Masterarbeit

Numerical Solution of the Wave Equation in Unbounded Domains

Martin Huber

ausgeführt unter der Leitung von
Prof. Dr. Stefan Sauter
und
Alexander Veit

Zürich, im März 2011
# Contents

1 Introduction .............................................. 1

2 Formulation of the Problem ......................... 3

3 Mathematical Framework ................................. 7
   3.1 Discrete Fourier Transform .......................... 7
   3.2 Fourier Transform .................................... 8
   3.3 Laplace Transform ................................... 9
   3.4 Sobolev Spaces ........................................
      3.4.1 Sobolev Spaces of Integer Order on Domains . 11
      3.4.2 Sobolev Spaces $H^s(\Omega)$, for $s \in \mathbb{R}_{\geq 0}$ . 12
      3.4.3 Sobolev Spaces on Surfaces ...................... 13
      3.4.4 Traces, Liftings and Extensions ................ 14
      3.4.5 Dual Spaces of Sobolev Spaces .................. 15
      3.4.6 Sobolev Spaces for Problems in Time and Space 15
   3.5 Weak Formulation of Equations .................... 16
   3.6 Derivation of the Convolution Quadrature .......... 18
   3.7 Derivation of BDF2 ................................... 21
   3.8 Spherical Harmonics .................................. 22

4 Theory for the Exact Problem ......................... 23
   4.1 Properties of the Wave Equation ................... 23
   4.2 Existence and Uniqueness for the full space Problem 24
   4.3 Motivation for the Single Layer Potential .......... 24
   4.4 Connection to the Helmholtz Equation ............... 25
   4.5 Theory for the Helmholtz Problem .................. 26
   4.6 Properties of the Single Layer Potential .......... 28
   4.7 Existence and Uniqueness for the Single Layer Potential 30

5 Numerical Discretisation ............................... 35
   5.1 Time Discretisation .................................. 35
   5.2 Space Discretisation ................................
      5.2.1 Surface Paneling ................................ 36
      5.2.2 Piecewise Constant Boundary Elements .......... 38
      5.2.3 Piecewise Linear Boundary Elements ............. 38
      5.2.4 Discretisation for a Given Basis ............... 39
   5.3 Approximation of the Convolution Weights .......... 41
   5.4 Decoupling the Systems .............................. 42
   5.5 Approximation of the Wave Function ............... 44
<table>
<thead>
<tr>
<th>6</th>
<th>Theory for the Discretised Problem</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Boundary Element Space</td>
<td>47</td>
</tr>
<tr>
<td>6.2</td>
<td>Approximation of the Convolution Weights</td>
<td>47</td>
</tr>
<tr>
<td>6.3</td>
<td>Convergence of the Method</td>
<td>48</td>
</tr>
<tr>
<td>7</td>
<td>MATLAB Implementation</td>
<td>51</td>
</tr>
<tr>
<td>7.1</td>
<td>Algorithm</td>
<td>51</td>
</tr>
<tr>
<td>7.2</td>
<td>Generating a Mesh</td>
<td>53</td>
</tr>
<tr>
<td>7.3</td>
<td>Data Structures for Mesh Handling and Global Parameters</td>
<td>53</td>
</tr>
<tr>
<td>7.4</td>
<td>Numerical Integration</td>
<td>58</td>
</tr>
<tr>
<td>7.5</td>
<td>Routines</td>
<td>61</td>
</tr>
<tr>
<td>7.5.1</td>
<td>Arranging the Pairs of Triangles</td>
<td>62</td>
</tr>
<tr>
<td>7.5.2</td>
<td>Calculating the System Matrices</td>
<td>62</td>
</tr>
<tr>
<td>7.5.3</td>
<td>Calculating the Right-Hand Side Vectors</td>
<td>63</td>
</tr>
<tr>
<td>7.5.4</td>
<td>Implementing the Scaled Fourier Transform</td>
<td>65</td>
</tr>
<tr>
<td>7.5.5</td>
<td>Calculating the Solution of the Wave Equation</td>
<td>65</td>
</tr>
<tr>
<td>7.5.6</td>
<td>Description of the Routines</td>
<td>66</td>
</tr>
<tr>
<td>7.6</td>
<td>Storage Costs</td>
<td>71</td>
</tr>
<tr>
<td>8</td>
<td>Numerical Tests</td>
<td>73</td>
</tr>
<tr>
<td>8.1</td>
<td>Test Case with known Analytical Solution</td>
<td>73</td>
</tr>
<tr>
<td>8.2</td>
<td>Error Measure</td>
<td>73</td>
</tr>
<tr>
<td>8.3</td>
<td>Convergence Tests</td>
<td>74</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Piecewise Constant Elements</td>
<td>76</td>
</tr>
<tr>
<td>8.3.2</td>
<td>Piecewise Linear Elements</td>
<td>78</td>
</tr>
<tr>
<td>8.4</td>
<td>Complexity Tests</td>
<td>79</td>
</tr>
<tr>
<td>8.5</td>
<td>Acoustic Tests</td>
<td>84</td>
</tr>
<tr>
<td>9</td>
<td>Conclusion</td>
<td>87</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Wave propagation is a fundamental process in nature. Hence the concept of waves arises in several disciplines of physics, although the physical processes, which are modeled by waves, differ fundamentally. On the one hand there are mechanical waves like water waves or acoustic waves, which are bounded to a medium, and on the other hand electromagnetic waves, like visible light or radio waves, that propagate even in the vacuum. Furthermore, the concept of waves is also used in quantum mechanics, as matter waves, as well as in the general theory of relativity, where gravitational waves are predicted. The common property of wave propagation is the transport of energy through the space without transporting any mass. This transport is obtained by a time dependent conversion of two physical quantities, e.g. kinetic and potential energy in the case of mechanical waves.

In the field of research of acoustics the scattering of acoustic waves at an obstacle is a common problem. A classical domain of that research is the analysis of the acoustics of buildings, e.g. operas, theaters or churches. In these buildings acoustic waves produced by human voice or musical instruments are supposed to reach the listeners as well as it is possible. The contrary intention governs in the domain of noise barriers, where scattering problems directly affect the everyday life. To protect residential areas efficiently and cost-effectively against noise, various tests about the placement and the materials of the barriers are necessary. Moreover, the problem of scattered waves arises as well in other domains, such as electromagnetic waves in the sonar technology, where conversely the scattered wave is received and the original wave has to be determined. Therefore, the investigation of incoming waves scattered at an obstacle delivers insights into various problems arising in applied sciences.

Examining the space where the investigated wave propagates often reveals that this space is an unbounded domain in the three dimensional space, e.g. for acoustic waves in the case of ancient outdoor theaters or noise calculations. Hence, efficient methods for the numerical solution of the wave equation in unbounded domains are needed. Discretizing an unbounded domain for applying a method, which is based on classical finite elements (FEM), leads to several problems, as the boundary at infinity somehow has to be modeled. Thus the approach presented in [BS08], that is based on an integral equation involving just the compact boundary of the scatterer, is applied in this thesis. This integral equation then is resolved with respect to the space by a boundary element method (BEM). That reduces the spatial dimension of the domains by one dimension, as surfaces and not volumes in $\mathbb{R}^3$ arise as origin of the discretisation, which will lead to an essential reduction of the costs of calculations.

The goal of this thesis is to implement a method based on the convolution quadrature with constant step size in time and a Galerkin boundary element method in space as presented in [BS08] and its precedent papers [HKS09] resp. [HKS07]. The calculation of the solution
in the exterior domain is divided into two parts: The main part is the numerical solution of an integral equation. In order to reduce the amount of time needed for the calculation, a special approximation of some convolution weights is applied that will reduce the problem to decoupled Helmholtz problems which can be computed parallely. The Galerkin boundary element method, applied on the weak formulation of a Helmholtz problem, gives rise to integrals of a singular kernel on the domain of the Cartesian product of two surface panels. To calculate these integrals, a specialized quadrature developed in [SS04], based on a tensorised Gauss-quadrature, is applied. This first part will provide an approximation of the unknown density determining the solution in the exterior domain. This solution then is calculated in the second part by evaluating an integral representation including the provided density. This implementation will allow the calculation of the scattering of an incoming wave at a sound soft bounded obstacle in the three dimensional space.

The outline of this thesis is as follows. In Chapter 2 the homogeneous exterior Dirichlet problem is deduced and the basic approach by a retarded single layer potential is introduced. Solving that problem theoretically and numerically needs several concepts and methods. These are presented in a compact manner in Chapter 3 and include topics like Laplace transform, Sobolev spaces on domains and surfaces as well as the derivation of the convolution quadrature and the linear multistep method BDF2.

In a second step in Chapter 4 the properties of the wave equation and its solution are examined. This also involves physical principles and an integral representation of the solution of the wave equation. This motivates the chosen approach by a retarded potential, whose properties are investigated consecutively. Applying the Laplace transform on a solution of the wave equation will show the strong link to the Helmholtz equation. As the Laplace transform of an integral kernel also is fundamental in the convolution quadrature, this connection is the basis for any existence and uniqueness result. A calculus presented in [Lub94] allows the transfer of the results obtained for the Helmholtz equation back to the wave equation, resulting in the theorem about existence and uniqueness for the retarded potential approach.

In Chapter 5 the numerical discretisation of the homogeneous exterior Dirichlet problem is presented. First of all the time discretisation via the convolution quadrature is introduced. After the definition of the boundary element spaces, restricted to the case of discontinuous piecewise constant resp. continuous piecewise linear elements, the Galerkin boundary element method can be established. Both discretisations together result in a Toeplitz system, which is decoupled by applying a Fourier style approximation of the convolution weights. In addition, the framework for the calculation of the exterior solution is given based on a linear interpolation in time. The existence, convergence and stability results for that presented discretisation is then stated in Chapter 6.

The main part of this thesis builds a MATLAB implementation of the introduced discretisation scheme, which is presented in Chapter 7 in detail. This includes the description of data structures and routines along with details about the quadrature methods. Furthermore, the complexity of the implementation is examined. As last step, the implementation is validated by checking the predicted convergence rates. For this purpose an error measure is developed and Dirichlet boundary conditions are presented, where the analytical solution is known exactly.
Chapter 2

Formulation of the Problem

This chapter is dedicated to the formulation and the motivation of the mathematical problem that we want to solve. The goal is to introduce it in a short but clear manner without any further theory, as that will be presented in the subsequent chapters.

Let $\Omega := \Omega^e \subset \mathbb{R}^3$ be an unbounded connected Lipschitz domain, called the exterior domain, with bounded complement $\Omega^i := \mathbb{R}^3 \setminus \overline{\Omega}$, the interior domain, and therefore bounded, compact boundary $\Gamma := \partial \Omega$. Assume that the inhomogeneity $f : \Omega \times (0,T) \to \mathbb{R}$ and the initial values $u_0, u_1 : \Omega \to \mathbb{R}$ are given. Consider now the following exterior scattering problem

$$\partial_t^2 u(x,t) - \Delta u(x,t) = f(x,t) \quad \text{in } \Omega \times (0,T) \quad (2.1a)$$
$$u(x,0) = u_0(x) \quad \forall x \in \Omega \quad (2.1b)$$
$$\partial_t u(x,0) = u_1(x) \quad \forall x \in \Omega \quad (2.1c)$$
$$u(x,t) = 0 \quad \text{on } \Gamma \times (0,T), \quad (2.1d)$$

for a fixed end time $T > 0$. That original problem, given by (2.1) and called $P_{\text{orig}}$, is an inhomogeneous exterior Dirichlet problem.

