^k = \widehat{R}_i \boldsymbol{e}_k$, for the r-th entry $(\widehat{R}_i^k)_r$ it holds that $(\widehat{R}_i^k)_r = \widehat{R}_i \delta_{kr}$. Then, by using the properties of the support of $R_i(\cdot)$, we get

$$\begin{aligned}
V_{h,ij}^{km} &= \int_\Gamma \int_\Gamma \sum_{r=1}^{2} \sum_{q=1}^{2} U_{rq}^*(x,y) \widehat{R}_j(y) \widehat{R}_i(x) \delta_{mq} \delta_{kr} \; dy \; dx \\
&= \int_\Gamma \int_\Gamma U_{km}^*(x,y) \widehat{R}_j(y) \widehat{R}_i(x) \; dy \; dx \\
&= \int_{[a,b)} \int_{[a,b)} U_{km}^*(\gamma(s),\gamma(t)) R_j(t) R_i(s) \left|\gamma'(s)\right| \left|\gamma'(t)\right| \; dt \; ds \\
&= \sum_{\ell_1=\max(i,1)}^{\min(i+p,N)} \sum_{\ell_2=\max(j,1)}^{\min(j+p,N)} \int_{[t_{\ell_1-1},t_{\ell_1}]} \int_{[t_{\ell_2-1},t_{\ell_2}]} \check{U}_{km}^*(s,t) \widetilde{R}_j(t) \widetilde{R}_i(s) \; dt \; ds \\
&= \sum_{\ell_1=\max(i,1)}^{\min(i+p,N)} \sum_{\ell_2=\max(j,1)}^{\min(j+p,N)} H_{\ell_1} H_{\ell_2} \int_{[0,1]} \int_{[0,1]} \check{U}_{km,\ell_1,\ell_2}^*(s,t) \widetilde{R}_{j,\ell_2}(t) \widetilde{R}_{i,\ell_1}(s) \; dt \; ds.
\end{aligned}$$

Now, for $H_{\ell_1}, H_{\ell_2} > 0$, we want to calculate the double integral

$$\int_{[0,1]} \int_{[0,1]} \check{U}_{km,\ell_1,\ell_2}^*(s,t) \widetilde{R}_{j,\ell_2}(t) \widetilde{R}_{i,\ell_1}(s) \; dt \; ds.$$

As the integrand is singular for $\gamma(s) = \gamma(t)$, we differentiate between three cases.

Case 1.$\gamma([t_{\ell_1-1}, t_{\ell_1}]) \cap \gamma([t_{\ell_2-1}, t_{\ell_2}]) = \emptyset$: In this case the integrand is continuous, we can apply Theorem 4.1 and use tensor-Gauss quadrature with weight function 1.

Case 2.$\gamma([t_{\ell_1-1}, t_{\ell_1}]) = \gamma([t_{\ell_2-1}, t_{\ell_2}])$: This implies that $\ell := \ell_1 = \ell_2$. Using the transformation $(s,t) \mapsto (s, s-t)$, it holds that

$$\begin{aligned}
&\int_{[0,1]} \int_{[0,1]} \check{U}_{km,\ell,\ell}^*(s,t) \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(t) \; dt \; ds \\
&= \underbrace{\int_{[0,1]} \int_{[0,s]} \check{U}_{km,\ell,\ell}^*(s,s-t) \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(s-t) \; dt \; ds}_{=:(\mathrm{I})} \\
&\quad + \underbrace{\int_{[0,1]} \int_{[s-1,0]} \check{U}_{km,\ell,\ell}^*(s,s-t) \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(s-t) \; dt \; ds}_{=:(\mathrm{II})}.
\end{aligned} \tag{4.9}$$

For the first summand (I) in (4.9), we use the *Duffy* transformation $(s,t) \mapsto (s, st)$ on $[0,1] \times [0,1]$ with Jacobi determinant s and then add and subtract $\log(st)$ in order for the argument of the *log* to be non-singular. The aim of the Duffy transformation is to bring

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the singularities to $s = 0$ or $t = 0$. We then obtain

$$\begin{aligned}
(\mathrm{I}) &= \int_{[0,1]} \int_{[0,1]} \breve{U}^*_{km,\ell,\ell}(s, s - st) \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(s - st) s \; dt \; ds \\
&= -\delta_{km} \frac{3\mu + \lambda}{4\pi\mu(2\mu + \lambda)} \\
&\quad \Bigg(\int_{[0,1]} \int_{[0,1]} \log \left(\frac{|\gamma(t_{\ell-1} + H_\ell s) - \gamma(t_{\ell-1} + H_\ell s(1-t))|}{st} \right) \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(s - st) s \; dt \; ds \\
&\quad + \int_{[0,1]} \int_{[0,1]} \log(s) \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(s - st) s \; dt \; ds \\
&\quad + \int_{[0,1]} \int_{[0,1]} \log(t) \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(s - st) s \; dt \; ds \Bigg) \\
&\quad + \frac{\mu + \lambda}{4\pi\mu(2\mu + \lambda)} \int_{[0,1]} \int_{[0,1]} \\
&\quad \frac{(\gamma_k(t_{\ell-1} + H_\ell s) - \gamma_k(t_{\ell-1} + H_\ell s(1-t)))(\gamma_m(t_{\ell-1} + H_\ell s) - \gamma_m(t_{\ell-1} + H_\ell s(1-t)))}{|\gamma(t_{\ell-1} + H_\ell s) - \gamma(t_{\ell-1} + H_\ell s(1-t))|^2} \\
&\quad \widetilde{R}_{i,\ell}(s) \widetilde{R}_{j,\ell}(s - st) s \; dt \; ds.
\end{aligned} \tag{4.10}$$

For the second summand (II) in (4.9), we transform $(s, t) \mapsto (1 - s, -t)$ before we apply

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the Duffy transformation:

$$
\begin{aligned}
\text{(II)} &= \int_{[0,1]} \int_{[0,s]} \check{U}^*_{km,\ell,\ell}(1-s, 1-s+t)\widetilde{R}_{i,\ell}(1-s)\widetilde{R}_{j,\ell}(1-s+t)\ dt\ ds \\
&= \int_{[0,1]} \int_{[0,1]} \check{U}^*_{km,\ell,\ell}(1-s, 1-s+st)\widetilde{R}_{i,\ell}(1-s)\widetilde{R}_{j,\ell}(1-s+st)s\ dt\ ds \\
&= -\delta_{km}\frac{3\mu+\lambda}{4\pi\mu(2\mu+\lambda)} \\
&\quad \Bigg(\int_{[0,1]} \int_{[0,1]} \log\left(\frac{|\gamma(t_{\ell-1}+H_\ell(1-s)) - \gamma(t_{\ell-1}+H_\ell(1-s+st))|}{st} \right) \\
&\quad \cdot \widetilde{R}_{i,\ell}(1-s)\widetilde{R}_{j,\ell}(1-s+st)s\ dt\ ds \\
&\quad + \int_{[0,1]} \int_{[0,1]} \log(s)\widetilde{R}_{i,\ell}(1-s)\widetilde{R}_{j,\ell}(1-s+st)s\ dt\ ds \\
&\quad + \int_{[0,1]} \int_{[0,1]} \log(t)\widetilde{R}_{i,\ell}(1-s)\widetilde{R}_{j,\ell}(1-s+st)s\ dt\ ds \Bigg) \\
&\quad + \frac{\mu+\lambda}{4\pi\mu(2\mu+\lambda)} \int_{[0,1]} \int_{[0,1]} \\
&\quad \frac{(\gamma_k(t_{\ell-1}+H_\ell(1-s)) - \gamma_k(t_{\ell-1}+H_\ell(1-s+st)))}{|\gamma(t_{\ell-1}+H_\ell(1-s)) - \gamma(t_{\ell-1}+H_\ell(1-s+st))|^2} \\
&\quad (\gamma_m(t_{\ell-1}+H_\ell(1-s))] - \gamma_m(t_{\ell-1}+H_\ell(1-s+st)))\widetilde{R}_{i,\ell}(1-s)\widetilde{R}_{j,\ell}(1-s+st)s\ dt\ ds.
\end{aligned}
\tag{4.11}
$$

For the remaining eight double integrals, we use Gauß quadrature with weight functions 1, $\log(s)$, $\log(t)$ resp. 1. The following corollary which strongly relies on [Gan14, Lemma 5.2] shows that we can apply Theorem 4.1.

Corollary 4.2. *If the parametrization γ is $q \geq 1$ times continuously differentiable on $[\check{x}^\gamma_{m-1}, \check{x}^\gamma_m]$ for $m \in \{1, \ldots, n_\gamma\}$, the integrands of the final terms in* (4.10) *and in* (4.11) *are, up to* $\log(s)$ *resp.* $\log(t)$, *$q-1$ times continuously partially differentiable on* $[0,1]^2$.

Proof. For the first three double integrals in (4.10) and (4.11) each, the proof is found in [Gan14, Lemma 5.2]. Therefore, we will only prove the statement for the last double integral in (4.10), and (4.11) can be treated analogously. First, we rewrite for $i = 1, 2$

$$
\begin{aligned}
&\gamma_i(t_{\ell-1}+H_\ell s) - \gamma_i(t_{\ell-1}+H_\ell s(1-t)) \\
&= \int_{[t_{\ell-1}+H_\ell s(1-t), t_{\ell-1}+H_\ell s]} \gamma_i'(\tau)\ d\tau \\
&= (t_{\ell-1}+H_\ell s - t_{\ell-1} - H_\ell s(1-t)) \int_{[0,1]} \gamma_i'(t_{\ell-1}+H_\ell s(1-t)+H_\ell st\tau)\ d\tau \\
&= (H_\ell st) \int_{[0,1]} \gamma_i'(t_{\ell-1}+H_\ell s(1-t)+H_\ell st\tau)\ d\tau.
\end{aligned}
$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

For $s,t \in (0,1]$, for the last integral in (4.10) we obtain that

$$\frac{(\gamma_k(t_{\ell-1}+H_\ell s)-\gamma_k(t_{\ell-1}+H_\ell s(1-t)))(\gamma_m(t_{\ell-1}+H_\ell s)-\gamma_m(t_{\ell-1}+H_\ell s(1-t)))}{|\gamma(t_{\ell-1}+H_\ell s)-\gamma(t_{\ell-1}+H_\ell s(1-t))|^2}$$
$$=\frac{\left(\int_{[0,1]}\gamma_k'(t_{\ell-1}+H_\ell s(1-t)+H_\ell st\tau)d\tau\right)\left(\int_{[0,1]}\gamma_m'(t_{\ell-1}+H_\ell s(1-t)+H_\ell st\tau)d\tau\right)}{\left|\int_{[0,1]}\gamma'(t_{\ell-1}+H_\ell s(1-t)+H_\ell st\tau)\;d\tau\right|^2}.$$

The above term can be continuously extended for $s=0$ or $t=0$ with

$$\frac{\left(\gamma_k'|_{[t_\ell-1,t_\ell]}(t_{\ell-1}+H_\ell s(1-t))\right)\left(\gamma_m'|_{[t_\ell-1,t_\ell]}(t_{\ell-1}+H_\ell s(1-t))\right)}{\left|\gamma|_{[t_\ell-1,t_\ell]}'(t_{\ell-1}+H_\ell s(1-t))\right|^2}.$$

Since γ is injective and γ' vanishes nowhere, its modulus is positive for all $s,t\in[0,1]$. Due to the smoothness of γ and the basis functions $\widetilde{R}_{i,\ell},\widetilde{R}_{j,\ell}$ the integrand of the last double integral in (4.10) is $q-1$ times continuously differentiable. ☐

Case 3.$|\gamma(t_{\ell_1-1},t_{\ell_1})\cap\gamma(t_{\ell_2-1},t_{\ell_2})|=1$: In this case we have adjacent elements. Without loss of generality, we assume that the singularity in the integrand appears at $s=0$ and $t=1$. The other case can be treated analogously. We either have $t_{\ell_1-1}=t_{\ell_2}$ or $t_{\ell_2}=b\wedge t_{\ell_1-1}=a$. Using the transformation $t\mapsto 1-t$, it holds that

$$\begin{aligned}&\int_{[0,1]}\int_{[0,1]}\check{U}^*_{km,\ell_1,\ell_2}(s,t)\widetilde{R}_{i,\ell_1}(s)\widetilde{R}_{j,\ell_2}(t)\;dt\;ds\\&=\underbrace{\int_{[0,1]}\int_{[0,s]}\check{U}^*_{km,\ell_1,\ell_2}(s,1-t)\widetilde{R}_{i,\ell_1}(s)\widetilde{R}_{j,\ell_2}(1-t)\;dt\;ds}_{=:\text{(I)}}\\&\quad+\underbrace{\int_{[0,1]}\int_{[s,1]}\check{U}^*_{km,\ell_1,\ell_2}(s,1-t)\widetilde{R}_{i,\ell_1}(s)\widetilde{R}_{j,\ell_2}(1-t)\;dt\;ds}_{=:\text{(II)}}.\end{aligned}\tag{4.12}$$

Again, we proceed similarly as in the previous case and split the two summands and first apply the Duffy transformation. For the first summand (I) in (4.12) we then add and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

subtract $\log(s)$:

$$
\begin{aligned}
&\int_{[0,1]}\int_{[0,1]} \check{U}^*_{km,\ell_1,\ell_2}(s,1-st)\widetilde{R}_{i,\ell_1}(s)\widetilde{R}_{j,\ell_2}(1-st)s\;dt\;ds\\
&= -\delta_{km}\frac{3\mu+\lambda}{4\pi\mu(2\mu+\lambda)}\\
&\quad\left(\int_{[0,1]}\int_{[0,1]} \log\left(\frac{|\gamma(t_{\ell_1-1}+H_{\ell_1}s)-\gamma(t_{\ell_2-1}+H_{\ell_2}(1-st))|}{s}\right)\widetilde{R}_{i,\ell_1}(s)\widetilde{R}_{j,\ell_2}(1-st)s\;dt\;ds\right.\\
&\quad\left.+\int_{[0,1]}\int_{[0,1]} \log(s)\widetilde{R}_{i,\ell_1}(s)\widetilde{R}_{j,\ell_2}(1-st)s\;dt\;ds\right)\\
&\quad+\frac{\mu+\lambda}{4\pi\mu(2\mu+\lambda)}\int_{[0,1]}\int_{[0,1]}\\
&\quad\frac{(\gamma_k(t_{\ell_1-1}+H_{\ell_1}s)-\gamma_k(t_{\ell_2-1}+H_{\ell_2}(1-st)))(\gamma_m(t_{\ell_1-1}+H_{\ell_1}s)-\gamma_m(t_{\ell_2-1}+H_{\ell_2}(1-st)))}{|\gamma(t_{\ell_1-1}+H_{\ell_1}s)-\gamma(t_{\ell_2-1}+H_{\ell_2}(1-st))|^2}\\
&\quad\widetilde{R}_{i,\ell_1}(s)\widetilde{R}_{j,\ell_2}(1-st)s\;dt\;ds.
\end{aligned}
\tag{4.13}
$$

For the second summand (II) in (4.12) we first apply Fubini's theorem to get the inner integral over the domain $[0,t]$, apply the Duffy transformation, and then apply Fubini's theorem again. Then, we add and subtract $\log(t)$:

$$
\begin{aligned}
&\int_{[0,1]}\int_{[0,1]} \check{U}^*_{km,\ell_1,\ell_2}(st,1-t)\widetilde{R}_{i,\ell_1}(st)\widetilde{R}_{j,\ell_2}(1-t)t\;dt\;ds\\
&= -\delta_{km}\frac{3\mu+\lambda}{4\pi\mu(2\mu+\lambda)}\\
&\quad\left(\int_{[0,1]}\int_{[0,1]} \log\left(\frac{|\gamma(t_{\ell_1-1}+H_{\ell_1}st)-\gamma(t_{\ell_2-1}+H_{\ell_2}(1-t))|}{t}\right)\widetilde{R}_{i,\ell_1}(st)\widetilde{R}_{j,\ell_2}(1-t)t\;dt\;ds\right.\\
&\quad\left.+\int_{[0,1]}\int_{[0,1]} \log(t)\widetilde{R}_{i,\ell_1}(st)\widetilde{R}_{j,\ell_2}(1-t)t\;dt\;ds\right)\\
&\quad+\frac{\mu+\lambda}{4\pi\mu(2\mu+\lambda)}\int_{[0,1]}\int_{[0,1]}\\
&\quad\frac{(\gamma_k(t_{\ell_1-1}+H_{\ell_1}st)-\gamma_k(t_{\ell_2-1}+H_{\ell_2}(1-t)))(\gamma_m(t_{\ell_1-1}+H_{\ell_1}st)-\gamma_m(t_{\ell_2-1}+H_{\ell_2}(1-t)))}{|\gamma(t_{\ell_1-1}+H_{\ell_1}st)-\gamma(t_{\ell_2-1}+H_{\ell_2}(1-t))|^2}\\
&\quad\widetilde{R}_{i,\ell_1}(st)\widetilde{R}_{j,\ell_2}(1-t)t\;dt\;ds.
\end{aligned}
\tag{4.14}
$$

For the remaining six double integrals, we use Gauß quadrature with weight functions 1, $\log(s)$ resp. $\log(t)$. Due to the following corollary we can again apply Theorem 4.1.

Corollary 4.3. *If the parametrization γ is $q \geq 1$ times continuously differentiable on $[\check{x}^\gamma_{j-1}, \check{x}^\gamma_j]$ for $j \in \{1,\dots,n_\gamma\}$, the integrands of the final terms in* (4.13) *and in* (4.14) *are, up to $\log(s)$ resp. $\log(t)$, $q-1$ times continuously partially differentiable on $[0,1]^2$.*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Proof. For the first two double integrals in (4.13) and (4.14), the proof is found in [Gan14, Lemma 5.3]. We will only prove the statement for the last double integral in (4.13) and (4.14) can be treated analogously. We first consider the case $t_{\ell_1-1} = t_{\ell_2}$. For $i = 1, 2$, we rewrite

$$\begin{aligned}\gamma_i(t_{\ell_1-1} + H_{\ell_1-1}s) - \gamma_i(t_{\ell_2-1} + H_{\ell_2}(1-st)) &= \int_{[t_{\ell_2}-H_{\ell_2}st, t_{\ell_2}+H_{\ell_1}s]} \gamma_i'(\tau)]\; d\tau \\ &= s\int_{[-H_{\ell_2}t, H_{\ell_1}]} \gamma_i'(t_{\ell_2}+s\tau)\; d\tau.\end{aligned}$$

Then, for $s \in (0,1]$, $t \in [0,1]$, we consider

$$\begin{aligned}&\frac{(\gamma_k(t_{\ell_1-1}+H_{\ell_1}s) - \gamma_k(t_{\ell_2-1}+H_{\ell_2}(1-st)))(\gamma_m(t_{\ell_1-1}+H_{\ell_1}s)] - \gamma_m(t_{\ell_2-1}+H_{\ell_2}(1-st)))}{|\gamma(t_{\ell_1-1}+H_{\ell_1}s) - \gamma(t_{\ell_2-1}+H_{\ell_2}(1-st))|^2} \\ &= \frac{\left(\int_{[-H_{\ell_2}t,H_{\ell_1}]}\gamma_k'(t_{\ell_2}+s\tau)\; d\tau\right)\left(\int_{[-H_{\ell_2}t,H_{\ell_1}]}\gamma_m'(t_{\ell_2}+s\tau)\; d\tau\right)}{\left|\int_{[-H_{\ell_2}t,H_{\ell_1}]}\gamma'(t_{\ell_2}+s\tau)\; d\tau\right|^2}. \qquad (4.15)\end{aligned}$$

Due to the smoothness of the integrand, it is $q-1$ times continuously partially differentiable. For the case $s \to 0$, we rewrite

$$\begin{aligned}\lim_{s\to 0}\int_{[-H_{\ell_2}t,H_{\ell_1}]}\gamma_i'(t_{\ell_2}+s\tau)\; d\tau &= \lim_{s\to 0}\int_{[-H_{\ell_2}t,0]}\gamma_i'(t_{\ell_2}+s\tau)\; d\tau + \lim_{s\to 0}\int_{[0,H_{\ell_1}]}\gamma_i'(t_{\ell_2}+s\tau)\; d\tau \\ &= \int_{[-H_{\ell_2}t,0]}\gamma_i'^{\ell}(t_{\ell_2})\; d\tau + \int_{[0,H_{\ell_1}]}\gamma_i'^{r}(t_{\ell_2})\; d\tau \\ &= H_{\ell_2}t\gamma_i'^{\ell}(t_{\ell_2}) + H_{\ell_1}\gamma_i'^{r}(t_{\ell_2}),\end{aligned}$$

where γ'^{ℓ} resp. γ'^{r} denote the left resp. right derivative of γ. Therefore, and due to the injectivity of γ and the fact that $\gamma'^{\ell}(t_{\ell_2})$ is no negative multiple of $\gamma'^{r}(t_{\ell_2})$, we can continuously extend the last term in (4.15) with

$$\frac{\left(H_{\ell_2}t\gamma_k'^{\ell}(t_{\ell_2}) + H_{\ell_1}\gamma_k'^{r}(t_{\ell_2})\right)\left(H_{\ell_2}t\gamma_m'^{\ell}(t_{\ell_2}) + H_{\ell_1}\gamma_m'^{r}(t_{\ell_2})\right)}{|H_{\ell_2}t\gamma'^{\ell}(t_{\ell_2}) + H_{\ell_1}\gamma'^{r}(t_{\ell_2})|^2}$$

at $s = 0$. By multiplying with the terms $\widetilde{g}_{p,\ell_2}(1-st)\widetilde{R}_{i,\ell_1}(s)$, we obtain the integrand of the last integral in (4.13). In the case that $t_{\ell_2} = b$ and $t_{\ell_1-1} = a$, one can shift t_{ℓ_1} to $t_{\ell_1-1} + (b-a) = b$ and the proof works as before. ☐