To get a boundary integral equation, we have to reduce $P_{\text{orig}}$ to a homogeneous problem. That is achieved by splitting the wave function $u$ into a part consisting of the incident wave $u_{\text{inc}}$ and a part consisting of the scattered wave $u_{\text{sca}}$. For $u_{\text{inc}}$ arises then the problem

$$\partial_t^2 u_{\text{inc}}(x,t) - \Delta u_{\text{inc}}(x,t) = \mathcal{f}(x,t) \quad \text{in } \mathbb{R}^3 \times (0,T) \quad (2.2a)$$
$$u_{\text{inc}}(x,0) = \mathcal{u}_0(x) \quad \forall x \in \mathbb{R}^3 \quad (2.2b)$$
$$\partial_t u_{\text{inc}}(x,0) = \mathcal{u}_1(x) \quad \forall x \in \mathbb{R}^3 \quad (2.2c)$$

where $\mathcal{f}$, $\mathcal{u}_0$, and $\mathcal{u}_1$ are prolongations of $f$, $u_0$, and $u_1$ to the full space $\mathbb{R}^3$ in the sense of Section 3.4.4. This problem for the incident wave defined by (2.2) is denoted by $P_{\text{inc}}$ and is a so called Cauchy problem for the wave equation on the full space.

Given a solution of $P_{\text{inc}}$ called $u_{\text{inc}}$, whose existence is discussed in Section 4.2 we are able to formulate the homogeneous exterior Dirichlet problem arising for $u_{\text{sca}}$

$$\partial_t^2 u_{\text{sca}}(x,t) - \Delta u_{\text{sca}}(x,t) = 0 \quad \text{in } \Omega \times (0,T) \quad (2.3a)$$
$$u_{\text{sca}}(x,0) = 0 \quad \forall x \in \Omega \quad (2.3b)$$
$$\partial_t u_{\text{sca}}(x,0) = 0 \quad \forall x \in \Omega \quad (2.3c)$$
$$u_{\text{sca}}(x,t) = -u_{\text{inc}}(x,t) \quad \text{on } \Gamma \times (0,T), \quad (2.3d)$$

denoted as $P_{\text{sca}}$. The goal of this thesis is to develop an efficient numerical solver for this problem. Merging the results of $P_{\text{inc}}$ and $P_{\text{sca}}$ leads to the solution for $P_{\text{orig}}$ by taking the superposition $u(x,t) := u_{\text{inc}}(x,t) + u_{\text{sca}}(x,t)$. 

3
As the interests of this thesis are focused on the scattering of an incoming wave by an obstacle and not on the propagation nor the formation of a wave in the full space, the problem $P_{\text{inc}}$ will not be considered in the numerical examples. We restrict ourselves to the case of a transient wave $u_{\text{inc}}$, i.e. $u_{\text{inc}}$ satisfies $P_{\text{inc}}$ with $f \equiv 0$ in $\Omega \times (0, T)$ and $u_{\text{inc}}$ vanishes in a neighbourhood of $\Omega^e$ at $t = 0$. That problem has its application for example in acoustics.

![Figure 2.1: Scattering of an incoming wave by an obstacle](image)

It models a scattering problem with an obstacle occupying $\Omega^i$ and placed into a homogeneous medium, e.g. fluids or air, that covers the unbounded exterior domain $\Omega^e$. At the time $t = 0$, a neighbourhood of the obstacle is not perturbed by any acoustic pressure. A wave $u_{\text{inc}}$, that arises from the far field, propagates with finite velocity and without any disturbance onto the obstacle. Starting at the point, where the incoming wave reaches the obstacle, a scattered wave appears, as the obstacle reflects some parts of the incoming wave. The amount, which is reflected, depends on the material properties of the obstacle. In the studied case the obstacle is a soft scatterer, which is modeled by imposing Dirichlet conditions on the boundary $\Gamma$. The cases of a hard resp. absorbing scatterer would be modeled by Neumann resp. absorbing boundary conditions (cf. [HD03]).

After that motivation by acoustics, we rewrite the problem $P_{\text{sca}}$ with simpler notation as

\begin{align}
\partial_t^2 u(x, t) - \Delta u(x, t) &= 0 \quad \text{in } \Omega \times (0, T) \\
u(x, 0) &= 0 \quad \forall x \in \Omega \\
v_t(x, 0) &= 0 \quad \forall x \in \Omega \\
\partial n u(x, t) &= g(x, t) \quad \text{on } \Gamma \times (0, T),
\end{align}

for a given $g$ on $\Gamma \times (0, T)$ and call it $P$ for further reference.

Motivated by the results of classical electromagnetic theory (cf. Section 4.3), we employ a
single layer potential ansatz for the unknown solution $u$

$$(S\varphi)(x,t) := \int_0^t \int_{\Gamma} k(x - y, t - \tau) \varphi(y, \tau) d\Gamma \, d\tau,$$  \hspace{1cm} (2.5)

to solve the problem $P$. The distributional $k$ is given by the fundamental solution of the wave equation (cf. Section 4.6)

$$k(z, s) = \frac{\delta(\|z\| - s)}{4\pi \|z\|}.$$ \hspace{1cm} (2.6)

Any $u := S\varphi$ solves $P$ with the exception of the boundary condition (2.4d) (cf. Section 4.6). Using the continuity (ibid.) of the single layer potential as $\Omega \ni x \to \Gamma$ the inner integral of the right-hand side in (2.5) has to be understood as an improper Riemann integral

$$(V\varphi)(x,t) := \int_0^t \int_{\Gamma} k(x - y, t - \tau) \varphi(y, \tau) d\Gamma \, d\tau \quad (x,t) \in \Gamma \times (0,T).$$ \hspace{1cm} (2.7)

To fulfill the boundary condition (2.4d) the integral equation

$$V\varphi = g \quad \text{on} \quad \Gamma \times (0,T).$$ \hspace{1cm} (2.8)

has to be solved. At the core of this thesis a method to solve this retarded potential integral equation numerically is developed.
CHAPTER 2. FORMULATION OF THE PROBLEM
Chapter 3

Mathematical Framework

During the development of the theoretical background and the numerical discretisation, various mathematical concepts are used. The aim of this chapter is to provide the analytic tools which later will be employed for the mathematical analysis for the retarded potential integral equation. For the proofs and further results we will give reference to the literature.

3.1 Discrete Fourier Transform

By approximating an integral of the form of the Cauchy integral formula by using the trapezoidal rule on a circle, the arising summation can be regarded as a certain formulation of the discrete Fourier transform. The goal is to provide the way on which it can be reduced to the discrete Fourier transform, in order to use the Fast Fourier Transform (FFT) for the implementation. Further theory and results can be found in [Hen86].

Given a vector $a = (a_0, \ldots, a_N) \in \mathbb{C}^{N+1}$, $N \in \mathbb{N}$ the discrete Fourier transform $\hat{a} = (\hat{a}_0, \ldots, \hat{a}_N) \in \mathbb{C}^{N+1}$ is defined by

$$\hat{a}_k = \sum_{j=0}^{N} a_j \zeta_{N+1}^{jk}, \quad (3.1)$$

for $k = 0, \ldots, N$ and $\zeta_{N+1} := e^{-\frac{2\pi i}{N+1}}$. To reconstruct $a$ from $\hat{a}$, one can apply the inverse transform

$$a_k = \frac{1}{N+1} \sum_{j=0}^{N} \hat{a}_j \zeta_{N+1}^{-jk}, \quad k = 0, \ldots, N. \quad (3.2)$$

Let $k \in \{0, \ldots, N\}$ and $\lambda \in \mathbb{R}_{>0}$. Consider the discrete Fourier transform of the vector $b = (\lambda^0 a_0, \ldots, \lambda^N a_N)$, which is the result of a special scaling of the vector $a$,

$$\hat{b}_k = \sum_{j=0}^{N} b_j \zeta_{N+1}^{jk} = \sum_{j=0}^{N} \lambda^j a_j \zeta_{N+1}^{jk}. \quad (3.3)$$

Written in terms of $a$ it holds

$$\lambda^k a_k = b_k = \frac{1}{N+1} \sum_{j=0}^{N} \hat{b}_j \zeta_{N+1}^{-jk} \quad \Rightarrow \quad a_k = \frac{\lambda^{-k}}{N+1} \sum_{j=0}^{N} \hat{b}_j \zeta_{N+1}^{-jk}. \quad (3.4)$$

This motivates the definition of a scaled discrete Fourier transform. The original Fourier transform is obtained for $\lambda = 1$. 

7
Definition 3.1.1. For a vector \( \mathbf{a} = (a_0, \ldots, a_N) \in \mathbb{C}^{N+1} \) and given parameter \( \lambda > 0 \) the scaled discrete Fourier transform is defined by
\[
\hat{a}_k = \sum_{j=0}^{N} \lambda^j a_j \zeta_{N+1}^j.
\] (3.5)

Lemma 3.1.2. The inverse of the scaled discrete Fourier transform is given by
\[
a_k = \frac{\lambda^{-k}}{N+1} \sum_{j=0}^{N} \hat{a}_j \zeta_{N+1}^{-jk}.
\] (3.6)

Remark 3.1.3. This derivation shows that the scaled discrete Fourier transform can be implemented by using the fast Fourier transform and by scaling the argument resp. the result by the matrix \( \Lambda \) resp. its inverse, with
\[
\Lambda := \begin{pmatrix}
\lambda^0 & 0 & & \\
\lambda^1 & \ddots & & \\
0 & \ddots & \ddots & \\
& & \ddots & \lambda^N
\end{pmatrix}.
\] (3.7)

3.2 Fourier Transform

In this section we will sketch some fundamentals of the Fourier transform. For further results and applications we refer to [SS03].

Definition 3.2.1. To express high order multidimensional derivation, it is convenient to use multi-indices. Let \( \alpha \in \mathbb{N}^n \) be a multi-index. Define the following quantities
\[
|\alpha| := \sum_{i=1}^{n} \alpha_i, \quad v^\alpha := \prod_{i=1}^{n} v_i^{\alpha_i} \quad \text{with} \quad v \in \mathbb{C}^n, \quad D^\alpha f(x) := D^\alpha_x f(x) := \frac{\partial^{|\alpha|} f(x)}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}.
\] (3.8)

Next we define the functional space used for the Fourier transform.

Definition 3.2.2. A function \( f : \mathbb{R}^n \to \mathbb{C} \) is called rapidly decreasing if
\[
\sup_{x \in \mathbb{R}^n} |x^\alpha f(x)| < \infty \quad \forall \alpha \in \mathbb{N}^n.
\] (3.9)

The space containing all indefinitely differentiable functions \( f : \mathbb{R}^n \to \mathbb{C} \) with
\[
\sup_{x \in \mathbb{R}^n} |x^\alpha D^\beta f(x)| < \infty \quad \forall \alpha, \beta \in \mathbb{N}^n,
\] (3.10)

is the Schwartz space \( S(\mathbb{R}^n) \).

For any \( f \in S(\mathbb{R}^n) \) all its derivatives are rapidly decreasing. Let us define now the Fourier transform for a function \( f \in S(\mathbb{R}^n) \)

Definition 3.2.3. For \( u \in S(\mathbb{R}^n) \) the Fourier transform \( F u : \mathbb{R}^n \to \mathbb{C} \) is defined by
\[
(Fu)(\xi) := (2\pi)^{-\frac{n}{2}} \int_{\mathbb{R}^n} e^{-i\xi \cdot x} u(x) dx,
\] (3.11)

where \( \cdot \) in the expression \( \xi \cdot x \) denominates the Euclidean scalar product on \( \mathbb{R}^n \).
Next we will recall the formula of Plancherel and the inverse Fourier transform.

**Theorem 3.2.4.** For a function \( u \in \mathcal{S}(\mathbb{R}^n) \), it holds that the \( L^2 \)-norm is preserved by the Fourier transform

\[
\int_{\mathbb{R}^n} |(F u)(\xi)|^2 d\xi = \int_{\mathbb{R}^n} |u(x)|^2 dx.
\]  

(3.12)

Furthermore the inverse mapping \( F^{-1} \), for a \( v \in \mathcal{S}(\mathbb{R}^n) \), is given by

\[
(F^{-1} v)(x) := (2\pi)^{-\frac{n}{2}} \int_{\mathbb{R}^n} e^{ix \cdot \xi} v(\xi) d\xi;
\]  

(3.13)

i.e. \( F^{-1}(F u) = u \) for all \( u \in \mathcal{S}(\mathbb{R}^n) \).

We conclude that the Fourier transform \( F \) is a bijective isometry of \( \mathcal{S}(\mathbb{R}^n) \) onto itself. The Fourier transform has several important properties. Fundamental is the property concerning the transformation of derivatives.

**Lemma 3.2.5.**

\[
F(D^\alpha u)(\xi) = i^{\left|\alpha\right|} |\xi|^{\alpha} (F u)(\xi) \quad \forall \; u \in \mathcal{S}(\mathbb{R}^n).
\]  

(3.14)

This is one of the reasons that the Fourier transform, and later on the Laplace transform, is often used in the context of partial differential equations, as the derivative of a function becomes a simple multiplication in the Fourier image. Therefore it is often simpler to solve a full space problem in the Fourier image and then transform the result back using the inverse Fourier transform.

As we have seen, the Fourier transform preserves the \( L^2 \)-norm. Furthermore the test functions \( C_0^\infty(\mathbb{R}^n) \), defined in Definition 3.4.1, are a subspace of the Schwartz space \( \mathcal{S}(\mathbb{R}^n) \) and lie dense in \( L^2(\mathbb{R}^n) \). Consequently we may extend the Fourier transform to Sobolev spaces, which will be defined in Section 3.4.1. That is done for example in [Hae92, Section 6.2.3]. We just quote the main result.

**Theorem 3.2.6.** \( F \) and \( F^{-1} \) are isometric mappings of \( L^2(\mathbb{R}^n) \) onto itself. The scalar product satisfies

\[
(u, v)_{L^2} = (F u, F v)_{L^2} \quad \forall \; u, v \in L^2(\mathbb{R}^n).
\]  

(3.15)

The theory of the Fourier transform can be extended to tempered distributions, denoted by \( \mathcal{S}'(\mathbb{R}^n) \), the dual space of \( \mathcal{S}(\mathbb{R}^n) \), i.e. the continuous, linear functions on \( \mathcal{S}(\mathbb{R}^n) \). This theory is developed, e.g., in [Jan71].

### 3.3 Laplace Transform

As we will see later, the discretisation with relation to the time strongly relies on the Laplace transform of the distributional integral kernel \( k \). The Laplace transform itself can be interpreted as a Fourier transform for special functions.

**Definition 3.3.1.** Let \( f \) be a function \( f: \mathbb{R} \to \mathbb{R} \) with \( f(t) = 0 \) for \( t \in (-\infty, 0) \), i.e. a causal function. \( f \) is Laplace transformable on the half-plane \( \{ s \in \mathbb{C} \mid \text{Re}(s) > \sigma_0 \} \) if there exists a \( \sigma_0 \in \mathbb{R} \), such that \( e^{-\sigma_0 t} f(t) \in \mathcal{S}(\mathbb{R}) \) i.e. is Fourier transformable.
Definition 3.3.2. For a Laplace transformable function $f : \mathbb{R} \to \mathbb{R}$ and $\sigma_0$ chosen as Definition 3.3.1, the Laplace transform is defined by

$$\hat{f}(s) := (\mathcal{L}f)(s) := \int_{0}^{\infty} f(t)e^{-st}dt, \quad (3.16)$$

for $s \in \mathbb{C}$ with $\text{Re}(s) > \sigma_0$.

Using the strong connection to the Fourier transform, the inverse Laplace transform can be determined.

Lemma 3.3.3. Let $f$ be Laplace transformable and $\sigma_0 \in \mathbb{R}$ as in Definition 3.3.1. Then, the Laplace inversion formula holds

$$f(t) = (\mathcal{L}^{-1}\hat{f})(t) := \frac{1}{2\pi i} \int_{\sigma+i\mathbb{R}} \hat{f}(s)e^{st}ds, \quad t \in \mathbb{R}, \sigma > \sigma_0. \quad (3.17)$$

Similarly as for the Fourier transform, there exists a formula for the Laplace transform of the derivatives of a function. Again the derivatives are turned into a polynomial expression which makes the Laplace transform suitable for partial differential equations that follow a certain causality.

Theorem 3.3.4. Let $n \in \mathbb{N}$. For a Laplace transformable function $f : \mathbb{R} \to \mathbb{R}$, which is smooth enough, it holds

$$(\mathcal{L}f^{(n)})(s) = s^n \hat{f}(s) - \sum_{k=0}^{n-1} s^{n-k-1}f^{(k)}(0). \quad (3.18)$$

Another important property is also inherited from the Fourier transform. It is the rule concerning the Laplace transform of a convolution.

Definition 3.3.5. Let $f \in C^\infty(\mathbb{R}), g \in C_0^\infty(\mathbb{R})$. The convolution $f,g : \mathbb{R} \to \mathbb{R}$ of $f$ and $g$ is defined by

$$(f \ast g)(x) := \int_{\mathbb{R}} f(x-y)g(y)dy, \quad \forall x \in \mathbb{R}. \quad (3.19)$$

Theorem 3.3.6. For the convolution of two Laplace transformable functions $f,g : \mathbb{R} \to \mathbb{R}$ it holds

$$(\mathcal{L}(f \ast g))(s) = \hat{f}(s) \cdot \hat{g}(s). \quad (3.20)$$

The theory for the Laplace transform can be extended to distributions. The basic properties and the rules, as it can be found, e.g., in [Jan71, Chapter 12], remain the same. Therefore, we apply $\mathcal{L}$ to the fundamental solution of the wave equation $k$, which is a fundamental step often used in this thesis. Let us take $k(d,t) = \frac{\delta(d-t)}{4\pi d^2}$, the fundamental solution of the wave equation, and calculate its Laplace transform with respect to $t$. Let be $d > 0$ and $s \in \mathbb{C}$, then it holds

$$\hat{k}(d,s) = \int_{0}^{\infty} e^{-st}k(d,t)dt = \frac{1}{4\pi d} \int_{0}^{\infty} e^{-st}\delta(d-t)dt \quad (3.21)$$

$$= \frac{1}{4\pi d}, \quad (3.22)$$

which is the fundamental solution of the Helmholtz equation, as we will see in Section 4.5.
3.4 Sobolev Spaces

To solve partial differential equations, one has to prescribe appropriate function spaces for its solution. For the boundary integral equation given by (2.8) in the formulation of the problem (cf. Chapter 2) the natural spaces for the proper formulation of existence, uniqueness and well-posedness results are certain Sobolev spaces. The introduction of the Sobolev Spaces on domains and surfaces can be found, e.g., in [SS04, Chapter 2.3] and [Hac92, Chapter 6]. The subsection about Sobolev Spaces on Banach Spaces is inspired by [Eva98, Section 5.9.2].

3.4.1 Sobolev Spaces of Integer Order on Domains

Let \( \Omega \subset \mathbb{R}^d \) be an open set. The starting point for the theory of Sobolev spaces is the Hilbert space \( L^2(\Omega) := \{ u : \Omega \to \mathbb{C} \mid u \text{ Lebesgue measurable and} \int_{\Omega} |u|^2 \, dx < \infty \} \), (3.23) with its scalar product \((u, v)_{0, \Omega} := (u, v)_{L^2(\Omega)} := \int_{\Omega} u(x) \overline{v(x)} \, dx\). The associated norm is denoted as \( \|u\|_{0, \Omega} \) or \( \|u\|_{L^2(\Omega)} \). As usual two functions are identified with each other if they are equal up to a set of measure zero.

An important role play the spaces of test functions which are used for various definitions and proofs using an appropriate density argument.

**Definition 3.4.1.** Let \( \Omega \subset \mathbb{R}^n \) be an open domain. Define the following sets of functions

\[
C^\infty(\Omega) := \{ u : \Omega \to \mathbb{C} \mid u^{(k)} \text{ exists and is continuous } \forall k \in \mathbb{N} \},
\]

\[
C_0^\infty(\Omega) := \{ u \in C^\infty(\Omega) \mid \text{supp } u \subset \subset \Omega \},
\]

(3.24)

with \( \text{supp } u := \{ x \in \Omega \mid u(x) \neq 0 \} \) and \( \Omega' \subset \subset \Omega :\iff \Omega' \text{ is a compact subset of } \Omega \).

The reason for the importance of these spaces of test functions is the fact that both are dense in \( L^2(\Omega) \) and therefore, any function \( f \in L^2(\Omega) \) can be approximated by infinitely differentiable functions, even with compact support in \( \Omega \).

As functions in \( L^2(\Omega) \) in general are not defined in a pointwise sense, one has to introduce a more general notion of derivative in this space, the weak derivative.

**Definition 3.4.2.** A function \( u \in L^2(\Omega) \) has a \( \alpha \)-th weak derivative \( g := D^\alpha u \in L^2(\Omega), \alpha \in \mathbb{N}^d \), if

\[
(g, v)_{0, \Omega} = (-1)^{|\alpha|}(u, D^\alpha v)_{0, \Omega} \quad \forall v \in C_0^\infty(\Omega).
\]

(3.26)

**Definition 3.4.3.** Let \( \Omega \subset \mathbb{R}^d \) be a domain. For \( k \in \mathbb{N} \) the Sobolev space \( H^k(\Omega) \) is defined by

\[
H^k(\Omega) := \{ u \in L^2(\Omega) \mid D^\alpha u \in L^2(\Omega) \quad \forall |\alpha| \leq k \}.
\]

(3.27)

The Sobolev spaces \( H^k(\Omega) \) are Hilbert spaces. This makes them a powerful framework for the theory of partial differential equations.

**Theorem 3.4.4.** The Sobolev space \( H^k(\Omega) \), for \( k \in \mathbb{N} \), with the scalar product

\[
(u, v)_k := \sum_{|\alpha| \leq k} (D^\alpha u, D^\alpha v)_{0, \Omega} = \sum_{|\alpha| \leq k} \int_{\Omega} D^\alpha u \overline{D^\alpha v} \, dx,
\]

(3.28)

forms a Hilbert space. The associated norm is defined by

\[
\|u\|_k := (u, u)_k^{1/2} = \sqrt{\sum_{|\alpha| \leq k} \|D^\alpha u\|_{L^2(\Omega)}^2},
\]

(3.29)
3.4.2 Sobolev Spaces $H^s(\Omega)$, for $s \in \mathbb{R}_{\geq 0}$

For the definition of traces of Sobolev functions onto boundaries, a generalization of Sobolev spaces to non-integer orders is needed. An elegant way to introduce these generalized Sobolev spaces on the full space $\Omega = \mathbb{R}^n$ is provided by Plancherel’s formula. The necessary results are listed in the next lemma:

**Lemma 3.4.5.**

(i) It holds by virtue of Plancherel’s formula

$$
\|u\|_k = \left\| \sum_{|\alpha| \leq k} |\xi|^{|\alpha|} (\mathcal{F}u)(\xi) \right\|_0 \quad \forall u \in H^k(\mathbb{R}^n).
$$

(ii) By

$$
\|u\|_k^F := \left\| (1 + |\xi|^2)^{\frac{k}{2}} (\mathcal{F}u)(\xi) \right\|_0,
$$

a norm is defined on $H^k(\mathbb{R}^n)$, that is equivalent to $\|\cdot\|_k$.

Let $s \geq 0$ and assume $\Omega = \mathbb{R}^n$. Using Point (ii) in the precedent Lemma 3.4.5 and Theorem 3.2.6 it makes sense to define for all $u, v \in L^2(\mathbb{R}^n)$ the scalar product

$$
(u, v)_s := \int_{\mathbb{R}^n} (1 + |\xi|^2)^{s} (\mathcal{F}u)(\xi) \overline{(\mathcal{F}v)(\xi)} d\xi, \quad \|u\|_s := \left\| (1 + |\xi|^2)^{\frac{s}{2}} (\mathcal{F}u)(\xi) \right\|_{L^2(\mathbb{R}^n)}.
$$

**Definition 3.4.6.** The Sobolev spaces $H^s(\mathbb{R}^n)$ for $s \geq 0$ are the completion of $C_0^\infty(\mathbb{R}^n)$ with respect to the norm $\|\cdot\|_s^F$.