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.1.2. Numerical computation of F_h

Before we start, we add some more definitions in addition to the abbreviations that were introduced at the beginning of Subsection 4.1.1:

$$\begin{aligned}
&\mathfrak{d}_{\boldsymbol{n},y}\check{\boldsymbol{U}}^*(s,t) := \mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(\gamma(s),\gamma(t)),\\
&\mathfrak{d}_{\boldsymbol{n},y}\check{\boldsymbol{U}}^*_{\ell_1,\ell_2}(s,t) := \mathfrak{d}_{\boldsymbol{n},y}\check{\boldsymbol{U}}^*(t_{\ell_1-1}+H_{\ell_1}s, t_{\ell_2-1}+H_{\ell_2}t),\\
&\check{\boldsymbol{g}}(s) := \boldsymbol{g}(\gamma(s)),\\
&\widetilde{\boldsymbol{g}}(s) := \check{\boldsymbol{g}}(s)\left|\gamma'(s)\right|,\\
&\check{\boldsymbol{g}}_\ell(s) := \check{\boldsymbol{g}}(t_{\ell-1}+H_\ell s),\\
&\widetilde{\boldsymbol{g}}_\ell(s) := \widetilde{\boldsymbol{g}}(t_{\ell-1}+H_\ell s),\\
&d(s,t)_{\ell_1,\ell_2} := \left|t_{\ell_2-1}-t_{\ell_1-1}+H_{\ell_2}t-H_{\ell_1}s\right|.
\end{aligned}$$

In this subsection, we aim to calculate the approximation of the right–hand side of (3.4)

$$(\boldsymbol{f},\psi_h)_\Gamma = ((K+\frac{1}{2}I)\boldsymbol{g},\psi_h)_\Gamma,$$

where $\psi_h \in \boldsymbol{\mathcal{S}}(\mathcal{T}_h) \leq \boldsymbol{H}^{-1/2}(\Gamma)$ with given knots $\check{\mathcal{K}}_h$ and weights $\mathcal{W}_h$ for $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$. This is realized by building up the vector F_h as defined in (4.6). The vector F_h is the sum of two other vectors $G_h := (G_h^1, G_h^2)$ and $Kg_h := (Kg_h^1, Kg_h^2)$, which are defined through the equation below:

$$\begin{aligned}
F_h &= \frac{1}{2}G_h + Kg_h\\
&= \frac{1}{2}(\underbrace{((\underbrace{\boldsymbol{g},\widehat{\boldsymbol{R}}_i^k)_\Gamma}_{=:G^k_{h,i}})_{i=1-p}^{N-\#b+1}}_{=:G_h^k})_{k=1}^2 + (\underbrace{((\underbrace{K\boldsymbol{g},\widehat{\boldsymbol{R}}_i^k)_\Gamma}_{=:Kg^k_{h,i}})_{i=1-p}^{N-\#b+1}}_{=:Kg_h^k})_{k=1}^2.
\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

We first consider G_h. For $k = 1, 2$ and $i = 1-p, \dots, N - \#b + 1$, it holds that

$$
\begin{aligned}
G^k_{h,i} &= (\boldsymbol{g}, \widehat{\boldsymbol{R}}^k_i)_\Gamma \\
&= \int_\Gamma \sum_{j=1}^{2} g_j(x)(\widehat{R}^k_i(x))_j \; dx \\
&= \int_\Gamma \sum_{j=1}^{2} g_j(x)\widehat{R}_i(x)\delta_{kj} \; dx \\
&= \int_\Gamma g_k(x)\widehat{R}_i(x) \; dx \\
&= \int_{[a,b)} \breve{g}_k(s)R_i(s) \left|\gamma'(s)\right| \; ds \\
&= \sum_{\ell=\max(i,1)}^{\min(i+p,N)} \int_{[t_{\ell-1},t_\ell]} \breve{g}_k(s)\widetilde{R}_i(s) \; ds \\
&= \sum_{\ell=\max(i,1)}^{\min(i+p,N)} H_\ell \int_{[0,1]} \breve{g}_{k,\ell}(s)\widetilde{R}_{i,\ell}(s) \; ds,
\end{aligned}
$$

where we used the properties of the support of $R_i(\cdot)$. Next, we consider Kg_h. For $k = 1, 2$ and $i = 1-p, \dots, N - \#b + 1$ and by using the properties of the support of $R_i(\cdot)$ and with (2.31) and where $(\gamma_{y,1}\boldsymbol{U}^*(x,y))_{kr}$ denotes the kr-th entry of $\gamma_{y,1}\boldsymbol{U}^*(x,y)$, it holds that

$$
\begin{aligned}
&Kg^k_{h,i} = \\
&= \lim_{\varepsilon\to 0} \int_\Gamma \int_{\substack{\Gamma \\ y\in\Gamma:|y-x|\geq\varepsilon}} \sum_{\ell=1}^{2}\sum_{r=1}^{2} (\mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(x,y))_{\ell r}\, g_r(y)(\widehat{R}^k_i(x))_\ell \; dy \; dx \\
&= \lim_{\varepsilon\to 0} \int_\Gamma \int_{\substack{\Gamma \\ y\in\Gamma:|y-x|\geq\varepsilon}} \sum_{r=1}^{2} (\mathfrak{d}_{\boldsymbol{n},y}\boldsymbol{U}^*(x,y))_{kr} g_r(y)\hat{R}_i(x) \; dy \; dx \\
&= \lim_{\varepsilon\to 0} \int_{[a,b)} \int_{\substack{[a,b) \\ t\in[a,b):|t-s|\geq\varepsilon}} \sum_{r=1}^{2} (\mathfrak{d}_{\boldsymbol{n},y}\breve{\boldsymbol{U}}^*(s,t))_{kr}\breve{g}_r(t)R_i(s)\left|\gamma'(s)\right|\left|\gamma'(t)\right| \; dt \; ds \\
&= \sum_{\ell_1=\max(i,1)}^{\min(i+p,N)} \sum_{\ell_2=1}^{N} \lim_{\varepsilon\to 0} \int_{[t_{\ell_1-1},t_{\ell_1}]} \int_{\{t\in[t_{\ell_2-1},t_{\ell_2}]:|t-s|\geq\varepsilon\}} \sum_{r=1}^{2} (\mathfrak{d}_{\boldsymbol{n},y}\breve{\boldsymbol{U}}^*(s,t))_{kr}\widetilde{g}_r(t)\widetilde{R}_i(s) \; dt \; ds \\
&= \sum_{\ell_1=\max(i,1)}^{\min(i+p,N)} \sum_{\ell_2=1}^{N} H_{\ell_1}H_{\ell_2} \\
&\qquad \lim_{\varepsilon\to 0} \int_{[0,1]} \int_{\{t\in[0,1]:d(s,t)_{\ell_1,\ell_2}\geq\varepsilon\}} \sum_{r=1}^{2} (\mathfrak{d}_{\boldsymbol{n},y}\breve{\boldsymbol{U}}^*_{\ell_1,\ell_2}(s,t))_{kr}\widetilde{g}_{r,\ell_2}(t)\widetilde{R}_{i,\ell_1}(s) \; dt \; ds.
\end{aligned}
$$

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

For $H_{\ell_1}, H_{\ell_2} > 0$, we take a closer look at the double integral

$$\lim_{\varepsilon\to 0}\int_{[0,1]}\int_{\{t\in[0,1]:d(s,t)_{\ell_1,\ell_2}\geq\varepsilon\}}\sum_{r=1}^{2}\left(\mathfrak{D}_{\boldsymbol{n},y}\check{\boldsymbol{U}}^*_{\ell_1,\ell_2}(s,t)\right)_{kr}\widetilde{g}_{r,\ell_2}(t)\widetilde{R}_{i,\ell_1}(s)\ dt\ ds. \tag{4.16}$$

Again, we differentiate between three cases.

Case 1.$\gamma([t_{\ell_1-1},t_{\ell_1}])\cap\gamma([t_{\ell_2-1},t_{\ell_2}])=\emptyset$: In this case the integrand has no singularities and (4.16) simplifies to

$$\int_{[0,1]}\int_{[0,1]}\sum_{r=1}^{2}\left(\mathfrak{D}_{\boldsymbol{n},y}\check{\boldsymbol{U}}^*_{\ell_1,\ell_2}(s,t)\right)_{kr}\widetilde{g}_{r,\ell_2}(t)\widetilde{R}_{i,\ell_1}(s)\ dt\ ds.$$

No further steps are required as the integrand is continuous. We use Gauss quadrature with weight function 1 and apply Theorem 4.1.

Case 2.$\gamma([t_{\ell_1-1},t_{\ell_1}])=\gamma([t_{\ell_2-1},t_{\ell_2}])$: This implies that $\ell:=\ell_1=\ell_2$ and we have identical elements. In this case, (4.16) is

$$\lim_{\varepsilon\to 0}\int_{[0,1]}\int_{\substack{[0,1]\\ t\in[0,1]:|t-s|\geq\varepsilon}}\underbrace{\sum_{r=1}^{2}\left(\mathfrak{D}_{\boldsymbol{n},y}\check{\boldsymbol{U}}^*_{\ell_1,\ell_2}(s,t)\right)_{kr}\widetilde{g}_{r,\ell_2}(t)\widetilde{R}_{i,\ell_1}(s)}_{=:\kappa(s,t)}\ dt\ ds, \tag{4.17}$$

and we have the singularity along the diagonal of the square $[0,1]^2$. In order to simplify notation, we let $\kappa(s,t)$ be the integrand of the above integral. Then, we set $z:=t-s$ and in the next step, we separate the integration domain and change the order of integration

$$\begin{aligned}&\lim_{\varepsilon\to 0}\int_{[0,1]}\int_{\{z\in[-s,1-s]:|z|\geq\varepsilon\}}\kappa(s,s+z)\ dz\ ds=\\ &=\lim_{\varepsilon\to 0}\left(\int_{[\varepsilon,1]}\int_{[0,1-z]}\kappa(s,s+z)\ ds\ dz+\int_{[-1,-\varepsilon]}\int_{[-z,1]}\kappa(s,s+z)\ ds\ dz\right).\end{aligned}$$

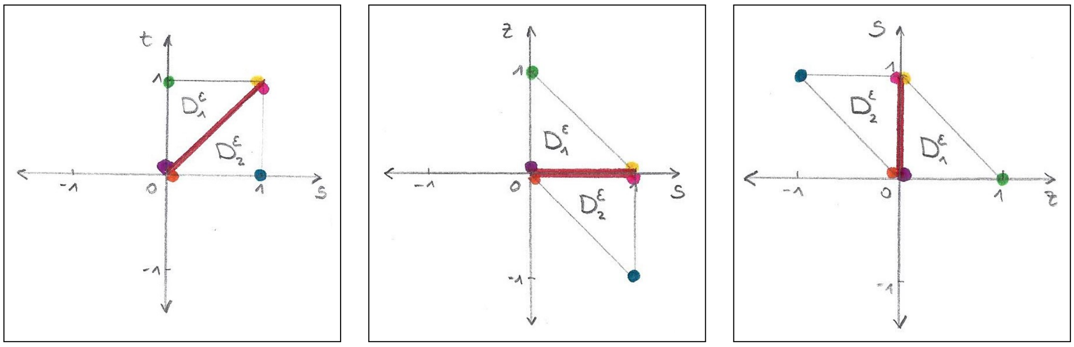
We can see in Figure 4.1 how the integration domain is transformed. A red line indicates the singularity. The original domain $[0,1]^2$ is divided into two triangles along the singularity, namely D_1^ε and D_2^ε. After the last step, the domains are as follows (cf. Figure 4.1c)

$$D_1^\varepsilon=\left\{\begin{matrix}\varepsilon\leq z\leq 1\\ 0\leq s\leq 1-z\end{matrix}\right\}\quad\text{and}\quad D_2^\varepsilon=\left\{\begin{matrix}-1\leq z\leq-\varepsilon\\ -z\leq s\leq 1\end{matrix}\right\}.$$

In the next step, we substitute $\widetilde{z}:=-z$ in the second integral and $\widetilde{s}:=s+z$ in the first integral and then redefine $s:=\widetilde{s}$ and $z:=\widetilde{z}$. We obtain that

$$\begin{aligned}&\lim_{\varepsilon\to 0}\left(\int_{[\varepsilon,1]}\int_{[z,1]}\kappa(\widetilde{s}-z,\widetilde{s})\ d\widetilde{s}\ dz+\int_{[\varepsilon,1]}\int_{[\widetilde{z},1]}\kappa(s,s-\widetilde{z})\ ds\ d\widetilde{z}\right)\\ &=\lim_{\varepsilon\to 0}\left(\int_{[\varepsilon,1]}\int_{[z,1]}\kappa(s-z,s)\ ds\ dz+\int_{[\varepsilon,1]}\int_{[z,1]}\kappa(s,s-z)\ ds\ dz\right) \qquad (4.18)\\ &=\lim_{\varepsilon\to 0}\left(\int_{[\varepsilon,1]}\int_{[z,1]}\kappa(s-z,s)+\kappa(s,s-z)\ ds\ dz\right).\end{aligned}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) original domain $[0, 1]^2$ (b) $z := t - s$ (c) change order of integration

Figure 4.1.: K, identical elements: Transformations step 1

The transformations of the integration domain are visualized in Figure 4.2. It is essential that the singularity for both domains D_1^ε and D_2^ε occurs along the same line segment $\{0\} \times [0, 1]$. Then, we see that the domains D_1^ε and D_2^ε are identical and we can merge the last two double integrals in (4.18).

Next, we substitute $w_1 := s$ and $w_2 := s - z$ to obtain that

$$\lim_{\varepsilon \to 0} \int_{[\varepsilon,1]} \int_{[0,w_1-\varepsilon]} \kappa(w_2, w_1) + \kappa(w_1, w_2)\; dw_2\; dw_1.$$

As a last step, we make a Duffy transformation with $(w_1, w_2) \mapsto (s, st)$ to obtain that

$$\lim_{\varepsilon \to 0} \int_{[\varepsilon,1]} \int_{[0,1-\varepsilon]} (\kappa(st, s) + \kappa(s, st))\, s\; dt\; ds. \tag{4.19}$$

In Figure 4.3, it is shown, how the last two transformations act on the integration domain. Note that the singularity is transformed from the diagonal to two sides of the square $[0, 1]^2$.

The following corollary shows that the limit in the statement (4.19) does exist.

Corollary 4.4. *If the parametrization* γ *is* $q \geq 2$ *times continuously differentiable on* $[\check{x}^\gamma_{m-1}, \check{x}^\gamma_m]$ *for* $m \in \{1, \ldots, n_\gamma\}$ *and if* $\boldsymbol{g} \circ \gamma$ *is* $q - 2$ *times continuously differentiable on* $[\check{x}^\gamma_{m-1}, \check{x}^\gamma_m]$ *for* $m \in \{1, \ldots, n_\gamma\}$*, then there exists a function* f *which is* $q - 2$ *times continuously differentiable so that*

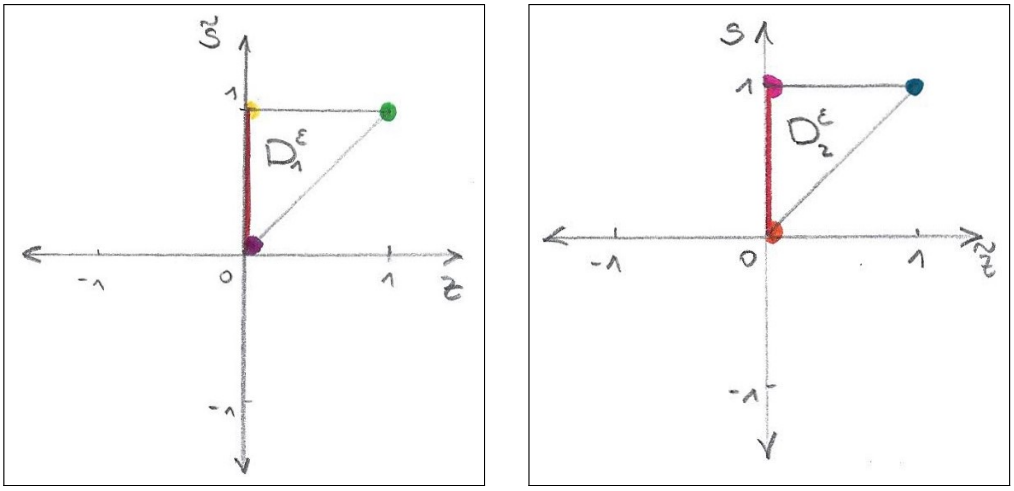
$$(\kappa(st, s) + \kappa(s, st))\, s = f(s, t), \quad \text{for } (s, t) \in (0, 1)^2,$$

and so that $\kappa(st, s) + \kappa(s, st)$ *can be continuously extended with* $f(s, t)$ *onto* $[0, 1]^2$.

Proof. As we can see with (2.31) and (4.3) the conormal derivative is a sum of terms of the following three different types

$$\underbrace{\frac{x_i - y_i}{|\boldsymbol{x} - \boldsymbol{y}|^2} n_j}_{=:(\mathrm{I})}, \quad \underbrace{\frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)}{|\boldsymbol{x} - \boldsymbol{y}|^4} n_r}_{=:(\mathrm{II})}, \quad \underbrace{\frac{\delta_{ik}(x_i - y_i) + \delta_{ij}(x_k - y_k)}{|\boldsymbol{x} - \boldsymbol{y}|^2} n_r}_{=:(\mathrm{III})},$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Wien Bibliothek Your knowledge hub



(a) $\widetilde{s} := s + z$ (b) $\widetilde{z} := -z$

Figure 4.2.: K, identical elements: Transformations step 2

for $\boldsymbol{x} = \gamma(t_\ell + H_\ell s)$ and $\boldsymbol{y} = \gamma(t_\ell + H_\ell t)$ and for $i, j, k, r = 1, 2$. First we consider (I) and rewrite

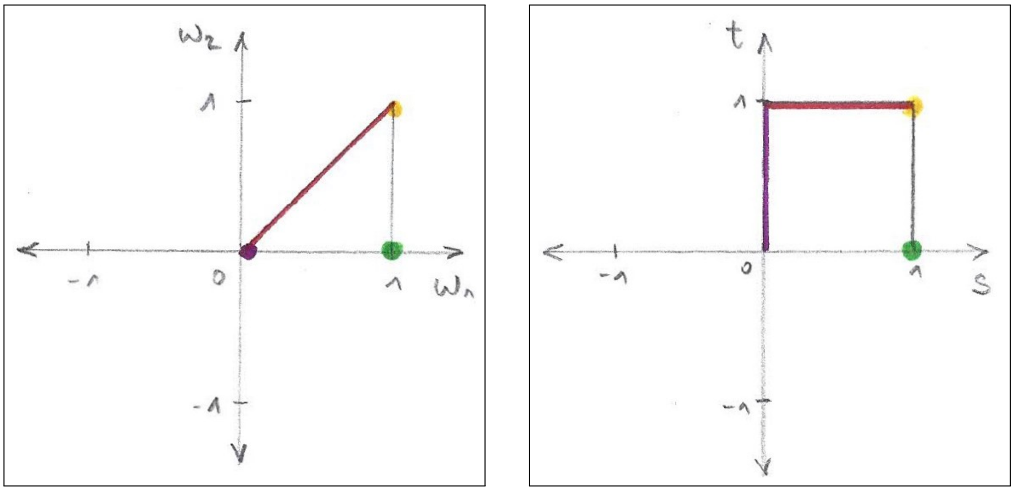
$$\begin{aligned} \gamma_i(t_\ell + H_\ell s) - \gamma_i(t_\ell + H_\ell t) &= \int_t^s \gamma_i'(t_\ell + H_\ell r) dr \\ &= H_\ell (s-t) \int_0^1 \gamma_i'(t_\ell + H_\ell t + \rho H_\ell (s-t))\, d\rho. \end{aligned}$$

Then, we consider

$$\begin{aligned} \kappa_1(s,t) &:= \frac{\gamma_i(t_\ell + H_\ell s) - \gamma_i(t_\ell + H_\ell t)}{|\gamma(t_\ell + H_\ell s) - \gamma(t_\ell + H_\ell t)|^2} n_j \\ &= \frac{1}{s-t} \frac{(\gamma_i(t_\ell + H_\ell s) - \gamma_i(t_\ell + H_\ell t))(s-t)}{|\gamma(t_\ell + H_\ell s) - \gamma(t_\ell + H_\ell t)|^2} n_j \\ &= \frac{1}{s-t} \frac{(s-t)^2 H_\ell \left(\int_0^1 \gamma_i'(t_\ell + H_\ell t + \rho H_\ell (s-t))\, d\rho \right)}{(s-t)^2 H_\ell^2 \left| \int_0^1 \gamma'(t_\ell + H_\ell t + \rho H_\ell (s-t))\, d\rho \right|^2} n_j \\ &= \frac{1}{s-t} \underbrace{\frac{\left(\int_0^1 \gamma_i'(t_\ell + H_\ell t + \rho H_\ell (s-t))\, d\rho \right)}{H_\ell \left| \int_0^1 \gamma'(t_\ell + H_\ell t + \rho H_\ell (s-t))\, d\rho \right|^2} n_j}_{=: f_1(s,t)} . \end{aligned} \tag{4.20}$$

We can see that if γ is $q-2$ times differentiable, then f_1 is $q-1$ times differentiable. Now,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) $w_1 := s$ and $w_2 := s - z$ (b) $w_1 \mapsto s$ and $w_2 \mapsto st$

Figure 4.3.: K, identical elements: Transformations step 3

we can rewrite

$$\begin{aligned}
\kappa_1(st, s) + \kappa_1(s, st) &= \frac{1}{st - s} f_1(st, s) + \frac{1}{s - st} f_1(s, st) \\
&= \left(\frac{1}{st - s} + \frac{1}{s - st} \right) f_1(st, s) - \frac{1}{s - st} (f_1(st, s) - f_1(s, st)) \\
&= -\frac{1}{s - st} (f_1(st, s) - f_1(s, st)) \\
&= \frac{1}{s - st} (f_1(s, st) - f_1(st, s)).
\end{aligned}$$

The function $c_{st}(\tau) = (st, s)^T + \tau(s - st, st - s)^T$ for $\tau \in [0, 1]$ describes the line segment from $(st, s)^T$ to $(s, st)^T$. Then, it holds that

$$\begin{aligned}
f_1(s, st) - f_1(st, s) &= \int_{(st,s)}^{(s,st)} Df_1(x)\, dx \\
&= \int_0^1 Df_1(c_{st}(\tau)) c'_{st}(\tau)\, d\tau \\
&= \int_0^1 Df_1(c_{st}(\tau)) \begin{pmatrix} s - st \\ st - s \end{pmatrix} d\tau \\
&= s(1 - t) \int_0^1 Df_1(c_{st}(\tau)) \begin{pmatrix} 1 \\ -1 \end{pmatrix} d\tau.
\end{aligned}$$

Since f_1 is smooth, the integral in the last step is also smooth. Consequently, it holds that

$$\kappa_1(st, s) + \kappa_1(s, st) = \int_0^1 Df_1(c_{st}(\tau)) \begin{pmatrix} 1 \\ -1 \end{pmatrix} d\tau.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

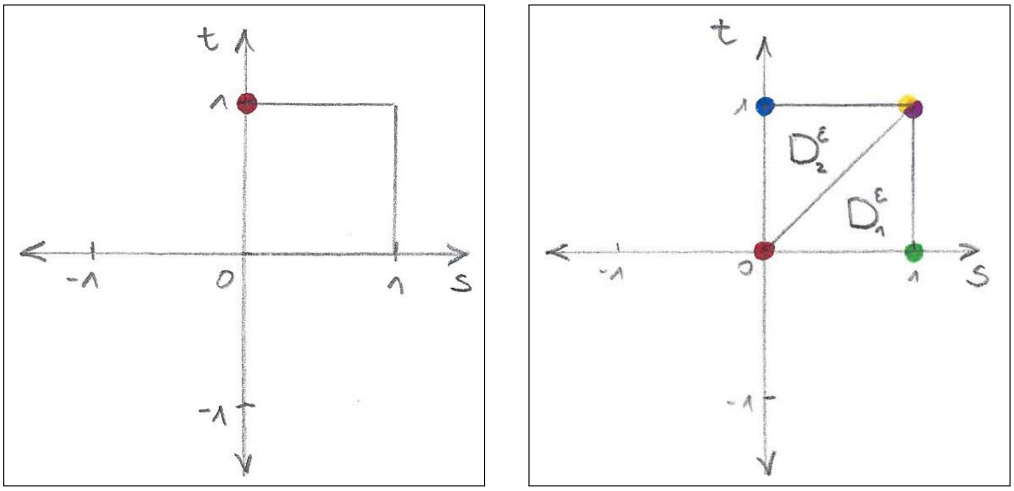
For the terms (II) and (III) we can analogously give a similar representation as in (4.20) and then proceed as before. The integral kernel κ defined in (4.17) is a sum of terms of type (I), (II) or (III) multiplied with $\boldsymbol{g} \circ \gamma$ and a basis function $\widetilde{R}_{i,\ell_1}$. Therefore, we can define a function $f(s,t) = (\kappa(st,s) + \kappa(s,st))\, s$, which is $q-2$ times differentiable. $\square$

For the remaining integral, we use Gauß quadrature with weight function 1 and with Corollary 4.4 also meet the requirements for Theorem 4.1.

Case 3. $|\gamma([t_{\ell_1-1}, t_{\ell_1}]) \cap \gamma([t_{\ell_2-1}, t_{\ell_2}])| = 1$: In this case, we have adjacent elements. Without loss of generality, we assume that the singularity in the integrand appears at $s = 0$ and $t = 1$. The other case can be treated analogously. We have either $t_{\ell_1-1} = t_{\ell_2}$ or $t_{\ell_2} = b \wedge t_{\ell_1-1} = a$. Using the transformation $(s,t) \mapsto (s, 1-t)$, it holds that

$$\begin{aligned}
&\lim_{\varepsilon\to 0} \int_{[0,1]} \int_{\{t\in[0,1]: d(s,t)_{\ell_1,\ell_2}\geq\varepsilon\}} \underbrace{\sum_{r=1}^{2} \left(\mathfrak{d}_{\boldsymbol{n},y} \breve{\boldsymbol{U}}^*_{\ell_1,\ell_2}(s,t)\right)_{kr} \widetilde{g}_{r,\ell_2}(t) \widetilde{R}_{i,\ell_1}(s)}_{=:\kappa(s,t)}\; dt\; ds \\
&= \underbrace{\lim_{\varepsilon\to 0} \int_{[0,1]} \int_{\{t\in[0,s]: d(s,1-t)_{\ell_1,\ell_2}\geq\varepsilon\}} \kappa(s,1-t)\; dt\; ds}_{\text{(I)}} \\
&\quad + \underbrace{\lim_{\varepsilon\to 0} \int_{[0,1]} \int_{\{t\in[s,1]: d(s,1-t)_{\ell_1,\ell_2}\geq\varepsilon\}} \kappa(s,1-t)\; dt\; ds,}_{\text{(II)}}
\end{aligned} \tag{4.21}$$

where we defined the integrand as $\kappa(s,t)$ for ease of notation.



(a) original domain $[0,1]^2$ (b) $t \mapsto 1-t$

Figure 4.4.: K, adjacent elements: Transformations step 1

In Figure 4.4, we show how the transformations in (4.21) act on the integral domain. The

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

singularity is indicated with a red dot. We divide the domain into two triangles

$$D_1^\varepsilon := \begin{Bmatrix} 0 \le s \le 1 \\ 0 \le t \le s \end{Bmatrix} \setminus U_\varepsilon \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad D_2^\varepsilon := \begin{Bmatrix} 0 \le s \le 1 \\ s \le t \le 1 \end{Bmatrix} \setminus U_\varepsilon \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Note that the singularity lies on the diagonal and therefore occurs in both triangles.

For the first summand (I) in (4.21), we apply the Duffy transformation

$$(\mathrm{I}) = \lim_{\varepsilon \to 0} \int_{[0,1]} \int_{\{t \in [0,s] : d(s,1-st)_{\ell_1,\ell_2} \ge \varepsilon\}} \kappa(s, 1-st) s \; dt \; ds. \tag{4.22}$$

As we can see from Figure 4.5 the singularity is extended from the point $(0,0)^T$ to the line segment $\{0\} \times [0,1]$ by the Duffy transformation.

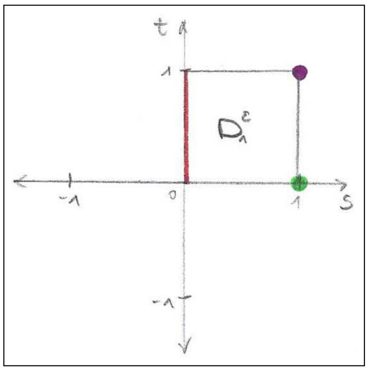


Figure 4.5.: K, adjacent elements: Transformations for first summand

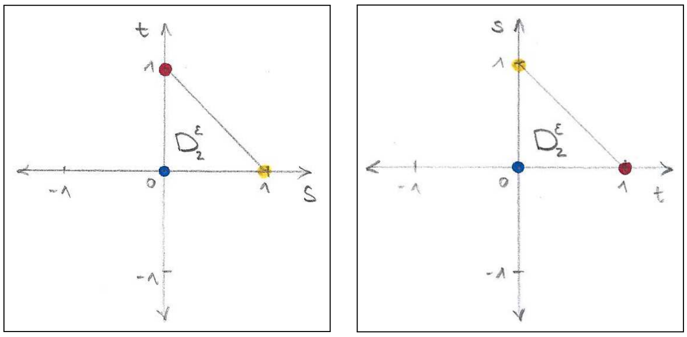
For the second summand (II) in (4.21), we first apply $t \mapsto 1-t$ and then apply Fubini's theorem to get the inner integral over the domain $[0, 1-t]$. Next, we apply the transformation $s \mapsto s(1-t)$ and then apply Fubini's theorem again. We obtain that

$$\begin{aligned} (\mathrm{II}) &= \lim_{\varepsilon \to 0} \int_{[0,1]} \int_{\{t \in [0,1-s] : d(s,t)_{\ell_1,\ell_2} \ge \varepsilon\}} \kappa(s,t) \; dt \; ds \\ &= \lim_{\varepsilon \to 0} \int_{[0,1]} \int_{\{s \in [0,1-t] : d(s,t)_{\ell_1,\ell_2} \ge \varepsilon\}} \kappa(s,t) \; ds \; dt \\ &= \lim_{\varepsilon \to 0} \int_{[0,1]} \int_{\{t \in [0,1] : d(s(1-t),t)_{\ell_1,\ell_2} \ge \varepsilon\}} \kappa(s(1-t),t)(1-t) \; dt \; ds. \end{aligned} \tag{4.23}$$

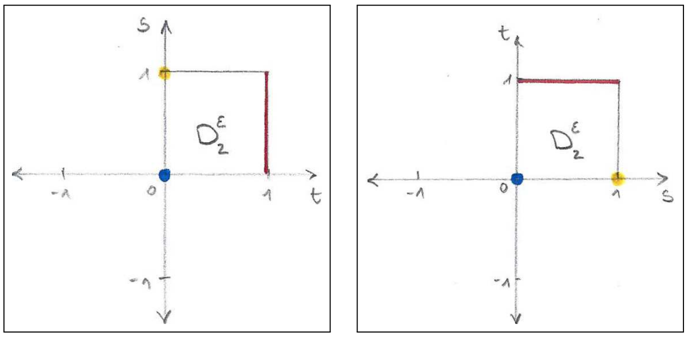
In Figure 4.6, we can see that the above transformations transform the singularity from a point to a line. Note however, that the first and the second integral from (4.21) do not have the singularities on the same line segment in the end.

The following corollary strongly relies on [Gan14, Lemma 5.5].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) $t \mapsto 1-t$ (b) change integration order



(c) $s \mapsto s(1-t)$ (d) change integration order

Figure 4.6.: K, adjacent elements: Transformations for second summand

Corollary 4.5. *If the parametrization* γ *is* $q \geq 2$ *times continuously differentiable on* $[\check{x}_{j-1}^\gamma, \check{x}_j^\gamma]$ *for* $j = 1, \ldots, n_\gamma$ *and if* $\boldsymbol{g} \circ \gamma$ *is* $q-1$ *times continuously differentiable on* $[\check{x}_{j-1}^\gamma, \check{x}_j^\gamma]$ *for* $j = 1, \ldots, n_\gamma$*, the integrands in* (4.22) *and* (4.23) *are* $q-1$ *times continuously partially differentiable on* $[0,1]^2$.

Proof. We prove the assertion for (4.22), while (4.23) can be treated analogously. First, we assume that $t_{\ell_1 - 1} = t_{\ell_2}$. As we can see with (2.31) and (4.3), the conormal derivative is a sum of terms of the following three different types

$$\underbrace{\frac{x_i - y_i}{|\boldsymbol{x} - \boldsymbol{y}|^2} n_j}_{=:(\mathrm{I})}, \quad \underbrace{\frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)}{|\boldsymbol{x} - \boldsymbol{y}|^4} n_r}_{=:(\mathrm{II})}, \quad \underbrace{\frac{\delta_{ik}(x_i - y_i) + \delta_{ij}(x_k - y_k)}{|\boldsymbol{x} - \boldsymbol{y}|^2} n_r}_{=:(\mathrm{III})}, \tag{4.24}$$

for $\boldsymbol{x} = \gamma(t_{\ell_1} + H_{\ell_1} s)$ and $\boldsymbol{y} = \gamma(t_{\ell_2} + H_{\ell_2}(1 - st))$ and for $i, j, k, r = 1, 2$. Therefore, the integrand occurring in (4.22) is composed of the above terms multiplied by the Jacobi

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

determinant s. For the transformed term (I) and for $s \in (0,1], t \in [0,1]$, it holds that

$$\begin{aligned}
&\frac{\gamma_i(t_{\ell_1}+H_{\ell_1}s)-\gamma_i(t_{\ell_2}+H_{\ell_2}(1-st))}{|\gamma(t_{\ell_1}+H_{\ell_1}s)-\gamma(t_{\ell_2}+H_{\ell_2}(1-st))|^2}\cdot s\cdot n_j \\
&=\frac{s^2}{|\gamma(t_{\ell_1}+H_{\ell_1}s)-\gamma(t_{\ell_2}+H_{\ell_2}(1-st))|^2}\cdot\frac{\gamma_i(t_{\ell_1}+H_{\ell_1}s)-\gamma_i(t_{\ell_2}+H_{\ell_2}(1-st)}{s}\cdot n_j \\
&=\frac{s^2}{\left|\int_{[t_{\ell_2}-H_{\ell_2}st,t_{\ell_2}+H_{\ell_1}s]}\gamma'(\tau)d\tau\right|^2}\cdot\frac{\int_{[t_{\ell_2}-H_{\ell_2}st,t_{\ell_2}+H_{\ell_1}s]}\gamma_i'(\tau)d\tau}{s}\cdot n_j \\
&=\frac{\int_{[-H_{\ell_2}t,H_{\ell_1}]}\gamma_i'(t_{\ell_2}+s\tau)d\tau}{\left|\int_{[-H_{\ell_2}t,H_{\ell_1}]}\gamma'(t_{\ell_2}+s\tau)d\tau\right|^2}\cdot n_j. \qquad (4.25)
\end{aligned}$$

For the case $s \to 0$, we rewrite

$$\begin{aligned}
\lim_{s\to 0}\int_{[-H_{\ell_2}t,H_{\ell_1}]}\gamma_i'(t_{\ell_2}+s\tau)d\tau &= \lim_{s\to 0}\int_{[-H_{\ell_2}t,0]}\gamma_i'(t_{\ell_2}+s\tau)d\tau+\lim_{s\to 0}\int_{[0,H_{\ell_1}]}\gamma_i'(t_{\ell_2}+s\tau)d\tau \\
&=\int_{[-H_{\ell_2}t,0]}\gamma_i'^{\ell}(t_{\ell_2})d\tau+\int_{[0,H_{\ell_1}]}\gamma_i'^{r}(t_{\ell_2})d\tau \\
&=H_{\ell_2}t\gamma_i'^{\ell}(t_{\ell_2})+H_{\ell_1}\gamma_i'^{r}(t_{\ell_2}),
\end{aligned}$$

where $\gamma_i'^{\ell}$ resp. $\gamma_i'^{r}$ denotes the left resp. right derivative of γ. Therefore, and since $\gamma_i'^{r}(t_{\ell_2})$ is not a negative multiple of $\gamma'\ell_i(t_{\ell_2})$, (4.25) can be continuously extended at $s = 0$ with

$$\frac{H_{\ell_2}t\gamma_i'^{\ell}(t_{\ell_2})+H_{\ell_1}\gamma_i'^{r}(t_{\ell_2})}{|H_{\ell_2}t\gamma'^{\ell}(t_{\ell_2})+H_{\ell_1}\gamma'^{r}(t_{\ell_2})|^2}\cdot n_j.$$

The above argumentation can be analogously applied for (II) and (III) from (4.24). Therefore, the conormal derivative also has the desired regularity and by multiplying with $\widetilde{g}_{r,\ell_2}(1-st)\widetilde{R}_{i,\ell_1}(s)$ we obtain the integrand in (4.22), which as a consequence has the desired regularity. In the case that $t_{\ell_2} = b$ and $t_{\ell_1-1} = a$ one can shift t_{ℓ_1} to $t_{\ell_1-1}+(b-a) = b$ the proof works as before. ☐

Corollary 4.5 states continuity for the integrands in (4.22) and (4.23). Therefore, we have proven existence of the limit $\lim_{\varepsilon\to 0}$ in (4.22)–(4.23) and also the requirements for Theorem 4.1. Overall, we can thus use Gauß quadrature with weight function 1.

4.2. Validation of code with numerical examples

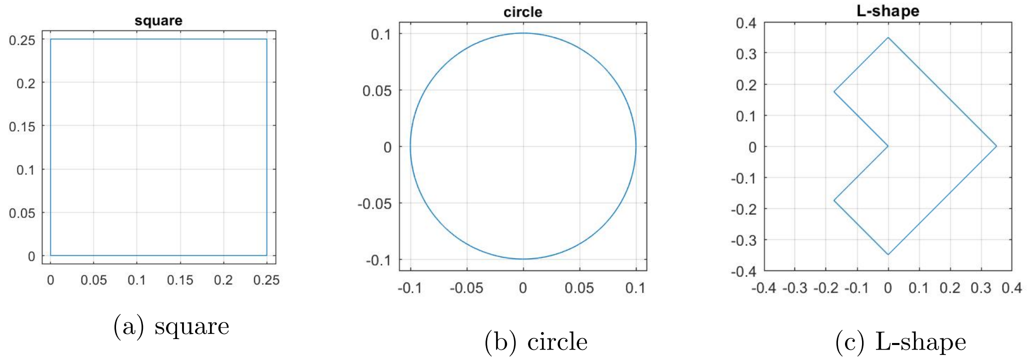
In this section, we validate the implementation of the integral operators V resp. K described in Section 4.1.1 resp. 4.1.2 using different examples. In all the examples, we will consider a Dirichlet boundary value problem as in (2.21). We therefore seek a solution $\boldsymbol{\phi} \in \boldsymbol{H}^{-1/2}(\Gamma)$ to Symm's integral equation (2.37). We then use the Galerkin method to find an approximate solution $\Phi_h \in X_h$ of (3.4), i.e., $(\Phi_h, \Psi_h)_V = ((1/2I + K)\boldsymbol{g}, \Psi_h)_\Gamma$ for all $\Psi_h \in X_h$. To this end, we proceed as described in Section 4.1.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The approximation spaces we use are $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ with given knots $\check{\mathcal{K}}_h$ and weights $\mathcal{W}_h$ and in some cases also $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$. The main idea of isogeometric analysis is to use the same NUBRS functions for the parametrization of the geometry as for the ansatz spaces. For the geometry, we have a polynomial degree $p_\gamma \in \mathbb{N}$, knots $\check{\mathcal{K}}^\gamma$ and weights $\mathcal{W}^\gamma$ of length N_γ. Furthermore, we have control points $(C_i)_{i=1-p}^{N_\gamma-\#b+1} \in \mathbb{R}^2$. Then, as in (4.2) we have

$$\gamma(t) = \sum_{i=1-p}^{N_\gamma-\#b+1} C_i R_{i,p_\gamma}^{\check{\mathcal{K}}_\gamma, \mathcal{W}_\gamma}(t) \quad \text{for all } t \in [a,b]. \tag{4.26}$$

Hence, for the polynomial degree p, the initial knots $\check{\mathcal{K}}_0$ and weights $\mathcal{W}_0$ for the initial ansatz space $\boldsymbol{\mathcal{S}}(\mathcal{T}_0)$ in Algorithm 3.5, we choose $p := p_\gamma$, $\check{\mathcal{K}}_0 := \check{\mathcal{K}}^\gamma$ and $\mathcal{W}_0 := \mathcal{W}^\gamma$. In some cases we also use $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ as an ansatz space. In this case, we chose the initial knots $\check{\mathcal{K}}^0$ the same as $\check{\mathcal{K}}^\gamma$ with the modification that we increase the multiplicity of each node to $p+1$. The initial weights $\mathcal{W}^0$ are all equal to 1. For the parameters of the Lamé equation, we used $\lambda = 0.4$ and $\mu = 0.4$ in accordance with Section 1.1.



(a) square

(b) circle

(c) L-shape

Figure 4.7.: Geometries for code validation

We perform uniform and adaptive refinement according to Algorithm 3.5 with adaptivity parameter $\theta = 0.9$. According to [SS11, Corollary 4.1.34] one can expect the convergence rate $\mathcal{O}(h^{3/2+p}) = \mathcal{O}(n^{-3/2-p})$ for the error and $(h\text{–}h/2)$–estimator for uniform refinement and smooth solution ϕ for the Laplace equation. For the Lamé equation, we expect the same convergence rate. The proof relies on the Céa-Lemma and some approximation property.

Another strategy that we use for mesh refinement is a special algorithm, that refines the mesh asymmetrically and concentrates the refinement on particular sections of the boundary. In this way, we can test how the implementation of the operators performs under difficult conditions.

Algorithm 4.6. INPUT : *initial mesh* $[\mathcal{T}_0]$*, initial weights* $\mathcal{W}_0$*, polynomial degree* $p \in \mathbb{N}_0$ *and the number of steps* $\ell := 0$.

1. *We mark the first and the last element of all the elements sorted accordingly to the parametrization* γ*. Additionally, if the number of steps* $\ell \geq 2$*, then we mark the*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

element which corresponds to the floor of the 75%-quantile $\lfloor q_{75} \rfloor$ of the number of nodes n.