It can be shown that this definition for $s \in \mathbb{N}$ is equivalent to the definition of Sobolev spaces of integer order introduced in Section 3.4.1 using the equivalence of the two norms.

For a general domain $\Omega \subset \mathbb{R}^n$ the method via the Fourier transform cannot be applied verbatim. However it is possible to define $H^s(\Omega)$ for such domains using the so called Sobolev-Slobodeckij norm

**Definition 3.4.7.** Let $\Omega \subset \mathbb{R}^n$ and $s \geq 0$ and $s \notin \mathbb{N}$. There exist $k \in \mathbb{N}$ and $0 < \lambda < 1$, such that $s = k + \lambda$. Define for all $u, v \in C^\infty(\Omega)$

$$
(u, v)_s := \sum_{|\alpha| \leq k} \left( \int_{\Omega} D^\alpha u(x) D^\alpha v(x) dx + \int_{\Omega} \int_{\Omega} \frac{(D^\alpha u(x) - D^\alpha u(y))(D^\alpha v(x) - D^\alpha v(y))}{|x-y|^{n+2\lambda}} dx dy \right),
$$

$$
\|u\|_s := \|u\|_{H^s(\Omega)} := \sqrt{(u, u)_s}.
$$

**Definition 3.4.8.** The Sobolev spaces $H^s(\Omega)$ for $s \geq 0$ are the completion of $\{u \in C^\infty(\Omega) \mid \|u\|_s < \infty\}$ with respect to the norm $\|\cdot\|_s$.

Thus Sobolev spaces are defined for general domains and positive real indices. Some fundamental properties, which are important in the context of partial differential equations, are listed in the next theorem.

**Theorem 3.4.9.** For $t, s \geq 0$, it holds

(i) $D^\alpha u \in H^{s-|\alpha|}(\Omega)$ for $|\alpha| \leq s$ and $u \in H^s(\Omega)$.

(ii) $H^s(\Omega) \subset H^t(\Omega)$ for $s \geq t$, i.e. $H^s(\Omega)$ is continuously embedded into $H^t(\Omega)$ for $s \geq t$. 

3.4. SOBOLEV SPACES

3.4.3 Sobolev Spaces on Surfaces

Recall that the problem given by \([2.8]\), formulated Chapter 2, contains an integral over the boundary \(\Gamma\). Thus, appropriate Sobolev spaces on the hypersurface \(\Gamma\) have to be established. A localization via a partition of unity is employed and the differentiability is specified then on the Euclidean parameter domain. The presented overview over these spaces follows [SS04]. Chapter 2.4, where a detailed introduction is given.

**Definition 3.4.10.** The open ball in \(\mathbb{R}^n\) about \(x = 0\) with radius \(r > 0\) is denoted by \(B_r := B_r(0)\). Moreover let us define the following subsets of \(B_r\)

\[
B_r^+ := \{\xi \in B_r | \xi_n > 0\}, \quad B_r^- := \{\xi \in B_r | \xi_n < 0\}, \quad B_r^0 := \{\xi \in B_r | \xi_n = 0\}. \quad (3.35)
\]

**Definition 3.4.11.** A domain \(\Omega \subset \mathbb{R}^n\) is a Lipschitz domain, i.e. \(\Omega \subset C^{0,1}\), if there exists a finite cover \(\mathcal{U}\) of some cardinality \(N \in \mathbb{N}\) of open subsets \(U_i \subset \mathbb{R}^n\), such that the associated bijective mappings \(\chi_i : \overline{B_1} \to \overline{U_i}\) have the following properties for \(i = 1, \ldots, N\):

(i) \(\chi_i \in C^{1,0}(\overline{B_1}, \overline{U_i})\), \(\chi_i^{-1} \in C^{1,0}(\overline{U_i}, \overline{B_1})\), that means \(\chi_i\) and \(\chi_i^{-1}\) both are Lipschitz.

(ii) \(\chi_i(B_0^0) = U_i \cap \Omega\), \(\chi_i(B_r^0) = U_i \cap \Omega\), \(\chi_i(B_r^+ \cap \Omega) = U_i \cap \Omega\), \(\chi_i(B_r^-) = \Omega \setminus (\mathbb{R}^n \setminus \overline{\Omega})\).

A domain \(\Omega\) is a \(C^k\)-domain for a \(k \in \mathbb{N} \cup \{\infty\}\), if (i) can be replaced by

\[
\chi_i \in C^k(\overline{B_1}, \overline{U_i}), \quad \chi_i^{-1} \in C^k(\overline{U_i}, \overline{B_1}). \quad (3.36)
\]

Assume now that \(\Omega \subset \mathbb{R}^n\) is a bounded Lipschitz domain with compact boundary \(\Gamma := \partial \Omega\). Let \((U_i, \chi_i)_{i \in \{1, \ldots, N\}}\) be as in Definition 3.4.11 and \(\chi_{i,0} := \chi_i|B_0^0\). Furthermore let \(\{\beta_i : \Gamma \to \mathbb{R} \mid i = 1, \ldots, N\}\) be a subordinated partition of unity satisfying

\[
1 = \sum_{i=1}^{N} \beta_i(x) \quad \forall x \in \Gamma, \quad \text{supp} \beta_i \subset U_i \cap \Gamma, \quad \beta_i \circ \chi_{i,0} \in C^{0,1}(\overline{B_0^0}). \quad (3.37)
\]

Therefore a function \(f : \Gamma \to \mathbb{C}\) can be localized for each \(i \in \{1, \ldots, N\}\) by

\[
f_i := f \circ \beta_i : \Gamma \to \mathbb{C}, \quad (3.38)
\]

and satisfies \(\text{supp} f_i \subset U_i \cap \Gamma\). Now the smoothness of a function \(f\) on \(\Gamma\) can be characterized by the smoothness of the localized pull-back

\[
\tilde{f}_i := f_i \circ \chi_{i,0} : B_0^0 \to \mathbb{C}, \quad i \in \{1, \ldots, N\}. \quad (3.39)
\]

Thus any \(f\) cannot be smoother than the mappings \(\chi_i\) corresponding to the domain. For example, if \(\Omega\) is a Lipschitz domain, the maximal smoothness of a function \(f : \Gamma \to \mathbb{R}\) is Lipschitz continuity. The definition of the smoothness can be extended analogously to \(C^k\) domains, where the upper limit for the smoothness of a function is \(k\). Hence for a definition of Sobolev spaces \(H^l(\Gamma)\) on a \(C^{0,1}\), i.e. Lipschitz, resp. \(C^k\) domain holds 0 \(\leq l \leq 1\) resp. \(0 \leq l \leq k\).

**Definition 3.4.12.** Let \(\Omega \subset \mathbb{R}^n\) be a bounded \(C^{0,1}\) resp. \(C^k\) domain with \(k \geq 1\). For \(l \in \mathbb{R}, l \geq 0\) with \(l \leq 1\) resp. \(l \leq k\) the Sobolev Space \(H^l(\Gamma)\) contains all functions \(f : \Gamma \to \mathbb{C}\) such that \(\tilde{f}_i \in H^l_i(B_0^0)\) (cf. footnote) for all \(i = 1, \ldots, N\). The norm is given in analogy to

\[
\text{for a domain } \Omega \subset \mathbb{R}^n, \text{ the spaces } H^l(\Omega) \text{ contain Sobolev functions with zero boundary conditions in the sense of traces (cf. Section 2.4.4) defined by }
\]

**Definition 3.4.13.** Let \(l \in \mathbb{R}_{\geq 0}\). The spaces \(H^l_0(\Omega)\) is given by the completion of \(C^\infty_0(\Omega)\) with respect to the norm \(\|\cdot\|_l\).
CHAPTER 3. MATHEMATICAL FRAMEWORK

(3.33) by
\[
\|f\|_{L^2}^2 :\sum_{|\alpha| \leq [l]} \|f_\alpha\|_{L^2(\Gamma)}^2 + \sum_{|\alpha| \leq [l]} \int_\Gamma \int_\Gamma \|f_\alpha(x) - f_\alpha(y)\|^2 \|x - y\|^\lambda \xi d\Gamma_x d\Gamma_y,
\]
where \( \lambda = l - \lfloor l \rfloor \) and \( f_\alpha : \Gamma \to \mathbb{C} \) with
\[
f_\alpha(x) := \sum_{i=1}^N D^\alpha (\tilde{f}_i)(\xi), \quad \text{with } x = \chi_{i,0}(\xi).
\]

In the case of \( l \in \mathbb{N} \) the second term of the right-hand side of (3.40), i.e. the integral term, is omitted.

It can be shown that the Sobolev space \( H^l(\Gamma) \) does not depend on the choice of \( (U_i, \chi_i)_{i \in \{1, \ldots, N\}} \) for \( l \leq 1 \) in the case of a Lipschitz domain resp. \( l \leq k \) for a \( C^k \) domain, i.e. they and their norms are equivalent.

3.4.4 Traces, Liftings and Extensions

To guarantee an unique solution of a partial differential equation, typically, some boundary conditions have to be imposed. In the problem (2.4), Dirichlet boundary conditions have to be fulfilled. As Sobolev functions are not defined in a classical pointwise way and the values on sets of measure zero are arbitrary, one has to define the restriction of Sobolev functions to a boundary in an appropriate weak way. One has to notice that the boundary is a subset of Lebesgue measure zero.

This question is answered by the following trace theorem for Lipschitz domains which can be found, e.g., in [SS04, Section 2.6].

**Theorem 3.4.14.** Let \( \Omega^i \) be an open, bounded Lipschitz domain with boundary \( \Gamma \) and \( \Omega^e := \mathbb{R}^n \setminus \overline{\Omega^i} \).

(i) For \( \frac{1}{2} < l < \frac{3}{2} \) there exists a continuous linear trace operator \( \gamma_0 : H^l(\mathbb{R}^n) \to H^{l-\frac{1}{2}}(\Gamma) \) with
\[
\gamma_0 u = u|_\Gamma \quad \forall u \in C^0(\mathbb{R}^n) \cap H^l(\mathbb{R}^n).
\]

(ii) For \( \Omega^s \) with \( s \in \{e, i\} \), i.e. the interior and exterior domain, there exists a one-sided continuous linear trace operator \( \gamma_0^s : H^l(\Omega^s) \to H^{l-\frac{1}{2}}(\Gamma) \) with
\[
\gamma_0^s u = u|_\Gamma \quad \forall u \in C^0(\overline{\Omega^s}) \cap H^l(\overline{\Omega^s}) \quad \text{and} \quad \gamma_0^e u = \gamma_0^i u = \gamma_0 u,
\]
almost everywhere for all \( u \in H^l(\mathbb{R}^n) \).

This result can be generalized to \( C^\infty \)-domains and higher order Sobolev spaces.

**Theorem 3.4.15.** Let \( \Omega^i \subset \mathbb{R}^n \) be an open, bounded \( C^k \) domain for \( k \in \mathbb{N} \cup \{\infty\} \) and \( \Omega^e := \mathbb{R}^n \setminus \overline{\Omega^i} \). Let the differentiation index satisfy the condition \( \frac{1}{2} < l < k \). Then the trace operator from Theorem 3.4.14 is a continuous operator \( \gamma_0 : H^l(\mathbb{R}^n) \to H^{l-\frac{1}{2}}(\Gamma) \) with the property
\[
\gamma_0 u = u|_\Gamma \quad \forall u \in C^\infty(\mathbb{R}^n).
\]
Furthermore, it can be shown that the trace of a Sobolev function $u$ only depends on the values of $u$ in a neighborhood of $\Gamma$. Therefore, this result can be extended to functions, which locally belong to Sobolev spaces. In that framework, the question of the lifting of a function defined on the boundary $\Gamma$ onto the whole domain $\Omega^0$ or $\Omega^e$ can be answered. To avoid further technical details, we just refer to the quoted literature.