2. *We then follow steps 4.–7. from Algorithm 3.5.*
3. *Update $\ell \leftarrow \ell + 1$ and go to step 1.*

OUTPUT : *refined mesh $\mathcal{T}_\ell$.*

4.2.1. Indirect BEM

When seeking solutions to the Dirichlet problem (2.21), another approach as opposed to the direct approach, is to use the so-called *indirect approach*. We look for solutions of the form

$$\boldsymbol{u} := \mathcal{V}\boldsymbol{\phi}$$

for $\boldsymbol{\phi} \in \boldsymbol{H}^{-1/2}(\Gamma)$. According to Remark 2.19 $\boldsymbol{u}$ solves $\boldsymbol{\mathcal{L}u} = \boldsymbol{0}$. In order to fulfil the given boundary conditions $\boldsymbol{u}|_\Gamma = \boldsymbol{g}$ for an arbitrary $\boldsymbol{g} \in \boldsymbol{H}^{1/2}(\Gamma)$, we obtain that $\boldsymbol{\phi}$ is $\boldsymbol{\phi} := V^{-1}\boldsymbol{g}$. Therefore, we have to solve a weakly singular integral equation

$$V\boldsymbol{\phi} = \boldsymbol{g} \quad \text{on } \Gamma. \tag{4.27}$$

Note that $\boldsymbol{\phi}$ in this case is not the conormal derivative of the solution $\boldsymbol{u}$ and does not have a natural physical interpretation. For this reason, we cannot easily calculate the corresponding $\boldsymbol{\phi}$, unless the solution $\boldsymbol{u}$ is prescribed and hence known.

Again, we can use the Galerkin method to find an approximate solution Φ_h in some finite dimensional subspace $X_h \subset \boldsymbol{H}^{-1/2}(\Gamma)$. To this end, we solve $(\Phi_h, \Psi_h)_V = (\boldsymbol{g}, \Psi_h)_\Gamma$ for all $\Psi_h \in X_h$ and then proceed as in Section 4.1.

4.2.2. Validation of K

We first validate our implementation of the right hand side F_h. The difficulty for the implementation of F_h lies mostly in the implementation of Kg_h. Moreover, computing Kg_h also takes up the largest part of the computational time.
We consider the rigid body motions $\boldsymbol{r} \in \mathcal{R}$ with their basis $(\boldsymbol{r}_k)_{k=1}^{\dim \mathcal{R}}$ as defined in (2.24). In the paragraph after (2.24), we explained that $\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{r} = 0$. Since $\boldsymbol{r}_k$, for $k = 1, \dots, \dim \mathcal{R}$ are a solution to the Lamé equation (2.16), we can construct a Dirichlet problem. We assume that $\boldsymbol{r}_k$ is a solution to (2.21) with $\boldsymbol{g} := \gamma\boldsymbol{r}_k$. We obtain that $V\mathfrak{d}_{\boldsymbol{n}}\boldsymbol{r}_k = (K+1/2)\gamma\boldsymbol{r}_k = 0$. Since V_h is a regular matrix, we can perform numerical tests to see whether the corresponding ϕ_h and F_h are equal to the zero vector.

We consider the square with edge length 0.25 (see Figure 4.7a) and then solve Symm's integral equation

$$V\boldsymbol{\phi} = \left(K + \frac{1}{2}\right)\gamma\boldsymbol{r}_k.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The boundary of the geometry is parametrized on $[0,1]$ by the NURBS curve induced by

$$\begin{aligned} p_\gamma &= 1, \\ \check{\mathcal{K}}^\gamma &= \left(\frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1, 1\right), \\ \mathcal{W}^\gamma &= (1,1,1,1,1), \\ (C_i)_{i=1-p}^{N_\gamma - \#b+1} &= \frac{1}{4}\left(\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0\\0\end{pmatrix}\right). \end{aligned} \tag{4.28}$$

As ansatz spaces, we consider $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$, where we use the knots $\check{\mathcal{K}}^\gamma$ and weights $\mathcal{W}^\gamma$ from (4.28) for the initial mesh $\mathcal{T}_0$, and then perform mesh refinement according to Algorithm 4.6. Furthermore, we also consider $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0,1,2\}$ as ansatz space and refinement according to Algorithm 4.6. In Figure 4.8 we can see how Algorithm 4.6 refines the mesh on the square after 13 steps with a result of $N = 50$ knots. The algorithm does refine the mesh asymmetrically and concentrates the refinement on a certain area of the boundary.

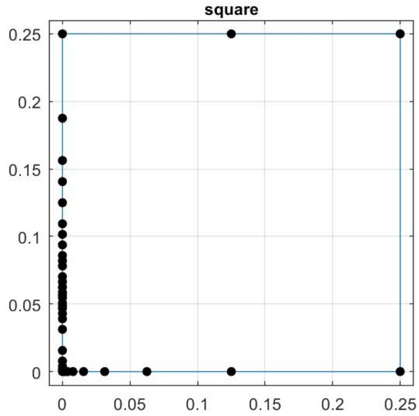


Figure 4.8.: Nodes on square after 13 steps of mesh refinement according to Algorithm 4.6 with resulting number of knots $N = 50$ for the validation of K from Section 4.2.2

First, we consider a constant $\boldsymbol{g}(x_1,x_2) = (1,1)^T$ for $(x_1,x_2)^T \in \Gamma$. As we can see from Table 4.1, the norm of the right–hand side vector $\|F_h\|_2$ is already very small from the beginning onwards for all tested ansatz spaces $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0,1,2\}$ and $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$. The same holds for the energy norms of the approximate solution $|||\Phi_h|||$ and the $(h$–$h/2)$–estimator $\widetilde{\mu}(\mathcal{T}_h)$. Also, after several steps of very concentrated refinement, the norms and also the $(h$–$h/2)$–estimator slightly increase. This may be explained by the fact that the mesh is strongly adapted with very small element size and therefore small instabilities arise. Another fact worth noting is that the the energy norm of the approximate solution $|||\Phi_h|||$ is significantly smaller than the norm of the right hand side $\|F_h\|$.

Second, we consider $\boldsymbol{g}(x_1,x_2) = (-x_2,x_1)^T$ for $(x_1,x_2)^T \in \Gamma$, which corresponds to a rotation of 90° in mathematical positive direction. In Table 4.2, we can see the results

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Ansatz space	N	$\|F_h\|_2$	$|||\Phi_h|||$	$\widetilde{\mu}(\mathcal{T}_h)$
$\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$	5	$\mathcal{O}(10^{-16})$	$\mathcal{O}(10^{-30})$	$\mathcal{O}(10^{-14})$
	53	$\mathcal{O}(10^{-14})$	$\mathcal{O}(10^{-24})$	$\mathcal{O}(10^{-11})$
$\boldsymbol{\mathcal{P}}^0(\mathcal{T}_h)$	4	$\mathcal{O}(10^{-16})$	$\mathcal{O}(10^{-30})$	$\mathcal{O}(10^{-14})$
	50	$\mathcal{O}(10^{-14})$	$\mathcal{O}(10^{-24})$	$\mathcal{O}(10^{-11})$
$\boldsymbol{\mathcal{P}}^1(\mathcal{T}_h)$	8	$\mathcal{O}(10^{-16})$	$\mathcal{O}(10^{-30})$	$\mathcal{O}(10^{-14})$
	52	$\mathcal{O}(10^{-14})$	$\mathcal{O}(10^{-23})$	$\mathcal{O}(10^{-11})$
$\boldsymbol{\mathcal{P}}^2(\mathcal{T}_h)$	12	$\mathcal{O}(10^{-16})$	$\mathcal{O}(10^{-29})$	$\mathcal{O}(10^{-14})$
	57	$\mathcal{O}(10^{-14})$	$\mathcal{O}(10^{-22})$	$\mathcal{O}(10^{-11})$

Table 4.1.: Results for Dirichlet data $\boldsymbol{g} = (1,1)^T$ on initial and refined mesh (according to Algorithm 4.6) for the validation of K from Section 4.2.2

for the ansatz spaces $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0,1,2\}$ and $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$. As for constant $\boldsymbol{g}$, we see that $\|F_h\|_2$, $|||\Phi_h|||$ and $\widetilde{\mu}(\mathcal{T}_h)$ are small already on the first grid and increase slightly with mesh refinement according to Algorithm 4.6.

Ansatz space	N	$\|F_h\|_2$	$|||\Phi_h|||$	$\widetilde{\mu}(\mathcal{T}_h)$
$\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$	5	$\mathcal{O}(10^{-17})$	$\mathcal{O}(10^{-32})$	$\mathcal{O}(10^{-15})$
	53	$\mathcal{O}(10^{-15})$	$\mathcal{O}(10^{-26})$	$\mathcal{O}(10^{-13})$
$\boldsymbol{\mathcal{P}}^0(\mathcal{T}_h)$	4	$\mathcal{O}(10^{-17})$	$\mathcal{O}(10^{-31})$	$\mathcal{O}(10^{-15})$
	50	$\mathcal{O}(10^{-15})$	$\mathcal{O}(10^{-26})$	$\mathcal{O}(10^{-13})$
$\boldsymbol{\mathcal{P}}^1(\mathcal{T}_h)$	8	$\mathcal{O}(10^{-17})$	$\mathcal{O}(10^{-32})$	$\mathcal{O}(10^{-15})$
	52	$\mathcal{O}(10^{-15})$	$\mathcal{O}(10^{-25})$	$\mathcal{O}(10^{-12})$
$\boldsymbol{\mathcal{P}}^2(\mathcal{T}_h)$	12	$\mathcal{O}(10^{-17})$	$\mathcal{O}(10^{-31})$	$\mathcal{O}(10^{-15})$
	57	$\mathcal{O}(10^{-15})$	$\mathcal{O}(10^{-24})$	$\mathcal{O}(10^{-12})$

Table 4.2.: Results for Dirichlet data $\boldsymbol{g} = (-x_2, x_1)^T$ on initial and refined mesh (according to Algorithm 4.6) for the validation of K from Section 4.2.2

In conclusion, we see that the operator K shows correct results for for the rigid body motions $\boldsymbol{r} \in \mathcal{R}$ on the square.

4.2.3. Validation of V

In order to validate the correct implementation of the operator V, the following result for the Laplace equation suggests that a similar behaviour can also be expected for the Lamé equation.

Corollary 4.7. *For Ω being the circle with radius $r > 0$ and midpoint in the origin, we set $\Gamma := \partial\Omega$ the circular line. We consider the homogeneous Laplace equation with constant*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Dirichlet boundary data, i.e.,

$$\begin{aligned} -\Delta u &= 0 \quad \text{in } \Omega \\ u &= 1 \quad \text{on } \Gamma. \end{aligned} \tag{4.29}$$

When following an indirect approach as in Section 4.2.1, we obtain the equation $V\phi = 1$*. Then, the solution* $\phi \in H^{-1/2}(\Gamma)$ *is constant.*

Proof. As a first step, we show that $K(1) = -1/2$ for any $c \in \mathbb{R}$. To this end, we consider the representation formula (see (2.32) for the Lamé equation), which holds accordingly for the Laplace equation when replacing the fundamental solution of the Lamé equation with the scalar valued fundamental solution of the Laplace equation and the conormal derivative $\mathfrak{d}_{\boldsymbol{n}} u$ with the normal derivative $\partial_{\boldsymbol{n}} u$. The representation formula reads

$$u = \mathcal{V}\phi - \mathcal{K}g, \tag{4.30}$$

with $g := u|_\Gamma$ and $\phi := \partial_{\boldsymbol{n}} u$. The function $u \equiv 1$ solves the Laplace equation (4.29) with $g \equiv 1$ and $\partial_{\boldsymbol{n}} u \equiv 0$. Next, we apply the external trace operator γ^{ext} on the representation formula (4.30) to obtain that

$$u|_\Gamma = V\phi - \left(K - \frac{1}{2}\right) g,$$

with $V := \gamma^{\text{ext}}\mathcal{V}$ and $K := \gamma^{\text{ext}}\mathcal{K} + 1/2$. When inserting for $u \equiv 1$, $g \equiv 1$ and $\phi \equiv 0$, we obtain that $K(1) = -1/2$.
As a next step, we consider an exterior boundary value problem, i.e.,

$$\begin{aligned} -\Delta u &= 0 \quad \text{in } \overline{\Omega}^C, \\ u &= g \quad \text{ on } \Gamma, \\ \mathcal{M}u &= 0 \quad \text{on } \mathbb{R}^2, \end{aligned} \tag{4.31}$$

with $g \in H^{1/2}(\Gamma)$ and where the condition $\mathcal{M}u = 0$ incorporates some assumptions about the behaviour of the solution at infinity (cf. [McL00, Chapter 7, Exterior Problems]). Let $g = -c$ for some $c \in \mathbb{R}$. We know that the fundamental solution of the Laplace equation

$$U(x) = -\frac{1}{2\pi} \log |x|$$

solves $-\Delta U = 0$. According to [McL00, Lemma 7.13], the fundamental solution also fulfils $\mathcal{M}U = 0$ in $\mathbb{R}^2$. Since the fundamental solution is radial symmetric, i.e., $U = U(|x|)$, which means that the argument only occurs as an absolute value, it holds that $U = d := -(1/(2\pi))\log(r)$ on Γ. By considering $\widetilde{u} := -(c/d)U$ we have found a solution to (4.31). According to [McL00, Lemma 7.15(i)], the problem (4.31) is equivalent to the boundary integral equation

$$V\phi = \left(K - \frac{1}{2}\right) g.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Let $c := 1 \in \mathbb{R}$ and $g = -1$. Together with $K(-1) = 1/2$ we then obtain $V\phi = 1$. The solution is then $\widetilde{u} = (1/d)U$ and $\phi = \partial_{\boldsymbol{n}}\widetilde{u}$. There holds

$$\partial_{\boldsymbol{n}} U(x) = -\frac{1}{2\pi}\frac{x\cdot \boldsymbol{n}}{|x|^2}$$

and consequently

$$\phi(x) = -\frac{x\cdot \boldsymbol{n}(x)}{2\pi d r^2}.$$

Since the normal vector $\boldsymbol{n}(x)$ in point x is simply $x/|x|$, it follows that $x\cdot\boldsymbol{n}(x) = (x\cdot x)/|x| = |x| = r$. Hence, ϕ is constant. ☐

The above result cannot easily be applied to the Lamé equation, as we were not able to show that the fractional term in the fundamental solution (2.30) of the Lamé equation is radial symmetric. However, our numerical results show that Φ_h seems to be constant.

Following the indirect approach we therefore consider

$$V\boldsymbol{\phi} = \begin{pmatrix}1\\1\end{pmatrix},$$

on the circle with radius $r = 1/10$, see Figure 4.7b. The boundary of the geometry is parametrized on $[0,1]$ by the NURBS curve induced by

$$\begin{aligned}
p_\gamma &= 2,\\
\check{\mathcal{K}}^\gamma &= \left(\frac{1}{4},\frac{1}{4},\frac{2}{4},\frac{2}{4},\frac{3}{4},\frac{3}{4},1,1,1\right),\\
\mathcal{W}^\gamma &= \left(1,\frac{1}{\sqrt{2}},1,\frac{1}{\sqrt{2}},1,\frac{1}{\sqrt{2}},1,1,\frac{1}{\sqrt{2}}\right),\\
(C_i)_{i=1-p}^{N_\gamma-\#b+1} &= \frac{1}{10}\left(\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}1\\1\end{pmatrix}\right).
\end{aligned} \tag{4.32}$$

As ansatz spaces, we consider $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$, where we use the knots $\check{\mathcal{K}}^\gamma$ and weights $\mathcal{W}^\gamma$ from (4.28) for the initial mesh $\mathcal{T}_0$, and then perform uniform mesh refinement and refinement according to Algorithm 4.6. Furthermore, we also consider $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0,1,2\}$ as ansatz space and refinement according to Algorithm 4.6. The results are presented in Table 4.3.

We see from the results that the $(h$–$h/2)$–estimator is already very small on the initial mesh for all tested ansatz spaces. Similar to the examples in Section 4.2.2, we see that the values for the estimator slightly rise with refinement. Only for the ansatz space $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$, the strongly adaptive refinement causes the estimator to increase in value. One possible explanation might be that the error occurs due to cancellation effects.

The energy norm of the approximate solution $|\!|\!|\Phi_h|\!|\!|$ stays relatively stable and is not very vulnerable to strongly adaptive refinement or different ansatz spaces. Changes only occur in the 15th digit.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

p	Refinement	N	$\lVert\Phi_h\rVert$	$\widetilde{\mu}(\mathcal{T}_h)$
$\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$	none	9	2.953798637040423	$\mathcal{O}(10^{-14})$
	uniform	133	2.953798637040424	$\mathcal{O}(10^{-13})$
	Algorithm 4.6	105	2.953798637040426	$\mathcal{O}(10^{-8})$
$\boldsymbol{\mathcal{P}}^0(\mathcal{T}_h)$	none	4	2.953798637040427	$\mathcal{O}(10^{-14})$
	uniform	128	2.953798637040423	$\mathcal{O}(10^{-14})$
	Algorithm 4.6	102	2.953798637040424	$\mathcal{O}(10^{-13})$
$\boldsymbol{\mathcal{P}}^1(\mathcal{T}_h)$	none	8	2.953798637040427	$\mathcal{O}(10^{-14})$
	uniform	128	2.953798637040425	$\mathcal{O}(10^{-13})$
	Algorithm 4.6	100	2.953798637040426	$\mathcal{O}(10^{-12})$
$\boldsymbol{\mathcal{P}}^2(\mathcal{T}_h)$	none	12	2.953798637040426	$\mathcal{O}(10^{-13})$
	uniform	192	2.953798637040425	$\mathcal{O}(10^{-12})$
	Algorithm 4.6	102	2.953798637040424	$\mathcal{O}(10^{-11})$

Table 4.3.: Results for Dirichlet data $\boldsymbol{g} = (1,1)^T$ on circle for $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0,1,2\}$ and $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ on initial and refined mesh (uniformly or according to Algorithm 4.6) for the validation of V from Section 4.2.3

In Figure 4.9, the solution Φ_h with ansatz space $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ is plotted on the initial mesh and on the uniformly refined mesh with $N = 133$. We see how the constant solution is approximated using quadratic ansatz functions. On the initial mesh, we clearly see the break points, i.e., the nodes of the geometry Γ, where Φ_h is only continuous, but not differentiable. On the refined mesh, it is clearly visible that Φ_h is approximating a constant function, but there are still some oscillations at the break points.

4.2.4. Combined validation for V and K

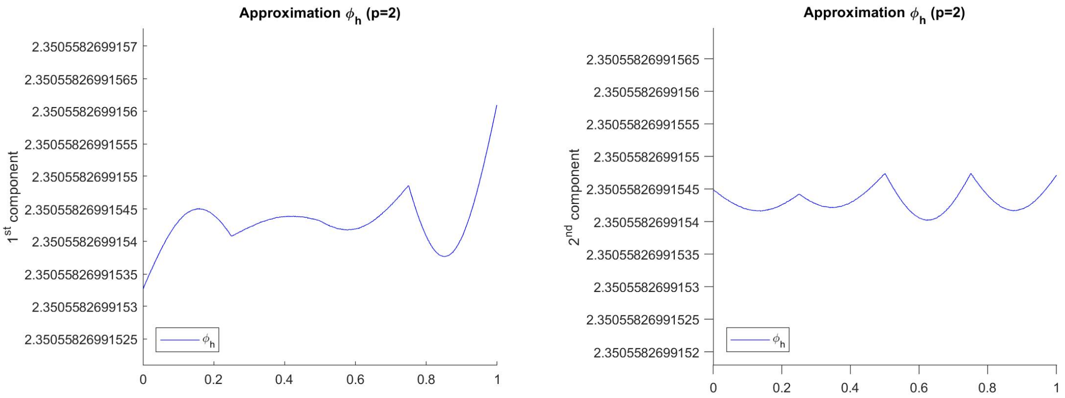
As a last validation, we consider the direct approach for the Dirichlet boundary value problem with a known solution $\boldsymbol{u}$. Since the column vectors of the fundamental solution $\boldsymbol{U}_1^*$, $\boldsymbol{U}_2^*$ (cf. Theorem 2.18) also solve the homogeneous Lamé equation (2.16), we may consider them as a solution. However, as the fundamental solution has a singularity in the origin, we shift the singularity to a point outside $\overline{\Omega}$. We define a shift vector $\boldsymbol{s} := (10,0)^T$ and set

$$\boldsymbol{u}(x) := \boldsymbol{U}_1^*(x + \boldsymbol{s}) \quad \text{for } x \in \Omega.$$

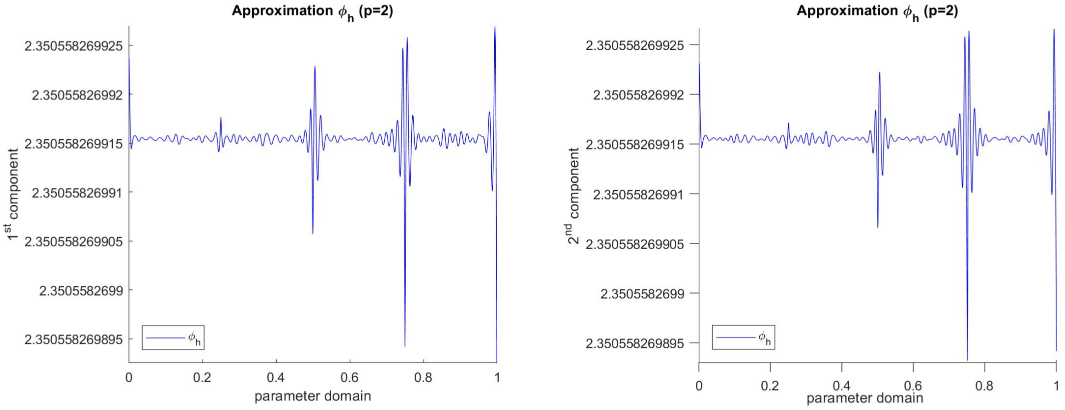
The corresponding Dirichlet data $\boldsymbol{g} = \boldsymbol{u}|_\Gamma$ are then obtained by evaluating $\boldsymbol{u}(x)$ at the boundary. The conormal derivative $\boldsymbol{\phi}$ is then approximated using Symm's integral equation. As we can calculate the conormal derivative $\boldsymbol{\phi} = \mathfrak{d}_{\boldsymbol{n}}\boldsymbol{u}$, we can compare the approximation Φ_h with the exact solution $\boldsymbol{\phi}$. The ℓ-th partial derivative of $\boldsymbol{U}_1^* = (U_{11}^*, U_{21}^*)^T$ is given by

$$\partial_{\ell,y} U_{i1}^*(x) = -\frac{3\mu+\lambda}{4\pi\mu(2\mu+\lambda)}\frac{x_\ell}{|x|^2}\delta_{1i} + \frac{\mu+\lambda}{4\pi\mu(2\mu+\lambda)}\frac{(\delta_{\ell 1}x_i + \delta_{\ell i}x_1)\,|x|^2 - 2x_\ell x_1 x_i}{|x|^4}.$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) 1^{st} component, initial mesh (b) 2^{nd} component, initial mesh