Another question that already appeared in this thesis is the extension of a Sobolev function $u^e$ on $\Omega^e$ onto the whole $\mathbb{R}^n$. This can be done by lifting the trace of $u^e$ onto $\Omega^i$ resulting in a function $u^i$. The extension then is obtained by the composition of these two functions. A question remaining is the regularity of this extension. These and further details are developed in [Néd01, Section 2.5.2].

### 3.4.5 Dual Spaces of Sobolev Spaces

As usual in the context of functional analysis, the dual spaces of the function spaces play an important role, particularly in the case of Hilbert spaces. Therefore we define the dual space of a Sobolev space with positive real index on surfaces.

**Definition 3.4.16.** Let $s \geq 0$. The dual space of the Sobolev space $H^s(\Gamma)$ is denoted by $H^{-s}(\Gamma) := (H^s(\Gamma))^\prime$. The norm is given by the operator norm

$$
\|v\|_{-s} := \sup_{u \in H^s(\Gamma) \setminus \{0\}} \frac{|v(u)|}{\|u\|_s}. \tag{3.45}
$$

The reason why the dual spaces are denoted as Sobolev spaces with negative, real indices is motivated by the following result.

**Proposition 3.4.17.** The following embeddings are continuous and dense for $s \geq 0$

$$
H^s(\Gamma) \subset L^2(\Gamma) \subset H^{-s}(\Gamma). \tag{3.46}
$$

Therefore the scalar product $(\cdot, \cdot)_{L^2(\Gamma)}$ can be continuously extended to a dual pairing on the product of the Sobolev space and its corresponding dual space. For illustration let us take $u \in H^s(\Gamma)$ and $v \in H^{-s}(\Gamma)$. By virtue of the last result there exists a sequence $\{v_n\}_{n \in \mathbb{N}} \subset L^2(\Gamma)$ with $\lim_{n \to \infty} \|v_n - v\|_{-s} = 0$. Therefore it holds

$$
v(u) = (v, u)_{H^{-s}(\Gamma) \times H^s(\Gamma)} = \lim_{n \to \infty} (v_n, u)_{H^{-s}(\Gamma) \times H^s(\Gamma)} = \lim_{n \to \infty} (v_n, u)_{L^2(\Gamma)} =: (v, u)_{L^2(\Gamma)}. \tag{3.47}
$$

### 3.4.6 Sobolev Spaces for Problems in Time and Space

As the solution $u$ and the density $\varphi$ depend both on space and time, we have to introduce other function spaces. What follows is a generalization of the theory developed in Sections 3.4.1 and 3.4.2 to Banach space valued functions. The whole Lebesgue integration theory can be generalized for such functions, instead of just looking at complex valued functions. The theory for that issue can be found in [AE08, Chapter X]. Furthermore, a short introduction to such Sobolev spaces is given in [Evauloch98, Chapter 5.9.2]. The fundamental difference is that in every case where the absolute value $|f|$ of a complex valued functions appears, it is replaced by the norm of the underlying Banach space $\|f\|$. Using Fourier techniques developed in Chapter 3.4.2, one can introduce Sobolev spaces avoiding weak derivatives with respect to the time, as we will see later. First of all, continuous functions are introduced (cf. [Evauloch98, ibid.]).
Problem, which also will appear in Section 4.7, where the derivation and further details are

We will apply this concept to the single layer potential equation for the exterior Helmholtz

grating that over the appropriate domain.

The variational formulation, is obtained by multiplying the equation with test functions and inte-

trating over the domain. The weak formulation, also called the variational formulation, is ob-

tained by multiplying the equation with test functions and integrating that over the appropriate domain.

As well as for the traditional Sobolev spaces, there are approximation theorems for these time

dependent Sobolev spaces. One of them is [Evans98, Theorem 2 in 5.9.2], where it is shown

that a function \( f \) in \( H^1(0, T; X) \) is in \( C(0, T; X) \), after possibly being redefined on a set with

measure zero. That motivates the choice of the function space in Theorem 4.6.2

\[ (3.48) \]

Instead of developing now a general theory about the corresponding Sobolev spaces, we just

introduce the spaces used for solving the problem \( P \) given by (2.4). For the appropriate

Sobolev space the definitions can be found in [HKS09]. For a general definition, the space

\( H^s(\Gamma) \) has to be replaced by a general Banach space \( Y \).

Definition 3.4.18. For \( r \in \mathbb{R} \) and \( s \in [-k, k] \), where \( k \) denotes the smoothness of the surface

\( \Gamma \) and \( k = 1 \) for a Lipschitz domain, the anisotropic Sobolev space \( H^r(\mathbb{R}, H^s(\Gamma)) \) is given by

\[ H^r(\mathbb{R}; H^s(\Gamma)) := \left\{ g : \Gamma \times \mathbb{R} \to \mathbb{R} \mid \| g \|_{H^r(\mathbb{R}, H^s(\Gamma))} < \infty \right\}, \tag{3.49} \]

with the norm

\[ \| g \|_{H^r(\mathbb{R}, H^s(\Gamma))}^2 := \int_\mathbb{R} (1 + |\omega|^2)^r |\mathcal{F}g(\cdot, \omega)|^2 \| \mathcal{F}g(\cdot, \omega) \|_{H^s(\Gamma)}^2 d\omega. \tag{3.50} \]

The definition of the norm is redolent to the definition of the Sobolev norms using Fourier

techniques in Chapter 3.4.2. For solving equation (2.4) we need functions \( g \) just for \( t \in [0, T] \).

Therefore we are looking at functions in the above defined space vanishing for negative times.

That gives us then already a certain regularity of the function at \( t = 0 \), to which will be

referred, that \( g \) is smooth enough and compatible.

Definition 3.4.19. For \( r \in \mathbb{R} \) and \( s \in [-k, k] \), where \( k \) denotes the smoothness of the surface

\( \Gamma \) and \( k = 1 \) for a Lipschitz domain, the anisotropic Sobolev space \( H^r(\mathbb{R}, H^s(\Gamma)) \) is given by

\[ H^r(\mathbb{R}; H^s(\Gamma)) := \left\{ g : \Gamma \times \mathbb{R} \to \mathbb{R} \mid \| g \|_{H^r(\mathbb{R}, H^s(\Gamma))} < \infty \right\}, \tag{3.49} \]

with the norm

\[ \| g \|_{H^r(\mathbb{R}, H^s(\Gamma))}^2 := \int_\mathbb{R} (1 + |\omega|^2)^r |\mathcal{F}g(\cdot, \omega)|^2 \| \mathcal{F}g(\cdot, \omega) \|_{H^s(\Gamma)}^2 d\omega. \tag{3.50} \]

The definition of the norm is redolent to the definition of the Sobolev norms using Fourier

techniques in Chapter 3.4.2. For solving equation (2.4) we need functions \( g \) just for \( t \in [0, T] \).

Therefore we are looking at functions in the above defined space vanishing for negative times.

That gives us then already a certain regularity of the function at \( t = 0 \), to which will be

referred, that \( g \) is smooth enough and compatible.

Definition 3.4.19. For \( r \in \mathbb{R} \) and \( s \in [-k, k] \), where \( k \) denotes the smoothness of the surface

\( \Gamma \) and \( k = 1 \) for a Lipschitz domain, the anisotropic Sobolev space \( H^r(\mathbb{R}, H^s(\Gamma)) \) is given by

\[ H^r(\mathbb{R}; H^s(\Gamma)) := \left\{ g : \Gamma \times \mathbb{R} \to \mathbb{R} \mid \| g \|_{H^r(\mathbb{R}, H^s(\Gamma))} < \infty \right\}, \tag{3.49} \]

with the norm

\[ \| g \|_{H^r(\mathbb{R}, H^s(\Gamma))}^2 := \int_\mathbb{R} (1 + |\omega|^2)^r |\mathcal{F}g(\cdot, \omega)|^2 \| \mathcal{F}g(\cdot, \omega) \|_{H^s(\Gamma)}^2 d\omega. \tag{3.50} \]

The definition of the norm is redolent to the definition of the Sobolev norms using Fourier

techniques in Chapter 3.4.2. For solving equation (2.4) we need functions \( g \) just for \( t \in [0, T] \).

Therefore we are looking at functions in the above defined space vanishing for negative times.

That gives us then already a certain regularity of the function at \( t = 0 \), to which will be

referred, that \( g \) is smooth enough and compatible.

Definition 3.4.20. The space \( H^r(0, T; H^s(\Gamma)) \) is defined by

\[ H^r_0(0, T; H^s(\Gamma)) := \{ g : \Gamma \times [0, T] \to \mathbb{R} \mid g = g^* \mid_{[0, T]} \text{ for some } g^* \in H^r(\mathbb{R}, H^s(\Gamma)) \}

\text{with } g^* \equiv 0 \text{ on } (-\infty, 0) \}, \tag{3.51} \]

and the norm on that space is given by

\[ \| g \|_{H^r_0(0, T; H^s(\Gamma))} := \min \{ \| g^* \|_{H^r(\mathbb{R}, H^s(\Gamma))} \mid g^* \in H^r(\mathbb{R}, H^s(\Gamma)) \}

\text{with } g = g^* \mid_{[0, T]} \text{ and } g^* \equiv 0 \text{ on } (-\infty, 0) \} \tag{3.52} \]

As well as for the traditional Sobolev spaces, there are approximation theorems for these time

dependent Sobolev spaces. One of them is [Evans98, Theorem 2 in 5.9.2], where it is shown

that a function \( f \) in \( H^1(0, T; X) \) is in \( C(0, T; X) \), after possibly being redefined on a set with

measure zero. That motivates the choice of the function space in Theorem 4.6.2

3.5 Weak Formulation of Equations

As the classical formulation of partial differential equations in the space of regular functions,

i.e. \( f \in C^k \), typically does not lead to sharp existence and uniqueness results, a variational

formulation is employed. The spaces introduced in the last sections provide a Hilbert space

frame to solve the wave equation in its weak formulation. The weak formulation, also called

variational formulation, is obtained by multiplying the equation with test functions and inte-

grating that over the appropriate domain.