(c) 1^{st} component, refined mesh (d) 2^{nd} component, refined mesh

Figure 4.9.: Φ_h on initial and uniformly refined mesh with $N = 133$ for $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ for the validation of V from Section 4.2.3

The conormal derivative of $\boldsymbol{U}_1^*$ then reads

$$(\mathfrak{d}_{\boldsymbol{n}} \boldsymbol{U}_1^*)_i = \sum_{j=1}^{2} \sigma_{ij}(\boldsymbol{U}_1^*) n_j = \sum_{j=1}^{2} \left(\lambda \delta_{ij} (\partial_1 U_{11}^* + \partial_2 U_{21}^*) + \mu (\partial_i U_{j1}^* + \partial_j U_{i1}^*)\right) n_j.$$

We consider the boundary of the so called L-shape (cf. Figure 4.7c). The boundary of

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the geometry is parametrized on $[0,1]$ by the NURBS curve induced by

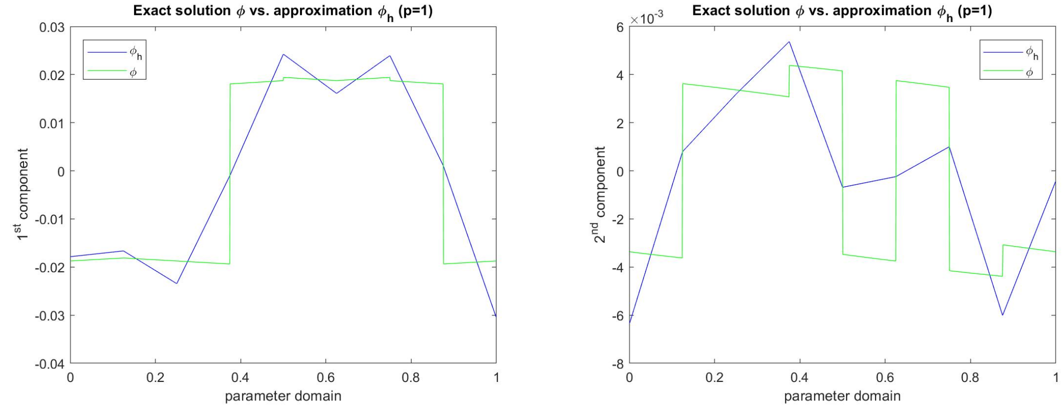
$$\begin{aligned} p_\gamma &= 1,\\ \check{\mathcal{K}}^\gamma &= \left(\frac{1}{8},\frac{2}{8},\frac{3}{8},\frac{4}{8},\frac{5}{8},\frac{6}{8},\frac{7}{8},1,1\right),\\ \mathcal{W}^\gamma &= (1,1,1,1,1,1,1,1,1),\\ (C_i)_{i=1-p}^{N_\gamma-\#b+1} &= \frac{7}{40}\left(\begin{pmatrix}2\\0\end{pmatrix},\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}0\\2\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix},\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}0\\-2\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix}\right). \end{aligned} \tag{4.33}$$

As ansatz spaces, we consider $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$, where we use the knots $\check{\mathcal{K}}^\gamma$ and weights $\mathcal{W}^\gamma$ from (4.28) for the initial mesh $\mathcal{T}_0$, and then perform uniform mesh refinement and adaptive refinement according to Algorithm 3.5 with the adaptivity constant $\theta = 0.9$. We use the $(h$–$h/2)$–estimator local contributions $\widetilde{\mu}(z)$ for $z \in \mathcal{N}_h$ to steer the adaptive refinement.

In Figure 4.10, we see the exact solution plotted against the approximation on the initial mesh and on the refined mesh, once for uniform refinement and once for adaptive refinement. As we chose the node multiplicity equal to $p = 1$ for all nodes except the last node, the approximation Φ_h is continuous at all those nodes. In Figure 4.10 the exact solution ϕ appears to be piecewise linear and has discontinuities at the corners of the L-shape geometry. The discontinuities stem from the fact that the normal vector jumps when passing over a corner. In Figure 4.10a and 4.10b we see that the piecewise linear, globally continuous function Φ_h cannot approximate the discontinuous ϕ very well. As we see in Figure 4.10c-4.10f, the approximation improves with refinement, in particular for adaptive refinement, which refines the mesh close to the corners. If we were to choose the nodes of the initial grid with multiplicity $p+1$ at every corner of the geometry, we would be able to also approximate the discontinuities and thus have a better approximation. For demonstration purposes we, however, used a globally continuous ansatz space.

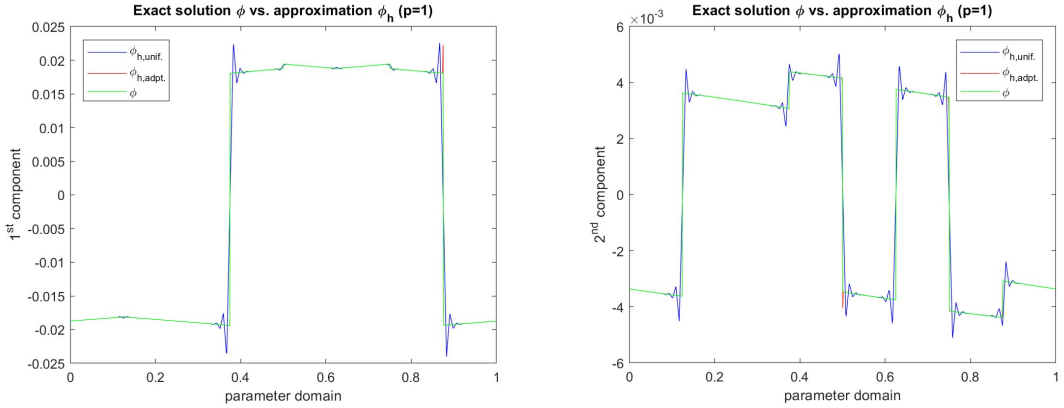
In conclusion we see that we have a well working implementation of V_h and F_h.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



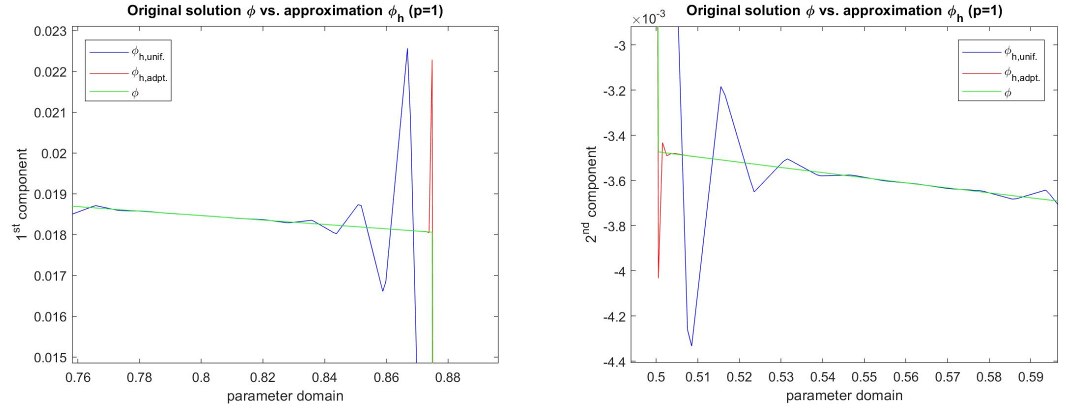
(a) 1^{st} component, initial mesh

(b) 2^{nd} component, initial mesh



(c) 1^{st} component, refined mesh

(d) 2^{nd} component, refined mesh



(e) 1^{st} component, refined mesh (magnified view)

(f) 2^{nd} component, refined mesh (magnified view)

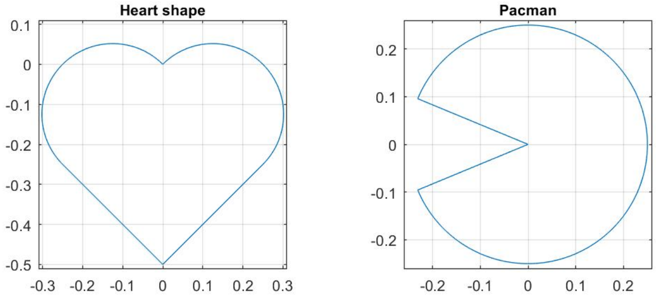
Figure 4.10.: Approximate solution Φ_h for $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ on initial mesh and uniformly ($N = 129$) and adaptively ($N = 102$) refined mesh for the combined validation of V and K from Section 4.2.4

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

5. Numerical examples

In this chapter, we present some numerical examples for the direct and indirect approach for solving the Dirichlet boundary value problem (2.21), where we also investigate the convergence rates of the $(h\text{–}h/2)$–estimator. We seek a solution $\boldsymbol{\phi} \in \boldsymbol{H}^{-1/2}(\Gamma)$ to Symm's integral equation (2.37) for the direct approach and (4.27) for the indirect approach. Using the Galerkin method, we then compute an approximate solution $\Phi_h \in X_h$ of $(\Phi_h, \Psi_h)_V = (\boldsymbol{F}, \Psi_h)_\Gamma$ for all $\Psi_h \in X_h$, where $\boldsymbol{F} = (K + 1/2)\boldsymbol{g}$ for the direct approach and $\boldsymbol{F} = \boldsymbol{g}$ for the indirect approach. We then again proceed as described in Section 4.1. for the Lamé parameters, we choose $\lambda = \mu = 0.4$.

For the geometry, we assume that the parametrization γ of Γ is given as in (4.2). Amongst others we will consider the heart–shape boundary and the pacman geometry in this chapter, which are visualized in Figure 5.1.



(a) heart–shape (b) pacman

Figure 5.1.: Geometries for numerical examples

For the heart–shape boundary (Figure 5.1a) the boundary of the geometry is parametrized on $[0,1]$ by the NURBS curve induced by

$$\begin{aligned} p_\gamma &= 2, \\ \check{\mathcal{K}}^\gamma &= \left(\frac{1}{6}, \frac{1}{6}, \frac{2}{6}, \frac{2}{6}, \frac{3}{6}, \frac{3}{6}, \frac{3}{6}, \frac{4}{6}, \frac{4}{6}, \frac{5}{6}, \frac{5}{6}, 1, 1, 1\right), \\ \mathcal{W}^\gamma &= \left(1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, 1, \frac{1}{\sqrt{2}}, 1, \frac{1}{\sqrt{2}}, 1, 1, 1, 1, 1\right), \\ (C_i)_{i=1-p}^{N_\gamma - \#b+1} &= \frac{1}{4\sqrt{2}} \begin{pmatrix} \cos\left(\frac{\pi}{4}\right) & \sin\left(\frac{\pi}{4}\right) \\ -\sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) \end{pmatrix} \cdot \left(\begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \right. \\ &\left. \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 \\ -1 \end{pmatrix} \right). \end{aligned} \tag{5.1}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Next we describe the boundary parametrization of the pacman. To this end, let $\alpha := (2\pi)7/8$. The boundary of the geometry is parametrized on $[0,1]$ by the NURBS curve induced by

$$\begin{aligned}
p_\gamma &= 2,\\
\check{\mathcal{K}}^\gamma &= \left(\frac{1}{6},\frac{1}{6},\frac{2}{6},\frac{2}{6},\frac{2}{6},\frac{3}{6},\frac{3}{6},\frac{3}{6},\frac{4}{6},\frac{4}{6},\frac{4}{6},\frac{5}{6},\frac{5}{6},1,1,1\right),\\
\mathcal{W}^\gamma &= \left(1,\cos\left(\frac{\alpha}{8}\right),1,1,1,1,1,1,1,1,\cos\left(\frac{\alpha}{8}\right),1,\cos\left(\frac{\alpha}{8}\right),\right.\\
&\qquad \left.1,1,\cos\left(\frac{\alpha}{8}\right)\right),\\
(C_i)_{i=1-p}^{N_\gamma-\#b+1} &= \frac{1}{4}\left(\begin{pmatrix}\cos\left(\frac{2\alpha}{8}\right)\\ \sin\left(\frac{2\alpha}{8}\right)\end{pmatrix},\begin{pmatrix}\cos\left(\frac{3\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\\ \sin\left(\frac{3\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\end{pmatrix},\begin{pmatrix}\cos\left(\frac{4\alpha}{8}\right)\\ \sin\left(\frac{4\alpha}{8}\right)\end{pmatrix},\begin{pmatrix}\cos\left(\frac{4\alpha}{8}\right)\\ \sin\left(\frac{4\alpha}{8}\right)\end{pmatrix},\right. \qquad (5.2)\\
&\quad \begin{pmatrix}\cos\left(\frac{4\alpha}{8}\right)/2\\ \sin\left(\frac{4\alpha}{8}\right)/2\end{pmatrix},\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}\cos\left(\frac{-4\alpha}{8}\right)/2\\ \sin\left(\frac{-4\alpha}{8}\right)/2\end{pmatrix},\begin{pmatrix}\cos\left(\frac{-4\alpha}{8}\right)\\ \sin\left(\frac{-4\alpha}{8}\right)\end{pmatrix},\\
&\quad \begin{pmatrix}\cos\left(\frac{-4\alpha}{8}\right)\\ \sin\left(\frac{-4\alpha}{8}\right)\end{pmatrix},\begin{pmatrix}\cos\left(\frac{-3\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\\ \sin\left(\frac{-3\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\end{pmatrix},\begin{pmatrix}\cos\left(\frac{-2\alpha}{8}\right)\\ \sin\left(\frac{-2\alpha}{8}\right)\end{pmatrix},\\
&\quad \left.\begin{pmatrix}\cos\left(\frac{-1\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\\ \sin\left(\frac{-1\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}\cos\left(\frac{1\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\\ \sin\left(\frac{1\alpha}{8}\right)/\cos\left(\frac{\alpha}{8}\right)\end{pmatrix}\right).
\end{aligned}$$

For both of the above described geometries, we chose the knot multiplicity at the corners of the geometry to $p_\gamma + 1$ so that we can approximate discontinuities due to jumps in the normal vector.

As approximation space, we again consider the NURBS space $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$, where we use the knots $\check{\mathcal{K}}^\gamma$ and weights $\mathcal{W}^\gamma$ from the corresponding geometry parametrization for the initial mesh $\mathcal{T}_0$, and then perform uniform and adaptive refinement according to Algorithm 3.5 with adaptivity constant $\theta = 0.9$. We use the local contributions $\widetilde{\mu}(z)$ for all $z \in \mathcal{N}_h$ of the $(h$–$h/2)$–estimator as refinement indicators. Furthermore, we also consider $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ as an ansatz space and again perform uniform and adaptive refinement as above. We expect a convergence rate of $\mathcal{O}(h^{3/2+p}) = \mathcal{O}(n^{-3/2-p})$ for the $(h$–$h/2)$–estimator in the case of adaptive refinement.

5.1. Indirect BEM with constant Dirichlet boundary data

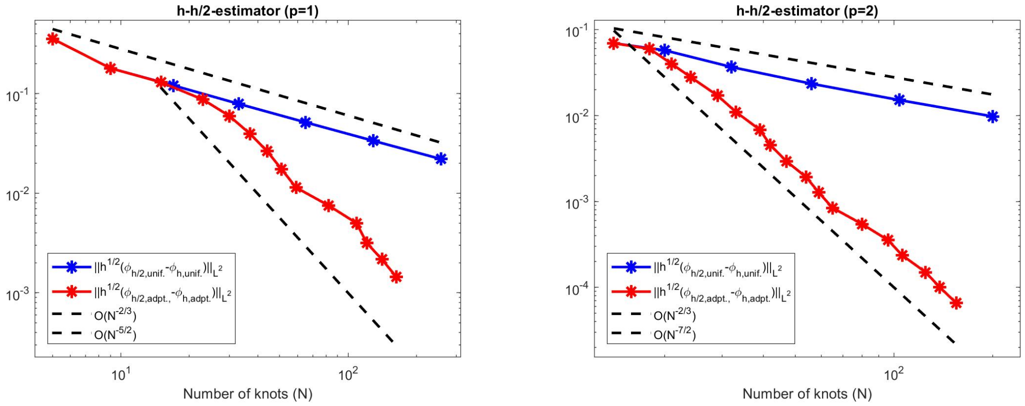
We first consider the indirect approach for the Dirichlet boundary value problem with $\boldsymbol{g} = (1,-1)^T$. The corresponding integral equation reads

$$V\boldsymbol{\phi} = \begin{pmatrix}1\\-1\end{pmatrix} \quad \text{on } \Gamma,$$

where we seek the unknown solution $\boldsymbol{\phi} \in \boldsymbol{H}^{-1/2}(\Gamma)$, which is presumably not smooth. We consider the following geometries, i.e., the square (see Figure 4.7a and NURBS curve data (4.28)), the heart–shape (see Figure 5.1a and NURBS curve data (5.1)) and the pacman (see Figure 5.1b and NURBS curve data (5.2)). As ansatz spaces, we consider $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$, where

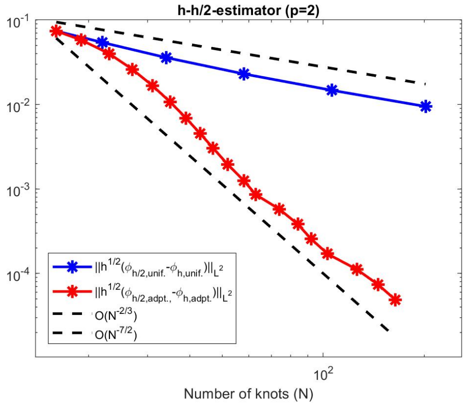
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub



(a) square

(b) heart–shape



(c) pacman

Figure 5.2.: $(h\text{–}h/2)$–estimator for $\Phi_h \in X_h = \boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ with uniformly and adaptively refined mesh for equation $V\boldsymbol{\phi} = (1, -1)^T$ from Section 5.1

we use the knots $\check{\mathcal{K}}^\gamma$ and weights $\mathcal{W}^\gamma$ from the corresponding geometry parametrization for the initial mesh $\mathcal{T}_0$. Furthermore, we also consider $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0, 1, 2, 3\}$.

In Figure 5.2, we see the $(h\text{–}h/2)$–estimator plotted over the number of knots N for uniform and adaptive refinement according to Algorithm 3.5 for $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$. We expect a convergence rate of $\mathcal{O}(N^{-2/3})$ for uniform refinement for all three examples considered, which can be deduced from considering an exterior value problem and then measuring the non convex outer angles.

On the heart–shape and the pacman geometry, the $(h\text{–}h/2)$–estimator shows the desired convergence rate for the case of uniform refinement. For the square, the rate is almost what we expect for uniform refinement. For adaptive refinement, all three geometries show the desired rate, whereas for the square we observe a longer pre-asymptotic phase.

In Figure 5.3, we compare the convergence rates of the $(h\text{–}h/2)$–estimator for different $p \in \{0, 1, 2, 3\}$ with ansatz space $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for uniform and adaptive refinement. For uniform refinement, the estimator behaves similar to the $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ case for all geometries and all p

considered. For adaptive refinement, we see that the estimator eventually reaches the desired convergence rate after some pre–asymptotic phase.

As a last example, we consider the spline space as ansatz space. Splines are piecewise polynomial functions which are $(p-1)$-times differentiable at each node. The spline space can be derived from the NURBS space $\hat{\mathscr{N}}^p(\check{\mathcal{K}}, \mathcal{W})$ when choosing multiplicity one at each node, except for the last node, where we have to choose multiplicity $p+1$. The weights $\mathcal{W}$ are then all chosen equal to 1. Similar as in Section 3.2.3, we then define the two-dimensional splines space on $\mathcal{T}_h$ and write $\mathbb{S}^p(\mathcal{T}_h)$. In Figure 5.4, we see the $(h$–$h/2)$–estimator for $X_h = \mathbb{S}^p(\mathcal{T}_h)$ for $p \in \{0, 1, 2\}$. As before, we observe the expected convergence rate of $\mathcal{O}(N^{-2/3})$ for uniform refinement and $\mathcal{O}(N^{-3/2-p})$ for adaptive refinement.

5.2. Direct BEM for analytic solution

We consider the holomorphic function $\boldsymbol{u}(z) = \bar{z}^{2/3}$, which can be interpreted as a function $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ and then is according to [ME14, Section 7.2.1 and Section 3.2] a solution to the homogeneous Lamé equation. For the Dirichlet boundary data, we evaluate $\boldsymbol{u}$ at the boundary to obtain

$$\boldsymbol{g}(x, y) := \boldsymbol{u}(x, y) = \begin{pmatrix} \mathrm{Re}(x - iy)^{2/3} \\ \mathrm{Im}(x - iy)^{2/3} \end{pmatrix}, \quad (x, y) \in \Gamma.$$

We make a direct approach and aim to solve Symm's integral equation (2.37).

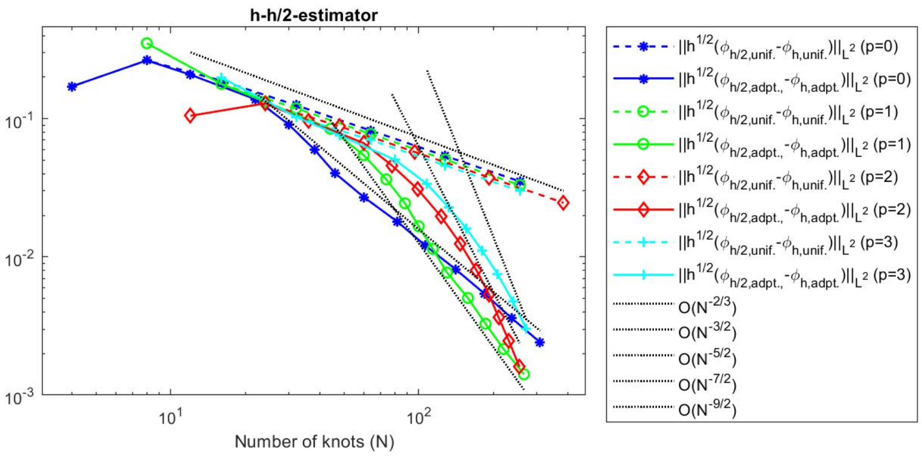
In [ME14, Section 7.2.1], this solution is investigated on the L-shape. Since we are able to deal with curved geometries, we chose to consider the pacman geometry (see Figure 5.1b) with the NURBS data for the boundary given in (5.2).

As ansatz spaces, we consider $\boldsymbol{\mathcal{S}}(\mathcal{T}_h)$, where we use the knots $\check{\mathcal{K}}^\gamma$ and weights $\mathcal{W}^\gamma$ from (5.2) for the initial mesh $\mathcal{T}_0$ and then perform uniform or adaptive refinement according to Algorithm 3.5. Furthermore, we also consider $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0, 1, 2\}$. For uniform refinement, we expect a convergence rate of $\mathcal{O}(N^{-2/3})$ for the $(h$–$h/2)$–estimator. For adaptive refinement, we expect $\mathcal{O}(N^{-3/2-p})$.

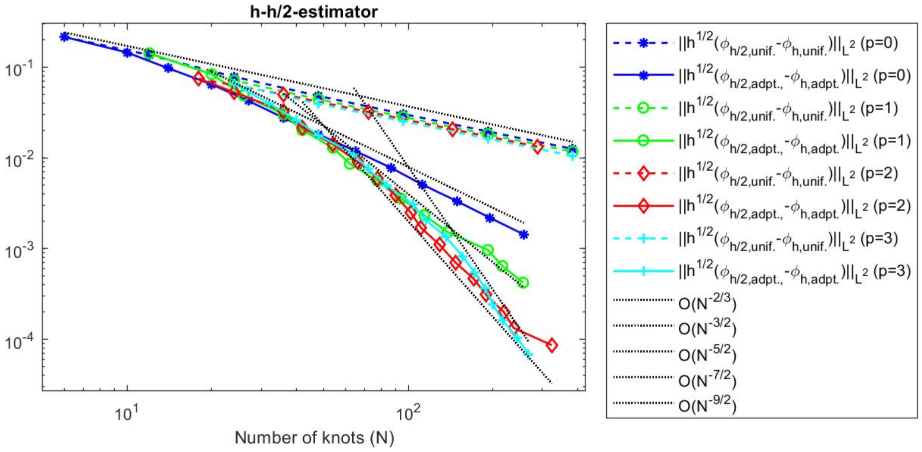
In Figure 5.5, we see that the estimator shows the expected rates. For adaptive refinement, the mesh is strongly refined close to to the re–entrant corner of the pacman geometry, where the conormal derivative $\boldsymbol{\phi}$ has a singularity.

In Figure 5.6, we see that the convergence rates of the $(h$–$h/2)$–estimator also hold for the ansatz spaces $\boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0, 1, 2\}$.

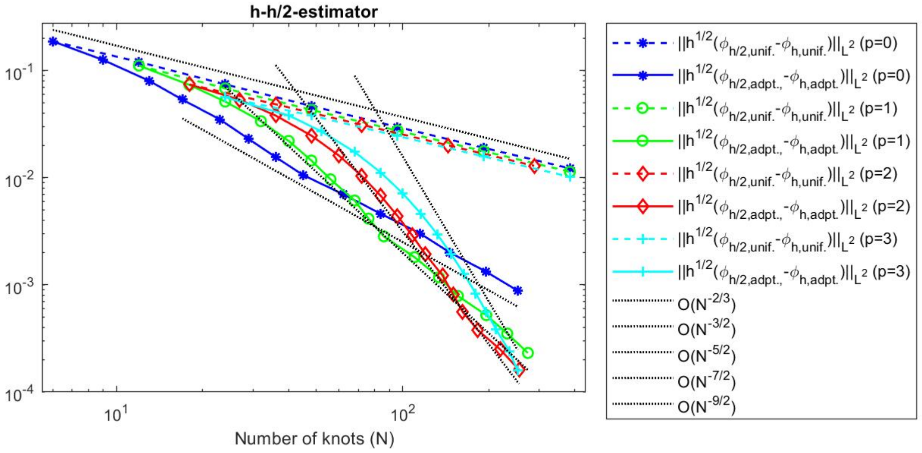
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) square



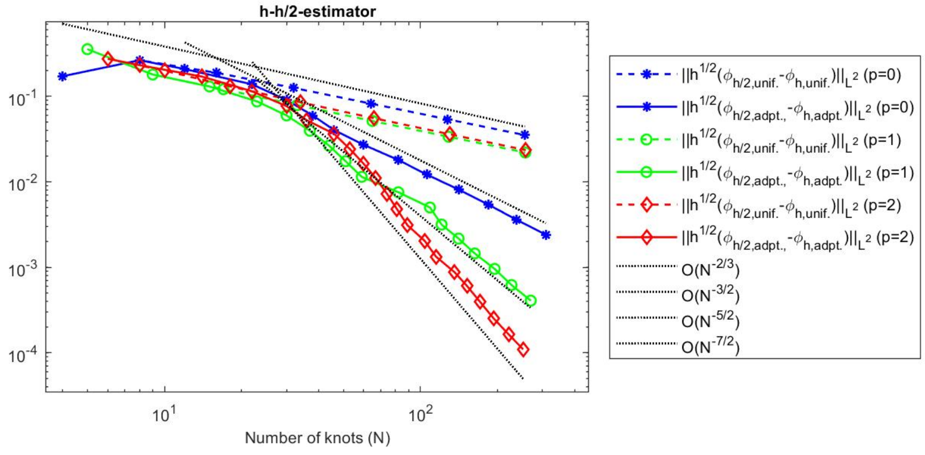
(b) heart–shape



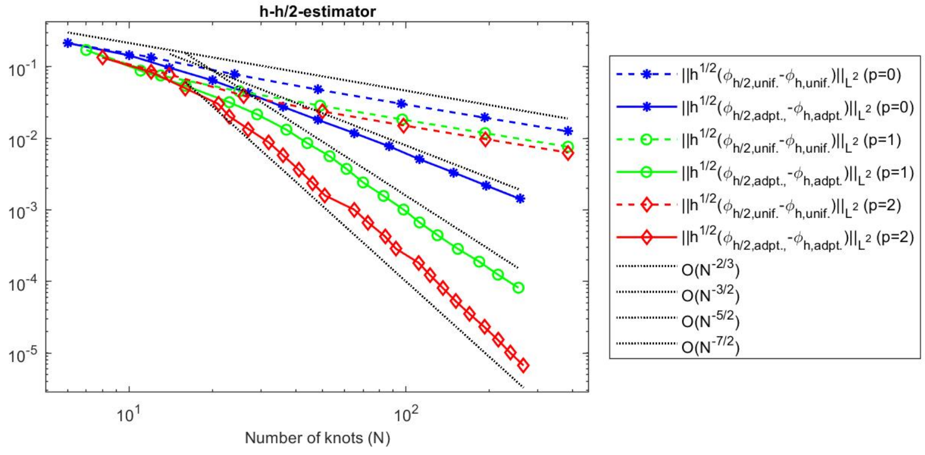
(c) pacman

Figure 5.3.: $(h\text{–}h/2)$–estimator for $\Phi_h \in X_h = \boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0, 1, 2, 3\}$ for uniformly and adaptively refined mesh for equation $V\boldsymbol{\phi} = (1, -1)^T$from Section 5.1

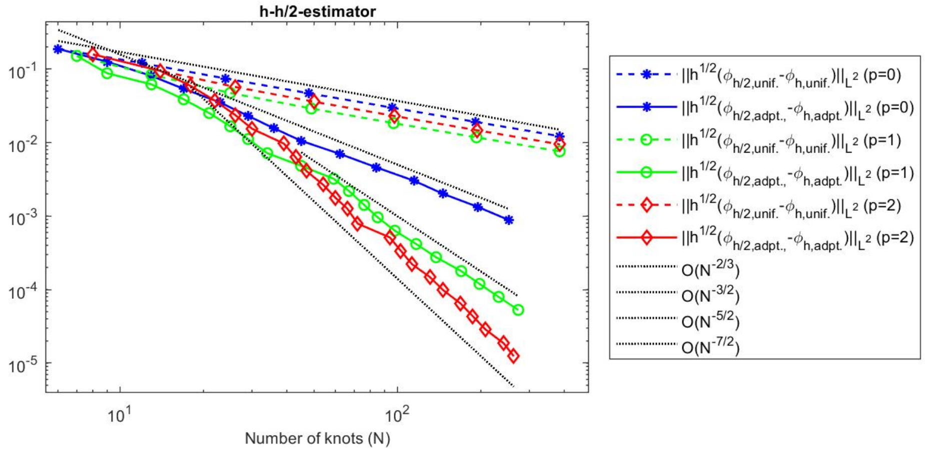
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.



(a) square



(b) heart–shape



(c) pacman

Figure 5.4.: $(h\text{–}h/2)$–estimator for $\Phi_h \in X_h = \mathbb{S}^p(\mathcal{T}_h)$ for $p \in \{0, 1, 2\}$ for uniformly and adaptively refined mesh for equation $V\boldsymbol{\phi} = (1, -1)^T$from Section 5.1

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

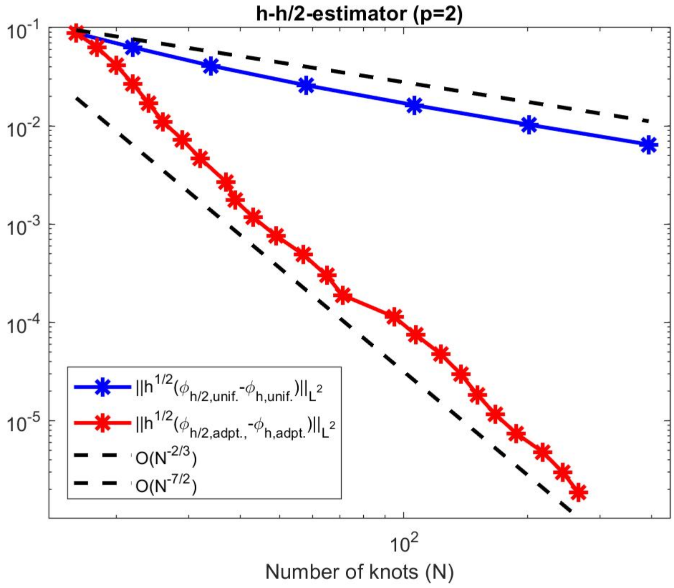


Figure 5.5.: $h - h/2-$estimator for $\Phi_h \in X_h = \boldsymbol{\mathcal{S}}(\mathcal{T}_h)$ for pacman for holomorphic $\boldsymbol{g}$ from Section 5.2

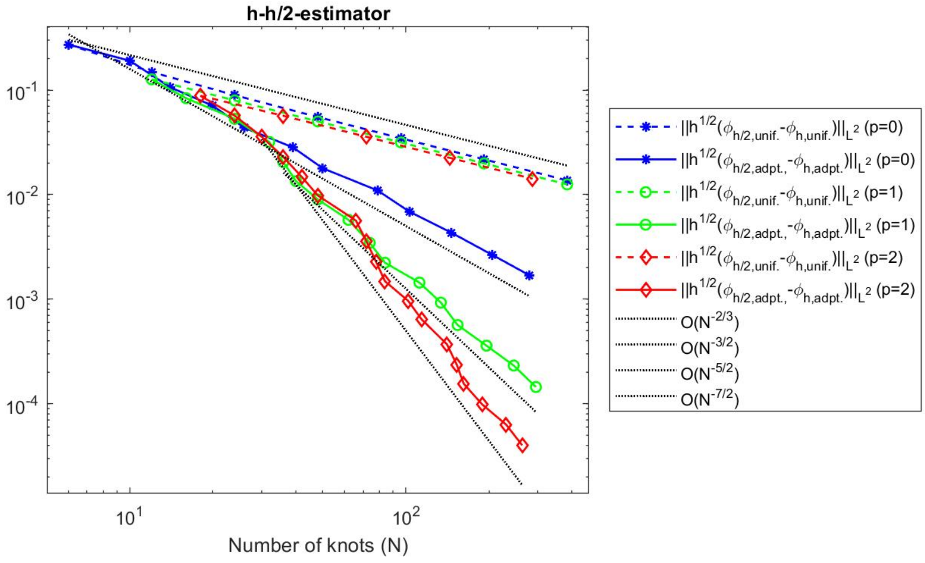


Figure 5.6.: $h - h/2-$estimator for $\Phi_h \in X_h = \boldsymbol{\mathcal{P}}^p(\mathcal{T}_h)$ for $p \in \{0, 1, 2\}$ for pacman for holomorphic $\boldsymbol{g}$ from Section 5.2

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

A. Implementation

Our implementation relies on the implementation for the Laplace equation from Gregor Gantner (cf. [Gan14]). For the Lamé equation, we did not only adapt the integral kernel for the V and K operator but also made some modifications in order implement the more dimensional setting of the Lamé equation. The fundamental solution of the Laplace equation and its normal derivative were replaced by the fundamental solution of the Lamé equation and its conormal derivative. For a detailed description of the complete implementation we refer to [Gan14].

A.1. `Vmatrix.h` and `Vmatrix.c`

The computation of the Matrix V_h is discussed in on Section 4.1.1.

Listing A.1: Vmatrix.h

```c
#ifndef _Vmatrix_h
#define _Vmatrix_h

#include <math.h>
#include <stdio.h>
#include "Spline.h"

/* parameters:
 Data_Gamma...NURBSData* for geometry Gamma, see Structures.h
 wcpoints1_gam...first component of weighted control points
 w_l^\gamma*C_l^\gamma for geometry corresponding to knots
 wcpoints2_gam...second component of weighted control points
 w_l^\gamma*C_l^\gamma for geometry corresponding to knots
 Data_Basis...NURBSData* for Basis of approximation space
 Data_Gauss...QuadData* for quadrature with weight function 1 om [0,1],
 see Structures.h
 Data_LogGauss...QuadData* for quadrature with weight function -log(x)
 on [0,1], see Structures.h

 comments:
 we assume that:
 -)path gamma induced by Data_Gamma and wcpoints_gam is either positively
 orientated regular closed curve,
 which parametizes boundary Gamma of Lipschitz domain Omega with diam(Omega)<1
 or regular open curve, in this case we assume #b=p_gam+1
 -)#t_i^gamma<=p_gam+1
 -)number of different entries in knots of Data_Gamma >= 4
 -){t_i^gamma:i=1...N_gam}<={t_i:i=1...N}
 -)#t_i<=p+1

 */


double SquareIntegrand_V_smooth(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                double* wcpoints2_gam,NURBSData* Data_Basis,
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
                                double s,double t,int i,int k,
                                double denominator,double Jdet);
// returns log(|\gamma(s)-\gamma(t)|/denominator)
// *\tilde{R}_i(s)*\tilde{R}_k(t)*Jdet,
// where s!=t in [a,b), i,k in {1-p,...,N-#b+1}, denominator>0 and Jdet in \R


double SquareIntegrand_V_log(NURBSData* Data_Gamma,double* wcpoints1_gam,
                             double* wcpoints2_gam,NURBSData* Data_Basis,
                             double s,double t,int i,int k,double Jdet);
// returns \tilde{R}_i(s)*\tilde{R}_k(t)*Jdet,
// where s!=t in [a,b), i,k in {1-p,...,N-#b+1} and Jdet in \R


double SquareIntegrand_V_rational(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                  double* wcpoints2_gam,NURBSData* Data_Basis,
                                  double s,double t,int i,int k,int j,int m,
                                  double Jdet);
// returns (\gamma_j(s)-\gamma_j(t))*(\gamma_m(s)-\gamma_m(t))/|\gamma(s)-\gamma(t)|
// *\tilde{R}_i(s)*\tilde{R}_k(t)*Jdet,
// where s!=t in [a,b), i,k in {1-p,...,N-#b+1}, j,m in {0,1} and Jdet in \R


double SquareIntegral_V_Identical(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                  double* wcpoints2_gam,NURBSData* Data_Basis,
                                  QuadData* Data_Gauss,QuadData* Data_LogGauss,
                                  int i,int k,int l,int j,int m,int delta_jm,
                                  double lambda,double mu);
// returns \int_0^1 \int_0^1 \check{G}_{l,l}(s,t) \tilde{R}_{i,l}(s)
// \tilde{R}_{k,l}(t) dt ds,
// where i,k in {1-p,...,N-#b+1},
// l in {max(i,1),...,min(i+p,N)} \cap {max(k,1)...,min(k+p,N)} with H_l>0,
// j,m,delta_jm in {0,1} and lambda,mu in \R


double SquareIntegral_V_Adjacent(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                 double* wcpoints2_gam,NURBSData* Data_Basis,
                                 QuadData* Data_Gauss,QuadData* Data_LogGauss,
                                 int i,int k,int l1,int l2,int j,int m,
                                 int delta_jm,double lambda,double mu);
// returns \int_0^1 \int_0^1 \check{G}_{l1,l2}(s,t)
// \tilde{R}_{i,l1}(s) \tilde{R}_{k,l2}(t) dt ds for adjacent elements,
// i.e. \gamma([t_{l_1-1},t_{l_1} \cap \gamma([t_{l_2-1},t_{l_2}]) consists
// of one point, singularity at s=0 and t=1,
// i,k in {1-p,...,N-#b+1}, l1 in {max(i,1),...,min(i+p,N)},
// l2 in {max(k,1),...,min(k+p,N)} with min(H_l1,H_l2)>0, j,m,delta_jm in {0,1}
// and lmbda,mu in \R


void build_Vmatrix(double* output,NURBSData* Data_Gamma,double* wcpoints1_gam,
                   double* wcpoints2_gam,NURBSData* Data_Basis,
                   QuadData* Data_Gauss,QuadData* Data_LogGauss,double lambda,
                   double mu);
// turns output[i+p-1+(k+p-1)*2*(N-multb+1+p)+j*(N-multb+1+p)+m*2*(N-multb+1+p)]
// into <V \hat{R}_k^m,\hat{R}_i^j>_{L^2(Gamma)} for i,k=1-p...N-#b+1, j,m=0,1,
// \hat{R}_i^j=R_{i,p}*e_j circ gamma^(-1) are the transformed basis functions
// with e_j being the j-th unit vector

#endif
```