We will apply this concept to the single layer potential equation for the exterior Helmholtz

problem, which also will appear in Section 4.7 where the derivation and further details are
listed. For densities \( \varphi \) which are regular enough, e.g. \( \varphi \in L^\infty(\Gamma) \), the single layer potential for the Helmholtz equation is given by the following representation\(^2\)

\[
\mathcal{V}(s) \varphi(x) := \int_{\Gamma} e^{-s \|x-y\|} \varphi(y) d\Gamma_y,
\]

(3.53)

The framework for the spaces of functions is given by the spaces \( H^{1/2}(\Gamma) \subset L^2(\Gamma) \subset H^{-1/2}(\Gamma) \) with dense embeddings. The problem to solve is the integral equation

\[
\mathcal{V}(s) \varphi = g,
\]

(3.54)

where \( \mathcal{V}(s) : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma) \) is the single layer potential for a given \( s \in \mathbb{C} \), \( g \in H^{1/2}(\Gamma) \) the given Dirichlet data and \( \varphi \in H^{-1/2}(\Gamma) \) the searched density. This equation is not considered pointwise. However one multiplies the equation by test functions, \( \eta \in H^{-1/2}(\Gamma) \), and integrates over \( \Gamma \). That leads to

\[
\int_{\Gamma} (\mathcal{V}(s) \varphi) \cdot \eta d\Gamma_x = \int_{\Gamma} g \cdot \eta d\Gamma_x \quad \forall \eta \in H^{-1/2}(\Gamma).
\]

(3.55)

This can be rewritten using the continuous extension of the inner product of \( L^2(\Gamma) \), as discussed in Section 3.4.5. That leads to the variational or weak form of the problem: Find \( \varphi \in H^{-1/2}(\Gamma) \) such that

\[
(\mathcal{V}(s) \varphi, \eta)_{L^2(\Gamma)} = (g, \eta)_{L^2(\Gamma)} \quad \forall \eta \in H^{-1/2}(\Gamma).
\]

(3.56)

A solution \( \varphi \) of that problem is called weak solution. If \( \varphi \) is regular enough, it can be shown, that \( \varphi \) is a solution in the classical sense. This weak form of the equation allows now to show uniqueness and existence of the solution via the Lemma of Lax-Milgram. The left-hand side of the weak formulation [3.56] defines a sesquilinear form \( b \) on \( H^{-1/2}(\Gamma) \times H^{-1/2}(\Gamma) \) by

\[
b(\varphi, \eta) := (\mathcal{V}(s) \varphi, \eta)_{L^2(\Gamma)},
\]

(3.57)

and the right-hand side of (3.56) a linear functional \( F \) on \( H^{-1/2}(\Gamma) \)

\[
F(\eta) := (g, \eta)_{L^2(\Gamma)}.
\]

(3.58)

**Lemma 3.5.1.** Let \( H \) be a Hilbert space and \( b : H \times H \to \mathbb{C} \) a continuous sesquilinear form. If \( b \) is \( H \)-elliptic, i.e.

\[
\exists \gamma > 0 \text{ and } \sigma \in \mathbb{C} \text{ with } |\sigma| = 1 \text{ such that } \text{Re} (\sigma b(u, u)) \geq \gamma \|u\|_H^2 \quad \forall u \in H,
\]

(3.59)

the variational problem

\[
b(u, v) = l(v) \quad \forall v \in H,
\]

(3.60)

has a unique solution \( u \in H \) for each \( l \in H' \) which satisfies

\[
\|u\|_H \leq \frac{1}{\gamma} \|l\|_{H'}.
\]

(3.61)

\(^2\)The representation of the single layer potential \( \mathcal{V}(s) \) for Helmholtz equation presented in (3.53) is just valid for \( \varphi \) regular enough. The general definition is based on the Newton potential \( \mathcal{N}(s) \) and the trace operator \( \gamma_0 \) resp. its adjoint operator \( \gamma_0' \). The single layer potential on \( \mathbb{R}^n \setminus \Gamma \) is given by \( S(s) := \mathcal{N}(s) \gamma_0' \) and its restriction on \( \Gamma \) by \( \mathcal{V}(s) := \gamma_0 \mathcal{N}(s) \gamma_0' \). Any details are omitted to the corresponding literature, e.g., [SS04, Section 3.1.1]. Using the mapping properties of the involved operators, the domain of definition of \( \mathcal{V}(s) \) can be extended to \( H^{-1/2}(\Gamma) \).
A detailed theory for elliptic problems can be found in [SS04].

Strongly linked to the variational formulation is the Galerkin method. The main principle is to approximate the space $H$, where the solution is searched, by finite dimensional subspaces $S \subset H$ of increasing dimension. This method is used in analysis for existence proofs as well as in numerical analysis for the discretisation of variational problems. As the exact solution typically does not belong to these subspaces, the test functions as well have to be taken from that finite dimensional subspace. On that way once more the Lemma of Lax-Milgram can be applied, as the weak formulation restricted to $S \subset H$ still fulfills the necessary conditions. Furthermore a quasi-optimality of the discretised solution as well as an orthogonality property of the error holds. These results can be found in [SS04, Section 4.2]. This theory can be extended to more general variational problems, including compact perturbations and numerical approximations.

### 3.6 Derivation of the Convolution Quadrature

As we will see in Equation (4.68) the single layer potential of the wave equation can be written as a convolution in time. In [Lub88a] and [Lub88b] a scheme for the discretisation for a convolution integral of the form using constant step size is developed for $f$ and $g$ complex valued functions. For a given time step $\Delta t$, let us define the equidistant, discrete points of time $t_n := n \cdot \Delta t$ for $n \in \mathbb{N}$. The goal is to obtain a discrete convolution $(f * \Delta t g)(t_n)$ at these equidistant $t_n$ of the form

$$
(f * g)(t_n) \approx (f * \Delta t g)(t_n) = \sum_{j=0}^{n} \omega_{n-j} \cdot g(t_j)
$$

with coefficients $\omega_j \in \mathbb{C}$ depending on $f$, for $j \in \{0, \ldots, n\}$. The derivation will be done for such complex valued functions. However it will be applied for Banach space valued functions, to which this approach can be extended.

To deduce a discrete convolution as proposed in (3.63) we use the Laplace transform $\hat{f}$ of $f$, an approach that will be also seen in Section 4.7. To avoid any technical details in this derivation, we do it formally. For more details we refer to the literature, i.e. [Lub88a] and [Lub88b].

We assume that $\hat{f}$ is analytic and there exist $\sigma \in \mathbb{R}, c \geq 0$ and $\mu > 0$, such that the inverse Laplace transform of $\hat{f}$ exists and

$$
|\hat{f}(s)| \leq c \cdot |s|^{-\mu} \quad \forall s \in \mathbb{C} \text{ with } \text{Re}(s) > \sigma.
$$

(3.64)

Let us insert the inverse Laplace transform into the convolution

$$
(f * g)(t) = \int_{0}^{t} f(t - \tau)g(\tau)d\tau = \int_{0}^{t} \left( \frac{1}{2\pi i} \int_{\sigma+i\mathbb{R}} \hat{f}(s)e^{s(t-\tau)}ds \right)g(\tau)d\tau
$$

(3.65)

$$
= \frac{1}{2\pi i} \int_{\sigma+i\mathbb{R}} \hat{f}(s) \int_{0}^{t} e^{s(t-\tau)}g(\tau)d\tau ds.
$$

(3.66)

A look at the inner integral $y_g(s,t) := \int_{0}^{t} e^{s(t-\tau)}g(\tau)d\tau$ and its time derivative shows

$$
\frac{\partial}{\partial t} y_g(s,t) = \int_{0}^{t} \frac{\partial}{\partial \tau} \left( e^{s(t-\tau)}g(\tau) \right)d\tau + e^{s(t-\tau)}g(\tau)|_{\tau=t} = s \int_{0}^{t} e^{s(t-\tau)}g(\tau)d\tau + g(t)
$$

(3.67)

$$
= s \cdot y_g(s,t) + g(t).
$$

(3.68)
3.6. DERIVATION OF THE CONVOLUTION QUADRATURE

Therefore, \( y_g(s, \cdot) \) satisfies the Cauchy problem
\[
\begin{align*}
\partial_t y_g(s, t) &= s \cdot y_g(s, t) + g(t), \quad t \geq 0 \\
y_g(s, 0) &= 0.
\end{align*}
\] (3.69a)

Thus, \( y_g(s, \cdot) \) can be approximated by a linear multistep method of order \( k \in N \) defined by
\[
\sum_{j=0}^{k} \alpha_j y_{n+j-k}(s) = \Delta t \sum_{j=0}^{k} \beta_j (s \cdot y_{n+j-k}(s) + g((n + j - k)\Delta t)),
\] (3.70)

with coefficients \( \alpha_j, \beta_j \in \mathbb{R} \), for \( 0 \leq j \leq k \), starting values \( y_{-k}(s) = \cdots = y_{-1}(s) = 0 \) and w.l.o.g. \( \alpha_k = 1 \). Thus, in this approach, the approximation \( y_g(t_n, s) \approx y_n(s) \) is obtained by a linear multistep method. Inserting that into (3.66) leads to
\[
(f \cdot g)(t_n) \approx \frac{1}{2\pi i} \int_{\sigma + i\mathbb{R}} \hat{f}(s) y_n(s) ds =: (f \cdot g)(t_n).
\] (3.71)

The next step is to transform (3.71) into a discrete convolution. Let us first define the ratio of the generating polynomials of the underlying multistep method
\[
\gamma(\zeta) := \frac{\sum_{j=0}^{k} \alpha_j \zeta^{-j}}{\sum_{j=0}^{k} \beta_j \zeta^{-j}}.
\] (3.72)

Furthermore we take both sides of (3.70) as the \( n \)-th coefficient of a formal power series resulting in
\[
\sum_{n=0}^{\infty} \left( \sum_{j=0}^{k} \alpha_j y_{n+j-k}(s) \right) \zeta^n = \sum_{n=0}^{\infty} \left( \sum_{j=0}^{k} \beta_j (s \cdot y_{n+j-k}(s) + g((n + j - k)\Delta t)) \right) \Delta t \cdot \zeta^n.
\] (3.73)

Using the Cauchy product formula on both sides and applying various renumbering leads to
\[
\left( \sum_{j=0}^{k} \alpha_j \zeta^{-j} \right) \cdot \left( \sum_{j=0}^{\infty} y_j(s) \zeta^j \right) = \left( \sum_{j=0}^{k} \beta_j \zeta^{-j} \right) \cdot \left( \sum_{j=0}^{\infty} (s y_j(s) + g(j \cdot \Delta t)) \Delta t \zeta^j \right).
\] (3.74)

Finally a representation of \( \sum_{j=0}^{\infty} y_j(s) \zeta^j \) results after a few steps of basic transformations
\[
\sum_{j=0}^{\infty} y_j(s) \zeta^j = \left( \frac{\gamma(\zeta)}{\Delta t} - s \right)^{-1} \cdot \sum_{j=0}^{\infty} g(j \Delta t) \zeta^j.
\] (3.75)

As next step we also turn (3.71) into a power series and insert the result of (3.75)
\[
\sum_{n=0}^{\infty} (f \cdot g)(t_n) \zeta^n = \sum_{n=0}^{\infty} \left( \frac{1}{2\pi i} \int_{\sigma + i\mathbb{R}} \hat{f}(s) y_n(s) ds \right) \zeta^n
\] (3.76)
\[
= \frac{1}{2\pi i} \int_{\sigma + i\mathbb{R}} \frac{\hat{f}(s)}{\gamma(\zeta)\Delta t - s} ds \cdot \left( \sum_{n=0}^{\infty} g(n \Delta t) \zeta^n \right).
\] (3.77)

Employing the Cauchy integral formula on the last integral results in
\[
\frac{1}{2\pi i} \int_{\sigma + i\mathbb{R}} \frac{\hat{f}(s)}{\gamma(\zeta)\Delta t - s} ds = \hat{f} \left( \frac{\gamma(\zeta)}{\Delta t} \right),
\] (3.78)
thus
\[ \sum_{n=0}^{\infty} (f * \Delta t g)(t_n) \zeta^n \]  
Let the $\omega_n^{\Delta t}$ denote the coefficients of the power series of $\hat{f} \left( \frac{\gamma(\zeta)}{\Delta t} \right)$.
That leads to a discrete convolution in the form of (3.63). Using (3.79) and the Cauchy product rule leads to
\[ \sum_{n=0}^{\infty} (f * \Delta t g)(t_n) \zeta^n = \hat{f} \left( \frac{\gamma(\zeta)}{\Delta t} \right) \cdot \left( \sum_{n=0}^{\infty} g(n\Delta t) \zeta^n \right) = \left( \sum_{n=0}^{\infty} \omega_n^{\Delta t} \zeta^n \right) \cdot \left( \sum_{n=0}^{\infty} g(n\Delta t) \zeta^n \right) \]
\[ = \sum_{n=0}^{\infty} \left( \sum_{j=0}^{n} \omega_{n-j}^{\Delta t} \cdot g(j\Delta t) \right) \zeta^n. \]  
In conclusion, that leads, by comparing the coefficients, to
\[ (f * \Delta t g)(t_n) = \sum_{j=0}^{n} \omega_{n-j}^{\Delta t} \cdot g(j\Delta t). \]  
Let us resume the most important points and results of this derivation:
(i) The replacement of $f$ by its Laplace transform $\hat{f}$ brings up a function fulfilling an ordinary differential equation, which is approximated by a linear multistep method.
(ii) The convolution weights $\omega_n^{\Delta t}$ are given implicitly by its definition
\[ \hat{f} \left( \frac{\gamma(\zeta)}{\Delta t} \right) = \sum_{n=0}^{\infty} \omega_n^{\Delta t} \zeta^n. \]  
By calculating this power series the convolution weights are obtained. They depend on the linear multistep method via $\gamma$, on the time step $\Delta t$ and on $\hat{f}$. It is important to note that the discrete convolution does not depend explicitly on $f$ anymore but only on its Laplace transform. Especially for distributional $f$, as for example the fundamental solution of the wave equation, this is an advantage, as $\hat{f}$ is still a classical function.
To finish this derivation of the convolution quadrature applied in this thesis, the main result for convergence is presented (cf. [Lub88a, Theorem 3.1]).