Listing A.2: Vmatrix.c

```
#include "Vmatrix.h"
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```c


double SquareIntegrand_V_smooth(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                double* wcpoints2_gam,NURBSData* Data_Basis,
                                double s,double t,int i,int k,
                                double denominator,double Jdet){

    double tmp1[2];
    double tmp2[2];
    double diff_gam[2];
    double R_til_i,R_til_k; // \tilde{R}_i(s), \tilde{R}_k(t)

    eval_NURBSCurve(tmp1,Data_Gamma,wcpoints1_gam,wcpoints2_gam,s);
    // gamma(s)
    eval_NURBSCurve(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma(t)
    diff_gam[0] = tmp1[0] - tmp2[0];
    diff_gam[1] = tmp1[1] - tmp2[1];
    eval_NURBSCurveDeriv(tmp1,Data_Gamma,wcpoints1_gam,wcpoints2_gam,s);
    // gamma'(s)
    eval_NURBSCurveDeriv(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma'(t)
    R_til_i = eval_NURBS(Data_Basis, i, s) * norm(tmp1);
    R_til_k = eval_NURBS(Data_Basis, k, t) * norm(tmp2);
    return log(norm(diff_gam)/denominator)*R_til_i*R_til_k*Jdet;
}


double SquareIntegrand_V_log(NURBSData* Data_Gamma,double* wcpoints1_gam,
                             double* wcpoints2_gam,NURBSData* Data_Basis,
                             double s,double t,int i,int k,double Jdet){


    double tmp1[2];
    double tmp2[2];
    double R_til_i,R_til_k; // \tilde{R}_i(s), \tilde{R}_k(t)

    eval_NURBSCurveDeriv(tmp1,Data_Gamma,wcpoints1_gam,wcpoints2_gam,s);
    // gamma'(s)
    eval_NURBSCurveDeriv(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma'(t)
    R_til_i = eval_NURBS(Data_Basis, i, s) * norm(tmp1);
    R_til_k = eval_NURBS(Data_Basis, k, t) * norm(tmp2);
    return -R_til_i*R_til_k*Jdet;
}


double SquareIntegrand_V_rational(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                  double* wcpoints2_gam,NURBSData* Data_Basis,
                                  double s,double t,int i,int k,int j,int m,
                                  double Jdet){

    double tmp1[2];
    double tmp2[2];
    double diff_gam[2];
    double R_til_i,R_til_k; // \tilde{R}_i(s), \tilde{R}_k(t)

    eval_NURBSCurve(tmp1,Data_Gamma,wcpoints1_gam,wcpoints2_gam,s);
    // gamma(s)
    eval_NURBSCurve(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma(t)
    diff_gam[0] = tmp1[0] - tmp2[0];
    diff_gam[1] = tmp1[1] - tmp2[1];
    eval_NURBSCurveDeriv(tmp1,Data_Gamma,wcpoints1_gam,wcpoints2_gam,s);
    // gamma'(s)
    eval_NURBSCurveDeriv(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma'(t)
    R_til_i = eval_NURBS(Data_Basis, i, s) * norm(tmp1);
    R_til_k = eval_NURBS(Data_Basis, k, t) * norm(tmp2);

    return R_til_i*R_til_k*Jdet*diff_gam[j]*diff_gam[m]/(norm(diff_gam))/
            (norm(diff_gam));
}


double SquareIntegral_V_Identical(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                  double* wcpoints2_gam,NURBSData* Data_Basis,
                                  QuadData* Data_Gauss,QuadData* Data_LogGauss,
                                  int i,int k,int l,int j,int m,int delta_jm,
                                  double lambda,double mu){

    double* nodes_gauss=get_QuadData_nodes(Data_Gauss);
    double* weights_gauss=get_QuadData_weights(Data_Gauss);
    int n_gauss=get_QuadData_n(Data_Gauss);
    double* nodes_loggauss=get_QuadData_nodes(Data_LogGauss);
    double* weights_loggauss=get_QuadData_weights(Data_LogGauss);
    int n_loggauss=get_QuadData_n(Data_LogGauss);
    int q1,q2;
    double t_lm1=knotseq(Data_Basis,l-1); // t_{l-1}
    double t_l=knotseq(Data_Basis,l); // t_{l}
    double H_l=t_l-t_lm1; // H_l
    double squareint=0; // integral over square
    double factor_log = (-3)*mu-lambda; // factor in front of log term
    double factor_rat = mu+lambda; // factor in front of rational term
    double factor = 1/(4*M_PI*mu*(2*mu+lambda)); // generall factor
    double intpoint1, intpoint2; // first and second integration point
    double denominator;
    double Jdet; // Jacobi determinant for Duffy transformation

    // smooth integrals
    for (q1=0;q1<n_gauss;q1=q1+1) {
        for (q2=0;q2<n_gauss;q2=q2+1) {
            // first double integral
            intpoint1= t_lm1+H_l*nodes_gauss[q1];
            intpoint2 = t_lm1+H_l*(nodes_gauss[q1]*(1-nodes_gauss[q2]));
            denominator=nodes_gauss[q1]*nodes_gauss[q2];
            Jdet=nodes_gauss[q1];
            if (delta_jm) {
                squareint += weights_gauss[q1]*weights_gauss[q2]*factor_log*
                SquareIntegrand_V_smooth(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                         Data_Basis,intpoint1,intpoint2,i,k,
                                         denominator,Jdet);
            }
            squareint += weights_gauss[q1]*weights_gauss[q2]*factor_rat*
            SquareIntegrand_V_rational(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,j,m,Jdet);
            // second double integral
            intpoint1=t_lm1+H_l*(1-nodes_gauss[q1]);
            intpoint2 = t_lm1 + H_l * (1+nodes_gauss[q1]*(nodes_gauss[q2]-1));
            denominator=nodes_gauss[q1]*nodes_gauss[q2];
            Jdet=nodes_gauss[q1];
            if (delta_jm) {
                squareint += weights_gauss[q1]*weights_gauss[q2]*factor_log*
                SquareIntegrand_V_smooth(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                         Data_Basis,intpoint1,intpoint2,i,k,
                                         denominator,Jdet);
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
            }
            squareint += weights_gauss[q1]*weights_gauss[q2]*factor_rat*
            SquareIntegrand_V_rational(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,j,m,Jdet);
        }
    }
    // integrals with s-logarithmic singularity
    if (delta_jm) {
        for (q1=0;q1<n_loggauss;q1=q1+1) {
            for (q2=0;q2<n_gauss;q2=q2+1) {
                // first double integral
                intpoint1=t_lm1+H_l*nodes_loggauss[q1];
                intpoint2 = t_lm1 + H_l*(nodes_loggauss[q1]*(1-nodes_gauss[q2]));
                Jdet=nodes_loggauss[q1];
                squareint += weights_loggauss[q1]*weights_gauss[q2]*factor_log
                *SquareIntegrand_V_log(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,Jdet);
                // second double integral
                intpoint1=t_lm1+H_l*(1-nodes_loggauss[q1]);
                intpoint2 = t_lm1+H_l*(1+nodes_loggauss[q1]*(nodes_gauss[q2]-1));
                Jdet=nodes_loggauss[q1];
                squareint += weights_loggauss[q1]*weights_gauss[q2]*factor_log
                *SquareIntegrand_V_log(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,Jdet);
            }
        }
    }
    // integrals with t-logarithmic singularity
    if (delta_jm) {
        for (q1=0;q1<n_gauss;q1=q1+1) {
            for (q2=0;q2<n_loggauss;q2=q2+1) {
                // first double integral
                intpoint1=t_lm1+H_l*nodes_gauss[q1];
                intpoint2 = t_lm1+H_l*(nodes_gauss[q1]*(1-nodes_loggauss[q2]));
                Jdet=nodes_gauss[q1];
                squareint += weights_gauss[q1]*weights_loggauss[q2]*factor_log
                *SquareIntegrand_V_log(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,Jdet);
                // second double integral
                intpoint1=t_lm1+H_l*(1-nodes_gauss[q1]);
                intpoint2 = t_lm1+H_l*(1+nodes_gauss[q1]*(nodes_loggauss[q2]-1));
                Jdet=nodes_gauss[q1];
                squareint += weights_gauss[q1]*weights_loggauss[q2]*factor_log
                *SquareIntegrand_V_log(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,Jdet);
            }
        }
    }
    return squareint*factor;
}


double SquareIntegral_V_Adjacent(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                 double* wcpoints2_gam,NURBSData* Data_Basis,
                                 QuadData* Data_Gauss,QuadData* Data_LogGauss,
                                 int i,int k,int l1,int l2,int j,int m,
                                 int delta_jm,double lambda,double mu){


    double* nodes_gauss=get_QuadData_nodes(Data_Gauss);
    double* weights_gauss=get_QuadData_weights(Data_Gauss);
    int n_gauss=get_QuadData_n(Data_Gauss);
    double* nodes_loggauss=get_QuadData_nodes(Data_LogGauss);
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
    double* weights_loggauss=get_QuadData_weights(Data_LogGauss);
    int n_loggauss=get_QuadData_n(Data_LogGauss);
    int q1,q2;
    double t_l1m1=knotseq(Data_Basis,l1-1); // t_{l_1-1}
    double t_l1=knotseq(Data_Basis,l1); // t_{l_1}
    double H_l1=t_l1-t_l1m1; // H_{l_1}
    double t_l2m1=knotseq(Data_Basis,l2-1); // t_{l_2-1}
    double t_l2=knotseq(Data_Basis,l2); // t_{l_2}
    double H_l2=t_l2-t_l2m1; // H_{l_1}
    double squareint=0; // integral over square
    double factor_log = (-3)*mu-lambda; // factor in front of log term
    double factor_rat = mu+lambda; // factor in front of rational term
    double factor = 1/(4*M_PI*mu*(2*mu+lambda)); // generall factor
    double intpoint1, intpoint2; // first and second integration point
    double denominator;
    double Jdet; // Jacobi determinant of Duffy transformation

    // smooth integrals
    for (q1=0;q1<n_gauss;q1=q1+1) {
        for (q2=0;q2<n_gauss;q2=q2+1) {
            // first double integral
            intpoint1=t_l1m1+H_l1*nodes_gauss[q1];
            intpoint2 = t_l2m1 + H_l2 *  (1-nodes_gauss[q1]*nodes_gauss[q2]);
            denominator=nodes_gauss[q1];
            Jdet=nodes_gauss[q1];
            if (delta_jm) {
                squareint += weights_gauss[q1]*weights_gauss[q2]*factor_log*
                SquareIntegrand_V_smooth(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                         Data_Basis,intpoint1,intpoint2,i,k,
                                         denominator,Jdet);
            }
            squareint += weights_gauss[q1]*weights_gauss[q2]*factor_rat*
            SquareIntegrand_V_rational(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,j,m,Jdet);
            // second double integral
            intpoint1=t_l1m1+H_l1*nodes_gauss[q1]*nodes_gauss[q2];
            intpoint2 = t_l2m1 + H_l2 * (1-nodes_gauss[q2]);
            denominator=nodes_gauss[q2];
            Jdet=nodes_gauss[q2];
            if (delta_jm) {
                squareint += weights_gauss[q1]*weights_gauss[q2]*factor_log*
                SquareIntegrand_V_smooth(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                         Data_Basis,intpoint1,intpoint2,i,k,
                                         denominator,Jdet);
            }
            squareint += weights_gauss[q1]*weights_gauss[q2]*factor_rat*
            SquareIntegrand_V_rational(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,j,m,Jdet);
        }
    }
    // integral with s-logarithmic singularity
    if (delta_jm) {
        for (q1=0;q1<n_loggauss;q1=q1+1) {
            for (q2=0;q2<n_gauss;q2=q2+1) {
                intpoint1=t_l1m1+H_l1*nodes_loggauss[q1];
                intpoint2 = t_l2m1+H_l2* (1-nodes_loggauss[q1]*nodes_gauss[q2]);
                Jdet=nodes_loggauss[q1];
                squareint += weights_loggauss[q1]*weights_gauss[q2]*factor_log
                *SquareIntegrand_V_log(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,Jdet);
            }
        }
    }
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
    // integral with t-logarithmic singularity
    if (delta_jm) {
        for (q1=0;q1<n_gauss;q1=q1+1) {
            for (q2=0;q2<n_loggauss;q2=q2+1) {
                intpoint1=t_l1m1+H_l1*nodes_gauss[q1]*nodes_loggauss[q2];
                intpoint2 = t_l2m1 + H_l2 * (1-nodes_loggauss[q2]);
                Jdet=nodes_loggauss[q2];
                squareint += weights_gauss[q1]*weights_loggauss[q2]*factor_log
                *SquareIntegrand_V_log(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                       Data_Basis,intpoint1,intpoint2,i,k,Jdet);
            }
        }
    }
    return squareint*factor;
}


void build_Vmatrix(double* output,NURBSData* Data_Gamma,double* wcpoints1_gam,
                   double* wcpoints2_gam,NURBSData* Data_Basis,
                   QuadData* Data_Gauss,QuadData* Data_LogGauss,double lambda,
                   double mu){

    double* nodes_gauss=get_QuadData_nodes(Data_Gauss);
    double* weights_gauss=get_QuadData_weights(Data_Gauss);
    int n_gauss=get_QuadData_n(Data_Gauss);
    double* nodes_loggauss=get_QuadData_nodes(Data_LogGauss);
    double* weights_loggauss=get_QuadData_weights(Data_LogGauss);
    int n_loggauss=get_QuadData_n(Data_LogGauss);
    double* knots=get_NURBSData_knots(Data_Basis);
    int N=get_NURBSData_N(Data_Basis);
    int p=get_NURBSData_p(Data_Basis);
    double tmp[2];
        int i,k,l1,l2,q1,q2,j,m;
    double b=knots[N-1];
        int multb=0; // #b
    while (nearly_equal(knotseq(Data_Basis,N-multb),b)) {multb=multb+1;}
        double R_til[N-multb+1+p][p+1][n_gauss];
    // R_til[i-1+p][l1-i][q1]=\tilde{R}_{i,l1}(nodes_gauss[q1])
        double gamma1[N][n_gauss];
    // gamma1[l1-1][q1] is first component of
    // gamma(t_{l1-1}+H_l1*nodes_gauss[q1])
        double gamma2[N][n_gauss];
    // gamma2[l1-1][q1] is second component of
    // gamma(t_{l1-1}+H_l1*nodes_gauss[q1])
    double squareint; // integral over square
    double t_l1m1,t_l1,H_l1,t_l2m1,t_l2,H_l2;
    // t_{l1-1},t_l1,H_l1,t_{l2-1},t_l2,H_l2
    double intpoint; // integration point
    int delta_jm;

    // calculation of R_til
    // R_i
        for (i=1-p;i<=(N-multb+1);i=i+1) {
        // elements with nonemty intersection with support of R_i
        for (l1=max(i,1);l1<=min(i+p,N);l1=l1+1) {
            // quadrature points
                for (q1=0;q1<n_gauss;q1=q1+1) {
                    t_l1m1=knotseq(Data_Basis,l1-1);
                t_l1=knotseq(Data_Basis,l1);
                H_l1=t_l1-t_l1m1;
                intpoint=t_l1m1+H_l1*nodes_gauss[q1];
                eval_NURBSCurveDeriv(tmp,Data_Gamma,wcpoints1_gam,
                                     wcpoints2_gam,intpoint);
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
                        R_til[i-1+p][l1-i][q1]=eval_NURBS(Data_Basis, i,intpoint)
                *norm(tmp);
                }
            }
        }

    // calculation of gamma1, gamma2
    for (l1=1;l1<=N;l1=l1+1){
        for (q1=0;q1<n_gauss;q1=q1+1){
            t_l1m1=knotseq(Data_Basis,l1-1);
            t_l1=knotseq(Data_Basis,l1);
            H_l1=t_l1-t_l1m1;
            intpoint=t_l1m1+H_l1*nodes_gauss[q1];
            eval_NURBSCurve(tmp,Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                            intpoint);
            gamma1[l1-1][q1]=tmp[0];
            gamma2[l1-1][q1]=tmp[1];
        }
    }

    // calculation of Vmatrix
    for (j=0;j<=1;j=j+1) {
        for (m=0;m<=j;m=m+1) {
            delta_jm = (j==m);
            // R_i
            for (i=1-p;i<=(N-multb+1);i=i+1) {
                // R_k
                for (k=1-p;k<=i;k=k+1) {
                    output[i+p-1+(k+p-1)*2*(N-multb+1+p)+j*(N-multb+1+p)+
                           m*2*(N-multb+1+p)*(N-multb+1+p)]=0;
                    // elements with nonemty intersection with support of R_i
                    for (l1=max(i,1);l1<=min(i+p,N);l1=l1+1) {
                        // elements with nonemty intersection with support of R_k
                        for (l2=max(k,1);l2<=min(k+p,N);l2=l2+1) {
                            t_l1m1=knotseq(Data_Basis,l1-1);
                            t_l1=knotseq(Data_Basis,l1);
                            t_l2m1=knotseq(Data_Basis,l2-1);
                            t_l2=knotseq(Data_Basis,l2);
                            H_l1=t_l1-t_l1m1;
                            H_l2=t_l2-t_l2m1;
                            // quadrature
                            if (0<min(H_l1,H_l2)){
                                // elements with no intersection
                                squareint=0;
                                if ((!nearly_equal(t_l1m1,t_l2m1))
                                    && (!nearly_equal(t_l1m1,t_l2))
                                    && (!nearly_equal(t_l1,t_l2m1))
                                    && (!nearly_equal(t_l1,t_l2))
                                    && ((l1!=(N-multb+1)) || (l2!=1))
                                    && ((l2!=(N-multb+1)) || (l1!=1))) {
                                    for (q1=0;q1<n_gauss;q1=q1+1) {
                                        for (q2=0;q2<n_gauss;q2=q2+1) {
                                            tmp[0]=gamma1[l1-1][q1]
                                            - gamma1[l2-1][q2];
                                            tmp[1]=gamma2[l1-1][q1]
                                            - gamma2[l2-1][q2];
                                            squareint+=
                                            weights_gauss[q1]*weights_gauss[q2]
                                            /(4*M_PI*mu*(2*mu+lambda))
                                            *(delta_jm*((-3)*mu-lambda)
                                            *log(norm(tmp))+(mu+lambda)*
                                            tmp[j]*tmp[m]/(norm(tmp))/
                                            (norm(tmp)))*R_til[i-1+p][l1-i][q1]*
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
                                            R_til[k-1+p][l2-k][q2];
                                        }
                                    }
                                }
                                // elements with intersection
                                else {
                                    // identical elements
                                    if (l1==l2){
                                        squareint+=SquareIntegral_V_Identical(
                                        Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                        Data_Basis,Data_Gauss,Data_LogGauss,
                                        i,k,l1,j,m,delta_jm,lambda,mu);
                                    }
                                    // adjacent elements
                                    else{
                                        // singularity at s=0,t=1
                                        if (nearly_equal(t_l1m1,t_l2)
                                            || ((l2==(N-multb+1)) && (l1==1))){
                                            squareint+=SquareIntegral_V_Adjacent(
                                            Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                            Data_Basis,Data_Gauss,Data_LogGauss,
                                            i,k,l1,l2,j,m,delta_jm,lambda,mu);
                                        }
                                        // singularity at s=1,t=0
                                        else{
                                            squareint+=SquareIntegral_V_Adjacent(
                                            Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                            Data_Basis,Data_Gauss,Data_LogGauss,
                                            k,i,l2,l1,j,m,delta_jm,lambda,mu);
                                        }
                                    }
                                }
                                output[i+p-1+(k+p-1)*2*(N-multb+1+p)+j*(N-multb+1+p)+
                                       m*2*(N-multb+1+p)*(N-multb+1+p)]+=
                                       H_l1*H_l2*squareint;
                            }
                        }
                    }
                    if (i!=k){
                        // V symmetric
                        output[k+p-1+(i+p-1)*2*(N-multb+1+p)+j*(N-multb+1+p)+
                                m*2*(N-multb+1+p)*(N-multb+1+p)] =
                        output[i+p-1+(k+p-1)*2*(N-multb+1+p)+j*(N-multb+1+p)+
                                m*2*(N-multb+1+p)*(N-multb+1+p)];
                    }
                    if (j!=m) {
                        // V symmetric in blocks
                        output[i+p-1+(k+p-1)*2*(N-multb+1+p)+m*(N-multb+1+p)+
                                j*2*(N-multb+1+p)*(N-multb+1+p)] =
                        output[i+p-1+(k+p-1)*2*(N-multb+1+p)+j*(N-multb+1+p)+
                                m*2*(N-multb+1+p)*(N-multb+1+p)];
                        if (i!=k){
                            // V symmetric
                            output[k+p-1+(i+p-1)*2*(N-multb+1+p)+m*(N-multb+1+p)+
                                    j*2*(N-multb+1+p)*(N-multb+1+p)] =
                            output[i+p-1+(k+p-1)*2*(N-multb+1+p)+m*(N-multb+1+p)+
                                    j*2*(N-multb+1+p)*(N-multb+1+p)];
                        }
                    }
                }
            }
        }
    }
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
}
```

A.2. `Fvector.h` and `Fvector.c`

The computation of the right-hand side vector F_h is discussed in on Section 4.1.2.