Theorem 3.6.1. Let $\hat{f}(s)$, the Laplace transform of $f$, be analytic in the half-plane $\text{Re}(s) > \sigma$ with $\sigma \in \mathbb{R}$ and satisfy there
\[ \exists c < \infty, \mu > 0 \text{ such that } |\hat{f}(s)| \leq c \cdot |s|^{-\mu}. \]  
Let $\gamma$ be the ratio of the generating polynomials of a linear multistep method, which is $A$-stable and of order $p$, with $\gamma$ analytic and without any zeros in a neighbourhood of the closed unit disc with the exception of a zero at $\zeta = 1$. Then it holds
\[ |(f * \Delta t g)(t) - (f * g)(t)| \leq C \cdot T^{\mu-1} \cdot \left( |\Delta t| g(0) | + \cdots + \Delta t^{p-1} |g^{(p-2)}(0)| + \Delta t^p \cdot |g^{(p-1)}(0)| \right) \]
\[ + C \cdot T^\mu \cdot \Delta t^p \cdot \max_{0 \leq s \leq t} |g^{(p)}(s)|, \]  
where $C$ is a constant not depending on $\Delta t \in (0, \Delta t_*)$, $t \in [\Delta t, T]$ with fixed $T < \infty$ and $g \in C^p([0, T])$. 

Therefore the convergence rate of the convolution quadrature equals the order of the underlying multistep method if \( g(0), g'(0), \ldots, g^{(p-1)}(0) \) vanish.

From this result a link can be established to the spaces defined in Section 3.4.6 more exactly in the Definition 3.4.20. As we postulate that the functions vanish for negative times and have certain regularity dependent of the choice of \( r \), one can imagine that \( g(0), g'(0), \ldots, g^{(p-1)}(0) \) vanish in some sense for \( r \) large enough.

### 3.7 Derivation of BDF2

The convolution quadrature presented in Section 3.6 is based on a linear multistep method. To derive such a linear multistep method let us consider an ordinary differential equation for a function \( y : [0, T] \to \mathbb{R} \)

\[
y'(t) = f(t, y(t)) \quad \text{with} \quad y(0) = y_0.
\]  

(3.86)

Discretise the interval \([0, T]\) into \( N \in \mathbb{N}_{>1} \) subintervals with equidistant, discrete times \( t_n = n \cdot \Delta t \), where \( \Delta t = \frac{T}{N} \) and \( n = 0, \ldots, N \). Let us denote the approximation of \( y \) at the discrete times by \( y(t_n) \approx y_n \). Assume that for \( n \geq 2 \) the approximation at \( y_{n-1} \) and \( y_{n-2} \) is known.

We want to construct a linear multistep method of order 2. Therefore we approximate \( y \) locally by a quadratic polynomial \( p(t) = at^2 + bt + c \), with \( a, b, c \in \mathbb{R} \). As conditions we impose

(i) \( p((n-2)\Delta t) = y_{n-2} \)

(ii) \( p((n-1)\Delta t) = y_{n-1} \)

(iii) \( p'(n\Delta t) = f(n\Delta t, y_n) \)

Using the first two conditions, it can be calculated that

\[
a = \frac{y_{n-2} - 2y_{n-1} + y_n}{2(\Delta t)^2} \quad \text{and} \quad b = \frac{y_{n-2} - 4y_{n-1} + 3y_n - 2n(y_{n-2} - 2y_{n-1} + y_n)}{2\Delta t}.
\]  

(3.87)

Inserting that into the third condition leads to the implicit equation

\[
\frac{1}{2}y_{n-2} - 2y_{n-1} + \frac{3}{2}y_n = \Delta t \cdot f(n\Delta t, y_n),
\]  

(3.88)

defining the Backward Differentiation Formula of order 2 (BDF2). To bring it into the form of (3.70), we have to take

\[
k = 2, \quad \alpha_0 = \frac{1}{2}, \quad \alpha_1 = -2, \quad \alpha_2 = \frac{3}{2}, \quad \beta_0 = \beta_1 = 0, \quad \beta_2 = 1.
\]  

(3.89)
Hence the ratio of the generating polynomials is given by
\[
\gamma(\zeta) = \frac{1}{2}\zeta^2 - 2\zeta + \frac{3}{2}
\]  
(3.90)

**Proposition 3.7.1.** BDF2 is an A-stable linear multistep method of order 2. Furthermore, \(\gamma\) has no poles on the unit circle, with the exception of \(\zeta = 1\).

For a proof we refer to [HW91, Chapter 5]. The condition for \(\gamma\) is easy to check, as \(\gamma\) is a polynomial. Furthermore the order of an A-stable linear multistep method cannot exceed the order 2, as it is shown in [ibid.]. Therefore BDF2 is an optimal choice for a linear multistep method fulfilling Proposition 3.7.1.

### 3.8 Spherical Harmonics

To check at the end of the project that the method is implemented correctly, test cases are needed to compare the theoretical asymptotic convergence rates with the results of the implementation. The best test case would be that the density \(\varphi\) is known analytically. It is natural to search for eigenfunctions of the involved operator, in order to construct such a known solution. In our example this is done for the Laplace operator, in the case where \(\Gamma\) is the unit sphere in \(\mathbb{R}^3\). Further details can be found in [Néd01, Section 2.4].

Therefore we examine the so-called spherical harmonics. The spherical harmonics are orthonormal eigenfunctions of the angle-part of the Laplace-Operator, more exactly: \(Y_l^m\) is the eigenfunction to the eigenvalue \(-l(l-1)\), for \(l \in \mathbb{N}\) and \(m \in \mathbb{Z}\) with \(-l \leq m \leq l\). They can be formulated in spherical coordinates as
\[
Y_l^m(\theta, \varphi) = \frac{1}{\sqrt{2\pi N_l^m}} N_l^m e^{im\varphi} P_l^m(\cos\theta),
\]  
(3.91)

with \(P_l^m\) the associated Legendre-polynomials
\[
P_l^m(x) := \frac{(-1)^m}{2l!} (1 - x^2)^{m/2} \frac{d^{l+m}}{dx^{l+m}}(x^2 - 1)^l,
\]  
(3.92)

and the scaling constants
\[
N_l^m := \sqrt{\frac{2l + 1}{2} \frac{(l - m)!}{(l + m)!}}.
\]  
(3.93)

As a test case defined in Chapter 8 we will encounter \(Y_1^1\), which is plotted in Figure 3.2.

![Figure 3.2: The real part (left) and the imaginary part (right) of \(Y_1^1\) on the unit sphere as a mesh of 9062 triangles](image-url)
Chapter 4

Theory for the Exact Problem

The wave equation (2.4) encountered in the formulation of the problem (cf. Chapter 2) is called the scalar acoustic wave equation. Before starting to investigate the existence and uniqueness of solutions, some fundamental properties of that equation are listed in the next section. This includes the physical background as well as some mathematical properties.

4.1 Properties of the Wave Equation

The scalar acoustic equation describes the propagation of an acoustic wave in a homogeneous medium. The wave is described by the difference of the pressure \( u \) with respect to the former resting state, where the pressure was in an equilibrium. Different media have different speed of sound \( c > 0 \). The scalar acoustic equation in its general form depends on the speed of sound \( c \) and is given by

\[
\frac{1}{c^2} u_{tt} - \Delta u = 0. \tag{4.1}
\]

However, the function \( \tilde{u}(x, t) := u(x, c \cdot t) \) solves the scalar acoustic equation \( \tilde{u}_{tt} - \Delta \tilde{u} = 0 \). Therefore, throughout this paper it is supposed that \( c = 1 \), as by a scaling of variable solutions for other arising speeds of sound can be obtained. The finite velocity of the wave is a fundamental property assigned to solutions of the wave equation. If \( u \) at a certain time is just non-zero inside a bounded domain, i.e. \( u \) has compact support, then \( u \) will have compact support at any time, as every disturbance propagates just with the finite velocity \( c \).

Furthermore, as we solve the wave equation in the three dimensional space \( \mathbb{R}^3 \), the waves are propagating according to the Huygens-Fresnel principle. This states that every point of a wave at a certain time is the origin of a new spherical wave, which will propagate starting at this time. A whole wave then is the superposition of all these spherical waves started at a certain point. In this context the notion of retarded time arises. Let us fix a point \( x \in \mathbb{R}^3 \) and a time \( t > 0 \). The goal is now to determine all possible origins and starting times of spherical waves that reach \( x \) at the time \( t \). Assuming \( y \in \mathbb{R}^3 \) is another point, then a spherical wave has an influence on the pressure at the point \( x \) and time \( t \) if and only if it has started at \( y \) at the time \( \tau = t - \frac{\|x-y\|}{c} \), as exactly in that time, the spherical wave with velocity \( c \) has reached \( x \). Furthermore the disturbance at \( x \) has to be smaller than the disturbance at the starting time in \( y \), as the energy

\[
E(t) = \int_{\Omega} |u_t(x, t)|^2 + \|\nabla u(x, t)\|^2 \, dx, \tag{4.2}
\]

is constant for the original problem \( P_{\text{orig}} \), supposing finite energy at the starting point \( t = 0 \). This follows from \( \frac{dE}{dt} = 0 \), which can be shown by using Green’s formulas, the wave equation and the imposed Dirichlet condition. This physical considerations are recognized in the form of the fundamental solution \( k \) (cf. (2.6)).
4.2 Existence and Uniqueness for the full space Problem

During the process of reducing our original problem $P_{\text{orig}}$ to a homogeneous equation the full space problem $P_{\text{inc}}$ arises as presented in Chapter 2. It is the Cauchy problem for the wave equation in the form of

\begin{align}
\partial_t^2 u(x, t) - \Delta u(x, t) &= f(x, t) \quad \text{in } \mathbb{R}^3 \times (0, T) \tag{4.3a} \\
u(x, 0) &= u_0(x) \quad \forall x \in \mathbb{R}^3 \tag{4.3b} \\
\partial_t u(x, 0) &= u_1(x) \quad \forall x \in \mathbb{R}^3 \tag{4.3c}
\end{align}

The theory to solve this problem is developed in [Wla72] pp. 161-165], where the classical as well as the weak solution can be found. We mention here just the classical solution in the three dimensional case, as it is well illustrating the influence of the given data on the solution $u$.

**Theorem 4.2.1.** Assume $f \in C^2(\mathbb{R}^3 \times [0, \infty))$, $u_0 \in C^3(\mathbb{R}^3)$, $u_1 \in C^2(\mathbb{R}^3)$. Then the classical solution of the Cauchy problem of the wave equation exists, is unique and is described by the Kirchhoff Formula

\[
u(x, t) = \int_{B_1(x)} \frac{f(y, t - \|x - y\|)}{4\pi \|x - y\|} dy + \frac{1}{4\pi} \int_{\partial B_1(x)} u_1(y) dS_y + \frac{1}{4\pi} \frac{\partial}{\partial t} \left( \frac{1}{t} \int_{\partial B_1(x)} u_0(y) dS_y \right) \tag{4.4}
\]

Examining this result in more details would show that this formula matches the physical considerations done in Section 4.1. Hence, to prove the existence and uniqueness of the solution for the original problem $P_{\text{orig}}$, the homogeneous exterior problem $P_{\text{inc}}$ still has to be solved. The existence and uniqueness of such a solution is stated in Section 4.7 using a single layer potential representation.