Listing A.3: Fvector.h

```
#ifndef _Fvector_h
#define _Fvector_h

#include <math.h>
#include <stdio.h>
#include "Spline.h"

/* parameters:
 Data_Gamma...NURBSData* for geometry Gamma, see Structures.h
 wcpoints1_gam...first component of weighted control points
 w_l^\gamma*C_l^\gamma for geometry corresponding to knots
 wcpoints2_gam...second component of weighted control points
 w_l^\gamma*C_l^\gamma for geometry corresponding to knots
 Data_Basis...NURBSData* for Basis of approximation space
 Data_Gauss...QuadData* for quadrature with weight function 1 on [0,1],
 see Structures.h
 Data_Gauss_small...QuadData* (see Structures.h) for quadrature with weight
 function 1 on [0,1] (with smaller number of nodes as Data_Gauss),
 used for quadrature for identical elements or adjacent elements
 in function build_Fvector
 with_K...0 to set Kg=0, 1 else

 comments:
 we assume that:
 -)path gamma induced by Data_Gamma and wcpoints_gam is either positively
 orientated regular closed curve,
 which parametizes boundary Gamma of Lipschitz domain Omega with diam(Omega)<1
 or regular open curve, in this case we assume #b=p_gam+1
 -)#t_i^gamma<=p_gam+1
 -)number of different entries in knots of Data_Gamma >= 4
 -){t_i^gamma:i=1...N_gam}<={t_i:i=1...N}
 -)#t_i<=p+1

 */

double PartDeriv_FundamentalSol(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                double* wcpoints2_gam,int i,int k,int m,
                                double s,double t,double Jdet,double lambda,double mu);
// returns \partial_{i,t}\check{U}_{km}(s,t)


double Sigma(NURBSData* Data_Gamma,double* wcpoints1_gam,
             double* wcpoints2_gam,int j,int p,int q,
             double s,double t,double Jdet,double lambda,double mu);
// returns sigma_{pq}(\check{U}_j)(s,t)


double SquareIntegrand_K(NURBSData* Data_Gamma,double* wcpoints1_gam,
                         double* wcpoints2_gam,NURBSData* Data_Basis,
                         int i,int j,double s,double t,double Jdet,
                         double lambda,double mu);
// returns \gamma_{1,y}\check{U}(s,t)*\tilde{g}(t)*\tilde{R}_i(s)*Jdet,
// where s!=t in [a,b), i,k in {1-p,...,N-#b+1} and Jdet in \R
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```


double SquareIntegral_K_Identical(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                  double* wcpoints2_gam,NURBSData* Data_Basis,
                                  QuadData* Data_Gauss,int i,int j,int l,
                                  double lambda,double mu);
// returns \int_0^1 \int_0^1 \gamma_{1,y}\check{U}_{l,l}(s,t)
// \tilde{R}_{i,l}(s) \tilde{g}(t) dt ds,
// where i in {1-p,...,N-#b+1} and l in {max(i,1),...,min(i+p,N)} with H_l>0


double SquareIntegral_K_Adjacent(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                 double* wcpoints2_gam,NURBSData* Data_Basis,
                                 QuadData* Data_Gauss,int i,int j,int l1,
                                 int l2,int singtype,double lambda,double mu);
// returns \int_0^1 \int_0^1 \gamma_{1,y}\check{U}_{l1,l2}(s,t)
// \tilde{R}_{i,l1}(s) \tilde{g}(t) dt ds for adjacent elements,
// i.e. \gamma([t_{l_1-1},t_{l_1} \cap \gamma([t_{l_2-1},t_{l_2}]) consists
// of one point, if singtype=0: singularity at s=0 and t=1
// else singularity at s=1 and t=0,
// i in {1-p,...,N-#b+1}, l1 in {max(i,1),...,min(i+p,N)} and l2 in {1,...,N}
// with min(H_l1,H_l2)>0


void build_Fvector(double* output,NURBSData* Data_Gamma,double* wcpoints1_gam,
                   double* wcpoints2_gam,NURBSData* Data_Basis,
                   QuadData* Data_Gauss,QuadData* Data_Gauss_small,int with_K,
                   double lambda,double mu);
// turns output[i+p-1] into <Kg+g/2,\hat{R}_i>_{L_2(Gamma)} for i=1-p...N-#b+1,
// \hat{R}_i=R_{i,p} \circ gamma^(-1) are the transformed basis functions


#endif
```

Listing A.4: Fvector.c

```
#include "Fvector.h"

double PartDeriv_FundamentalSol(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                double* wcpoints2_gam,int i,int k,int m,
                                double s,double t,double Jdet,double lambda,
                                double mu){

    double den=4*M_PI*mu*(2*mu+lambda);
    double deriv=0;
    double deriv_part=0;
    double tmp1[2];
    double tmp2[2];
    double diff_gam[2]; // gamma(s)-gamma(t)

    eval_NURBSCurve(tmp1,Data_Gamma,wcpoints1_gam,wcpoints2_gam,s);
    // gamma(s)
    eval_NURBSCurve(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma(t)
    diff_gam[0] = tmp1[0] - tmp2[0];
    diff_gam[1] = tmp1[1] - tmp2[1];

    if (i==k){
        deriv_part += diff_gam[m]/norm(diff_gam);
    }
    if (i==m){
        deriv_part += diff_gam[k]/norm(diff_gam);
    }
    deriv = (-1)*(Jdet/norm(diff_gam))*deriv_part;
    deriv += 2*(diff_gam[i]/norm(diff_gam))*(diff_gam[k]/norm(diff_gam))*
            (diff_gam[m]/norm(diff_gam))*(Jdet/norm(diff_gam));
    deriv *= (mu+lambda)/den;
    if (k==m){
        deriv += (3*mu+lambda)/den*((diff_gam[i])/norm(diff_gam))*
                 (Jdet/norm(diff_gam));
    }
    return deriv;
}


double Sigma(NURBSData* Data_Gamma,double* wcpoints1_gam,
             double* wcpoints2_gam,int j,int p,int q,
             double s,double t,double Jdet,double lambda,double mu){

    int r;
    double div=0;
    double sigma_val=0;
    double eps_j_pq=0;

    if (p==q){
        for (r=0;r<=1;r=r+1){
            div += PartDeriv_FundamentalSol(Data_Gamma,wcpoints1_gam,
                                            wcpoints2_gam,r,j,r,s,t,Jdet,
                                            lambda,mu);
        }
        sigma_val += lambda*div;
    }
    sigma_val += mu*PartDeriv_FundamentalSol(Data_Gamma,wcpoints1_gam,
                                               wcpoints2_gam,p,j,q,s,t,Jdet,
                                               lambda,mu);
    sigma_val += mu*PartDeriv_FundamentalSol(Data_Gamma,wcpoints1_gam,
                                               wcpoints2_gam,q,j,p,s,t,Jdet,
                                               lambda,mu);
    return sigma_val;
}


double SquareIntegrand_K(NURBSData* Data_Gamma,double* wcpoints1_gam,
                         double* wcpoints2_gam,NURBSData* Data_Basis,
                         int i,int j,double s,double t,double Jdet,
                         double lambda,double mu){

    double tmp1[2];
    double tmp2[2];
    double nu[2];
    double g_vec[2];
    int p,q;
    double R_til_i; // \tilde{R}_{i,p}(s), first resp. second
    // coordinate of \check{g}(t)|\gamma'(t)|\nu(\gamma(t))
    double squareint=0;


    eval_NURBSCurve(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma(t)
    g(g_vec,tmp2);
    // g_vec = g(gamma(t))
    eval_NURBSCurveDeriv(tmp1,Data_Gamma,wcpoints1_gam,wcpoints2_gam,s);
    // gamma'(s)
    eval_NURBSCurveDeriv(tmp2,Data_Gamma,wcpoints1_gam,wcpoints2_gam,t);
    // gamma'(t)
    R_til_i = eval_NURBS(Data_Basis,i,s) * norm(tmp1);
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
    nu[0] = tmp2[1];
    nu[1] = -tmp2[0];

    for (p=0;p<=1;p=p+1) {
        for (q=0;q<=1;q=q+1) {
            // conormal derivative
            squareint += nu[q] * g_vec[p] * Sigma(Data_Gamma,wcpoints1_gam,
                                                  wcpoints2_gam,j,p,q,s,t,
                                                  Jdet,lambda,mu);
        }
    }
    return squareint * R_til_i;
}


double SquareIntegral_K_Identical(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                  double* wcpoints2_gam,NURBSData* Data_Basis,
                                  QuadData* Data_Gauss,int i,int j,int l,
                                  double lambda,double mu){

    double* nodes_gauss=get_QuadData_nodes(Data_Gauss);
    double* weights_gauss=get_QuadData_weights(Data_Gauss);
    int n_gauss=get_QuadData_n(Data_Gauss);
    int q1,q2;
    double t_lm1=knotseq(Data_Basis,l-1); // t_{l-1}
    double t_l=knotseq(Data_Basis,l); // t_{l}
    double H_l=t_l-t_lm1; // H_l
    double squareint=0; // integral over square
    double intpoint1, intpoint2; // first and second integration point
    double Jdet; // Jacobi determinant for Duffy transformation
    double tmp1,tmp2;

    for (q1=0;q1<n_gauss;q1=q1+1) {
        for (q2=0;q2<n_gauss;q2=q2+1) {
            // first double integral
            intpoint1=t_lm1+H_l*nodes_gauss[q1]*nodes_gauss[q2];
            intpoint2=t_lm1+H_l*nodes_gauss[q1];
            Jdet=nodes_gauss[q1];
            tmp1=SquareIntegrand_K(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                   Data_Basis,i,j,intpoint1,intpoint2,Jdet,
                                   lambda,mu);
            squareint+=tmp1*weights_gauss[q1]*weights_gauss[q2];
            // second double integral
            intpoint1=t_lm1+H_l*nodes_gauss[q1];
            intpoint2=t_lm1+H_l*nodes_gauss[q1]*nodes_gauss[q2];
            Jdet=nodes_gauss[q1];
            tmp2=SquareIntegrand_K(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                   Data_Basis,i,j,intpoint1,intpoint2,Jdet,
                                   lambda,mu);
            squareint+=tmp2*weights_gauss[q1]*weights_gauss[q2];
        }
    }
    return squareint;
}


double SquareIntegral_K_Adjacent(NURBSData* Data_Gamma,double* wcpoints1_gam,
                                 double* wcpoints2_gam,NURBSData* Data_Basis,
                                 QuadData* Data_Gauss,
                                 int i,int j,int l1,int l2,int singtype,
                                 double lambda,double mu){

    double* nodes_gauss=get_QuadData_nodes(Data_Gauss);
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
    double* weights_gauss=get_QuadData_weights(Data_Gauss);
    int n_gauss=get_QuadData_n(Data_Gauss);
    int q1,q2;
    double t_l1m1=knotseq(Data_Basis,l1-1); // t_{l_1-1}
    double t_l1=knotseq(Data_Basis,l1); // t_{l_1}
    double H_l1=t_l1-t_l1m1; // H_{l_1}
    double t_l2m1=knotseq(Data_Basis,l2-1); // t_{l_2-1}
    double t_l2=knotseq(Data_Basis,l2); // t_{l_2}
    double H_l2=t_l2-t_l2m1; // H_{l_1}
    double squareint=0; // integral over square
    double intpoint1, intpoint2; // first and second integration point
    double Jdet; // Jacobi determinant of Duffy transformation

    if (singtype==0){
        for (q1=0;q1<n_gauss;q1=q1+1) {
            for (q2=0;q2<n_gauss;q2=q2+1) {
                // first double integral
                intpoint1=t_l1m1+H_l1*nodes_gauss[q1];
                intpoint2 = t_l2m1+H_l2*(1-nodes_gauss[q1]*nodes_gauss[q2]);
                Jdet=nodes_gauss[q1];
                squareint+=weights_gauss[q1]*weights_gauss[q2]
                *SquareIntegrand_K(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                   Data_Basis,i,j,intpoint1,intpoint2,Jdet,
                                   lambda,mu);
                // second double integral
                intpoint1=t_l1m1+H_l1*nodes_gauss[q1]*(1-nodes_gauss[q2]);
                intpoint2 = t_l2m1 + H_l2 * nodes_gauss[q2];
                Jdet=1-nodes_gauss[q2];
                squareint+=weights_gauss[q1]*weights_gauss[q2]
                *SquareIntegrand_K(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                   Data_Basis,i,j,intpoint1,intpoint2,Jdet,
                                   lambda,mu);
            }
        }
    } else {
        for (q1=0;q1<n_gauss;q1=q1+1) {
            for (q2=0;q2<n_gauss;q2=q2+1) {
                // first double integral
                intpoint1=t_l1m1+H_l1*(1-nodes_gauss[q1]*nodes_gauss[q2]);
                intpoint2 = t_l2m1 + H_l2 *nodes_gauss[q2];
                Jdet=nodes_gauss[q2];
                squareint+=weights_gauss[q1]*weights_gauss[q2]
                *SquareIntegrand_K(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                   Data_Basis,i,j,intpoint1,intpoint2,Jdet,
                                   lambda,mu);
                // second double integral
                intpoint1=t_l1m1+H_l1*nodes_gauss[q1];
                intpoint2 = t_l2m1
                + H_l2 * nodes_gauss[q2] * (1 - nodes_gauss[q1]);
                Jdet=1-nodes_gauss[q1];
                squareint+=weights_gauss[q1]*weights_gauss[q2]
                *SquareIntegrand_K(Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                   Data_Basis,i,j,intpoint1,intpoint2,Jdet,
                                   lambda,mu);
            }
        }
    }
    return squareint;
}


void build_Fvector(double* output,NURBSData* Data_Gamma,double* wcpoints1_gam,
                   double* wcpoints2_gam,NURBSData* Data_Basis,
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
                   QuadData* Data_Gauss,QuadData* Data_Gauss_small,
                   int with_K,double lambda,double mu){

    double* nodes_gauss=get_QuadData_nodes(Data_Gauss);
    double* weights_gauss=get_QuadData_weights(Data_Gauss);
    int n_gauss=get_QuadData_n(Data_Gauss);
    double* nodes_gauss_small=get_QuadData_nodes(Data_Gauss_small);
    double* weights_gauss_small=get_QuadData_weights(Data_Gauss_small);
    int n_gauss_small=get_QuadData_n(Data_Gauss_small);
    double* knots_gam=get_NURBSData_knots(Data_Gamma);
    int N_gam=get_NURBSData_N(Data_Gamma);
    double* knots=get_NURBSData_knots(Data_Basis);
    int N=get_NURBSData_N(Data_Basis);
    int p=get_NURBSData_p(Data_Basis);
    double tmp[2];
    double g_vec[2];
        int i,j,k,l1,l2,q1,q2;
    double b=knots[N-1];
        int multb=0; // #b
    while (nearly_equal(knotseq(Data_Basis,N-multb),b)) {multb=multb+1;}
        double R_til[N-multb+1+p][p+1][n_gauss];
    // R_til[i-1+p][l1-i][q1]=\tilde{R}_{i,l1}(nodes_gauss[q1])
    double squareint; // integral over square
    double Jdet; // Jacobi determinant of Duffy transformation
    double t_l1m1,t_l1,H_l1,t_l2m1,t_l2,H_l2;
    // t_{l1-1},t_l1,H_l1,t_{l2-1},t_l2,H_l2
    double intpoint1, intpoint2; // first and second integration point
    int index; // help index

    // calculation of R_til
    // R_i
    for (i=1-p;i<=(N-multb+1);i=i+1) {
        // elements with nonemty intersection with support of R_i
        for (l1=max(i,1);l1<=min(i+p,N);l1=l1+1) {
            // quadrature points
            for (q1=0;q1<n_gauss;q1=q1+1) {
                t_l1m1=knotseq(Data_Basis,l1-1);
                t_l1=knotseq(Data_Basis,l1);
                H_l1=t_l1-t_l1m1;
                intpoint1=t_l1m1+H_l1*nodes_gauss[q1];
                eval_NURBSCurveDeriv(tmp,Data_Gamma,wcpoints1_gam,
                                     wcpoints2_gam,intpoint1);
                R_til[i-1+p][l1-i][q1] = eval_NURBS(Data_Basis,i,intpoint1)
                * norm(tmp);
            }
        }
    }



    if (with_K==1){
        // calculation of <Kg,\hat{R}_i^j>_{L_2(Gamma)}
        for (j=0;j<=1;j=j+1) {
            // R_i
            for (i=1-p;i<=(N-multb+1);i=i+1) {
                output[i+p-1+j*(N-multb+1+p)]=0;
                // elements with nonemty intersection with support of R_i
                for (l1=max(i,1);l1<=min(i+p,N);l1=l1+1) {
                    // all elements
                    for (l2=1;l2<=N;l2=l2+1) {
                        t_l1m1=knotseq(Data_Basis,l1-1);
                        t_l1=knotseq(Data_Basis,l1);
                        t_l2m1=knotseq(Data_Basis,l2-1);
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
                        t_l2=knotseq(Data_Basis,l2);
                        H_l1=t_l1-t_l1m1;
                        H_l2=t_l2-t_l2m1;
                        // quadrature
                        if (0<min(H_l1,H_l2)){
                            squareint=0;
                            // elements with no intersection
                            if ((!nearly_equal(t_l1m1,t_l2m1))
                                && (!nearly_equal(t_l1m1,t_l2))
                                && (!nearly_equal(t_l1,t_l2m1))
                                && (!nearly_equal(t_l1,t_l2))
                                && ((l1!=(N-multb+1)) || (l2!=1))
                                && ((l2!=(N-multb+1)) || (l1!=1))) {
                                for (q1=0;q1<n_gauss;q1=q1+1) {
                                    for (q2=0;q2<n_gauss;q2=q2+1) {
                                        intpoint1 = t_l1m1+H_l1*nodes_gauss[q1];
                                        intpoint2 = t_l2m1+H_l2*nodes_gauss[q2];

                                        squareint +=
                                        weights_gauss[q1]*weights_gauss[q2]*
                                        SquareIntegrand_K(Data_Gamma,
                                        wcpoints1_gam,wcpoints2_gam,
                                        Data_Basis,i,j,intpoint1,intpoint2,
                                        1,lambda,mu);
                                    }
                                }
                            }
                            // elements with intersection
                            else {
                                // identical elements
                                if (l1==l2){
                                    squareint=SquareIntegral_K_Identical(
                                              Data_Gamma,wcpoints1_gam,
                                              wcpoints2_gam,Data_Basis,
                                              Data_Gauss_small,i,j,l1,
                                              lambda,mu);
                                }
                                // elements with point intersection
                                else {
                                    // singularity at s=0,t=1
                                    if (nearly_equal(t_l1m1,t_l2)
                                        || ((l2==(N-multb+1)) && (l1==1))){
                                        index=1;
                                        while(!nearly_equal(t_l2,knots_gam[index-1])){
                                            index=index+1;
                                            if (index==(N_gam+1)){break;}
                                        }
                                        // t_l2 no knot of Gamma
                                        if (index==(N_gam+1)){
                                            squareint=SquareIntegral_K_Adjacent(
                                                      Data_Gamma,wcpoints1_gam,
                                                      wcpoints2_gam,Data_Basis,
                                                      Data_Gauss_small,i,j,
                                                      l1,l2,0,lambda,mu);
                                        }
                                        // t_l2 knot of Gamma
                                        else {
                                            squareint=SquareIntegral_K_Adjacent(
                                                      Data_Gamma,wcpoints1_gam,
                                                      wcpoints2_gam,Data_Basis,
                                                      Data_Gauss,i,j,l1,l2,0,
                                                      lambda,mu);
                                        }
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

```
                                    }
                                    // singularity at s=1,t=0
                                    else{
                                        index=1;
                                        while(!nearly_equal(t_l1,knots_gam[index-1])){
                                            index=index+1;
                                            if (index==(N_gam+1)){break;}
                                        }
                                        // t_l1 no knot of Gamma
                                        if (index==(N_gam+1)){
                                            squareint=SquareIntegral_K_Adjacent(
                                                      Data_Gamma,wcpoints1_gam,
                                                      wcpoints2_gam,Data_Basis,
                                                      Data_Gauss_small,i,j,
                                                      l1,l2,1,lambda,mu);
                                        }
                                        // t_l1 knot of Gamma
                                        else {
                                            squareint=SquareIntegral_K_Adjacent(
                                                      Data_Gamma,wcpoints1_gam,
                                                      wcpoints2_gam,Data_Basis,
                                                      Data_Gauss,i,j,l1,l2,1,
                                                      lambda,mu);
                                        }
                                    }
                                }
                            }
                            output[i+p-1+j*(N-multb+1+p)] += H_l1 * H_l2 * squareint;
                        }
                    }
                }
            }
        }
    }


    // calculation of <Kg+g/2,R_hat_i>_{L_2(Gamma)}
    for (j=0;j<=1;j=j+1) {
    // R_i
        for (i=1-p;i<=(N-multb+1);i=i+1) {
            if (with_K==0){
                output[i+p-1+j*(N-multb+1+p)]=0;
            }
            // elements with nonemty intersection with support of R_i
            for (l1=max(i,1);l1<=min(i+p,N);l1=l1+1) {
                t_l1m1=knotseq(Data_Basis,l1-1); // t_{l1-1}
                t_l1=knotseq(Data_Basis,l1); // t_l1
                H_l1=t_l1-t_l1m1;
                // quadrature
                if (0<H_l1) {
                    for (q1=0;q1<n_gauss;q1=q1+1) {
                        eval_NURBSCurve(tmp,Data_Gamma,wcpoints1_gam,wcpoints2_gam,
                                        t_l1m1+H_l1*nodes_gauss[q1]);
                        g(g_vec,tmp);
                        output[i+p-1+j*(N-multb+1+p)] += H_l1 *
                        weights_gauss[q1] * g_vec[j]/2 * R_til[i-1+p][l1-i][q1];
                    }
                }
            }
        }
    }
}
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bibliography

[CP06] Carsten Carstensen and Dirk Praetorius. Averaging techniques for the effective numerical solution of Symm's integral equation of the first kind. *SIAM J. Sci. Comput.*, 27(4):1226–1260, 2006.

[dB86] Carl de Boor. B(asic)-spline basics. Technical report, Mathematics Research Center, University of Wisconsin-Madison, 1986.

[FGHP16] Michael Feischl, Gregor Gantner, Alexander Haberl, and Dirk Praetorius. Adaptive 2D IGA boundary element methods. *Eng. Anal. Bound. Elem.*, 62:141–153, 2016.

[FGHP17] Michael Feischl, Gregor Gantner, Alexander Haberl, and Dirk Praetorius. Optimal convergence for adaptive IGA boundary element methods for weakly-singular integral equations. *Numer. Math.*, 136(1):147–182, 2017.

[FGP15] Michael Feischl, Gregor Gantner, and Dirk Praetorius. Reliable and efficient a posteriori error estimation for adaptive iga boundary element methods for weakly-singular integral equations. *Computer Methods in Applied Mechanics and Engineering*, 290:362 – 386, 2015.

[Gan14] Gregor Gantner. Adaptive isogeometric BEM. Master thesis, TU Wien, Institute for Analysis and Scientific Computing, 2014.

[LL59] Lev Davidovich Landau and Evgeny Mikhailovich Lifshitz. *Theory of elasticity.* Course of Theoretical Physics, Vol. 7. Translated by J. B. Sykes and W. H. Reid. Pergamon Press, London-Paris-Frankfurt; Addison-Wesley Publishing Co., Inc., Reading, Mass., 1959.

[McL00] William McLean. *Strongly elliptic systems and boundary integral equations.* Cambridge University Press, Cambridge [u.a.], 1. publ. edition, 2000.

[ME14] Gregor Mitscha-Eibl. Adaptive BEM und FEM-BEM-Kopplung für die Lamé-Gleichung. Master thesis, TU Wien, Institute for Analysis and Scientific Computing, 2014.

[NH80] Jindřich Nečas and Ivan Hlaváček. *Mathematical theory of elastic and elasto-plastic bodies: an introduction*, volume 3 of *Studies in Applied Mechanics.* Elsevier Scientific Publishing Co., Amsterdam-New York, 1980.

[Pra07] Dirk Praetorius. Boundary element method. Lecture Notes, TU Wien, 2007.

[Pra10] Dirk Praetorius. Numerische Mathematik. Lecture Notes, TU Wien, 2010.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[Pra17] Dirk Praetorius. Finite element method. Lecture Notes, TU Wien, 2017.

[SS11] Stefan A. Sauter and Christoph Schwab. *Boundary element methods.* Springer, Berlin, 2011.

[Ste08] Olaf Steinbach. *Numerical Approximation Methods for Elliptic Boundary Value Problems.* Springer, New York, 2008.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar.
The approved original version of this thesis is available in print at TU Wien Bibliothek.