4.3 Motivation for the Single Layer Potential

As noted the Chapter 2 we want to find a solution $u$ in the form of a single layer potential

\[
u(x, t) = \int_0^t \int_{\Gamma} k(x - y, t - \tau) \varphi(y, \tau) d\Gamma_y d\tau \quad \forall (x, t) \in \Omega \times [0, T]. \tag{4.5}
\]

In [HD03] Section 2.1 a result of classical electromagnetical scattering theory is presented. Let $u$ be a solution of the homogeneous wave equation with arbitrary boundary conditions and vanishing initial conditions on $\mathbb{R}^3 \setminus \Gamma$, i.e. $u_0$ resp. $u_1$ solves the problem on $\Omega^e \times (0, T)$ resp. $\Omega^i \times (0, T)$ and $u|_{\Omega^e} = u_0$ resp. $u|_{\Omega^i} = u_1$. Let us define the jump of $u$ and its normal derivative

\[
\gamma_0 u(x, t) := u_0(x, t) - u_1(x, t), \quad \gamma_1 u(x, t) := \frac{\partial u_0}{\partial \nu}(x, t) - \frac{\partial u_1}{\partial \nu}(x, t) \quad (x, t) \in \Gamma \times (0, T), \tag{4.6}
\]

where $\nu$ is the outer normal vector on $\Gamma$. Then the solution $u$ has the following integral representation

\[
u(x, t) = \int_{\Gamma} \frac{\nu(y) \cdot (x - y)}{4\pi \|x - y\|^2} \left( \frac{\gamma_0 u(y, s)}{\|x - y\|^2} + \frac{\partial \gamma_0 u(y, s)}{\partial \nu} \|x - y\| \right) d\Gamma_y \\
- \int_{\Gamma} \frac{\gamma_1 u(y, s)}{4\pi \|x - y\|^2} d\Gamma_y, \tag{4.7}
\]
with the retarded time \( s = t - \|x - y\| \). Under the assumption that we are solving a Dirichlet problem with the same boundary condition for the interior and exterior problem. If a continuous solution exists for both problems the jump of the function \( u \) over the boundary vanishes, i.e., \( [\gamma_0 u] = 0 \). Hence, the first integral term in the representation (4.7) disappears and it holds

\[
u(x, t) = \int_{\Gamma} \frac{-[\gamma_1 u](y, s)}{4\pi \|x - y\|} \ d\Gamma_y.
\]

(4.8)

Thus, it makes sense to apply a single layer potential in order to solve a Dirichlet problem.

### 4.4 Connection to the Helmholtz Equation

As a first try to solve the wave equation, one could try the separation of time and space variable, i.e. \( u(x, t) = X(x)T(t) \). Furthermore, assume that \( u \) is harmonic in time. Therefore, \( T \) is of the form \( T(t) = e^{\kappa t} \) for a real \( \kappa \). Applying the wave equation to such a \( u \) leads to

\[
(\partial_t^2 - \Delta)u(x, t) = -\kappa^2 X(x)T(t) - \Delta X(x)T(t) = 0.
\]

(4.9)

As \( T \) is known one gets an equation for \( X \) by dividing formally by \( T(t) \)

\[-\kappa^2 X(x) - \Delta X(x) = 0.
\]

(4.10)

This is the homogenous Helmholz equation.

For a general solution \( u \) of the wave equation given by (2.4) the Laplace transform is applied. Therefore, let \( u \) be a solution of the exterior Dirichlet problem \( P \) and assume that \( u \) is Laplace transformable for \( \text{Re}(s) > \sigma \) with respect of \( t \) for a \( \sigma > 0 \). The aim is to derive a partial differential equation, that the Laplace transform \( \hat{u} \) satisfies. Hence, let us apply the Laplace transform on (2.4a). For \( s \in \mathbb{C} \) with \( \text{Re}(s) > \sigma \) it holds

\[
\int_0^\infty e^{-st}(u_{tt}(x, t) - \Delta u(x, t)) dt = \int_0^\infty e^{-st}u_{tt}(x, t) dt - \Delta \int_0^\infty e^{-st}u(x, t) dt.
\]

(4.11)

The first term has to be considered furthermore. Using partial integration leads to

\[
\int_0^\infty e^{-st}u_{tt}(x, t) dt = u_t(x, 0) \cdot e^{-st}|_0^\infty + s \int_0^\infty e^{-st}u_t(x, t) dt = u_t(x, 0) + s \int_0^\infty e^{-st}u_t(x, t) dt
\]

(4.12)

\[
= u_t(x, 0) + s(u(x, t) \cdot e^{-st})|_0^\infty + s^2 \int_0^\infty e^{-st}u(x, t) dt
\]

(4.13)

\[
= u_t(x, 0) + su(x, 0) + s^2 \hat{u}(x, s) = s^2 \hat{u}(x, s),
\]

(4.14)

where in the last step the zero initial conditions (2.4b) and (2.4c) were used. For the boundary condition (2.4d) we get

\[
\hat{u}(x, s) = \hat{g}(x, s), \quad x \in \Gamma,
\]

(4.15)

under the assumption that \( g \) is Laplace transformable. Hence, the partial differential equation for \( \hat{u} \) is given by

\[-\Delta \hat{u}(\cdot, s) + s^2 \hat{u}(\cdot, s) = 0 \quad \text{in } \Omega
\]

(4.16a)

\[
\hat{u}(\cdot, s) = \hat{g}(\cdot, s) \quad \text{on } \Gamma.
\]

(4.16b)
To bring that equation into the form, in which a Helmholtz problem is usually stated (cf. (4.10)), one substitutes $\kappa = is$. Dropping the second argument, as it is just a parameter, leads to the Helmholtz equation for $\hat{u}$

$$-\Delta \hat{u} - \kappa^2 \hat{u} = 0 \quad \text{in } \Omega$$

$$\hat{u} = \hat{g} \quad \text{on } \Gamma.$$  

(4.17) (4.18)

This is the motivation to look at the Helmholtz equation in more detail.

4.5 Theory for the Helmholtz Problem

The theory about the Helmholtz problem is developed in [Néd01] and [SS04]. The problem to solve is defined by

$$-\Delta u - \kappa^2 u = 0 \quad \text{on } \Omega^e$$

$$u = g \quad \text{on } \Gamma$$

$$|u(x)| \leq C_1 \|x\|^{-1} \quad \text{for } \|x\| \to \infty$$

$$\left| \frac{\partial u}{\partial r} - i\kappa u \right| \leq C_2 \|x\|^{-2} \quad \text{for } \|x\| \to \infty,$$  

(4.19a) (4.19b) (4.19c) (4.19d)

for given boundary data $g$, wave number $\kappa \in \mathbb{C}$, $r = \frac{x}{\|x\|}$ and $\Omega^e \subset \mathbb{R}^3$ an exterior unbounded domain. The two conditions (4.19c) and (4.19d) assure the uniqueness of the solution. (4.19c) guarantees the decay towards infinity and (4.19d) is called Sommerfeld radiation condition. It assures that only outgoing waves are considered as solutions i.e. no new waves arise from the boundary at infinity. To solve the problem (4.19), a weighted Sobolev space is introduced for incorporating the radiation conditions into the space, where the function is searched. Let us introduce the differential operator $L_\kappa := -\Delta - \kappa^2$. One defines the scalar product

$$(u,v)_{H^1(L_\kappa, \Omega^e)} := \int_{\Omega^e} \left( \nabla \cdot \nabla \bar{v} + u \bar{v} \right) \frac{1}{1 + \|x\|^2} + \left( \frac{\partial u}{\partial r} - i\kappa u \right) \left( \frac{\partial \bar{v}}{\partial r} - i\kappa \bar{v} \right) \, dx.$$  

(4.20)

The space $H^1(L_\kappa, \Omega^e)$ is defined as the completion of $C^\infty_c(\Omega^e) := \{ \varphi |_{\Omega^e} \mid \varphi \in C^\infty_0(\mathbb{R}^3) \}$ with respect to the norm $\| \cdot \|_{H^1(L_\kappa, \Omega^e)} := \sqrt{\langle \cdot, \cdot \rangle_{H^1(L_\kappa, \Omega^e)}}$. The existence and uniqueness can then be found in [Néd01] Theorem 2.6.6, which are proved using the Lemma of Lax-Milgram.

**Theorem 4.5.1.** If $g \in H^{1/2}(\Gamma)$ then the variational formulation of (4.19) admits a unique solution in the space $H^1(L_\kappa, \Omega^e)$. The associated mapping is continuous from $H^{1/2}(\Gamma)$ into $H^1(L_\kappa, \Omega^e)$.

The proof is based on the fact that the single layer potential operator of the Helmholtz problem differs only in a compact perturbation in comparison to the one of the Laplace problem, which is elliptic.

An interesting remark is that the exterior problem (4.19), in contrary to the interior problem, that is given by replacing $\Omega^e$ by $\Omega^i$ in (4.19), has a solution for every $\kappa$, whereas the interior problem only has a solution if and only if $\kappa^2$ is not an eigenvalue of the interior problem for the Laplace equation. Unfortunately this problem arises if one wants to solve the exterior problem (4.19) using the single layer potential. The related theory can be found in [SS04] Chapter 3.9. We just want to recall the most important points concerning the application of a single layer potential ansatz for the Helmholtz problem. The fundamental solution of the Helmholtz equation is given by

$$G_\kappa(x) = \frac{e^{i\kappa\|x\|}}{4\pi\|x\|}.$$  

(4.21)
An important remark is the connection to the Laplace transformed fundamental solution of the wave equation $\hat{k}$ (cf. (3.21)). Applying once more the substitution $\kappa = is$ one gets the link

$$G_{\kappa}(x) = \hat{k}(|x|, -i\kappa).$$

(4.22)

The single layer potential for the Helmholtz equation is defined as (further details are given in Section 3.5)

$$(S\varphi)(x) := \int_{\Gamma} G_{\kappa}(x - y) \varphi(y) d\Gamma_y, \quad x \in \Omega^e.$$  

(4.23)

As well as for the wave equation, there is a result motivating that choice. In [SS04] the representation of the solution of elliptic partial differential equation using Green’s third identity is deduced. [SS04, Theorem 3.1.13] just gives the representation

**Theorem 4.5.2.** Consider the Helmholtz operator $L_{\kappa}u := -\Delta u - \kappa^2 u$ in the three dimensional space for a wavenumber $\kappa > 0$. If $u \in H^1(\Omega^i) \times H^1(L_{\kappa}, \Omega^e)$ solves $L_{\kappa}u = 0$ in $\mathbb{R}^3 \setminus \Gamma$, then it holds

$$u = -S[\gamma_1 u] + D[\gamma_0 u],$$

(4.24)

with $S$ the single layer operator and $D$ the double layer operator\(^1\) for the Helmholtz problem.

Using the same arguments as for the wave equation developed in Section 4.7 we get the motivation for solving the Dirichlet problem of the Helmholtz problem using a single layer potential.

Although the exterior Dirichlet problem for the Helmholtz equation admits a unique solution for each $\kappa \in \mathbb{C}$, there arise problems by applying the approach of a single layer potential. The main result is given by [SS04, Theorem 3.9.1] and states that there are critical values for $\kappa$, for which the single layer potential operator is not invertible.

**Theorem 4.5.3.** The single layer potential operator of the Helmholtz problem for a $\kappa \in \mathbb{C}$ is invertible on $H^{-1/2}(\Gamma)$ if and only if $\kappa^2$ is not an eigenvalue of the interior problem of the Laplace equation.

Therefore, if one wants to solve an exterior Helmholtz problem using the single layer potential, one has to assure that $\kappa^2$ is not an eigenvalue of the interior Laplace problem. Therefore it is important to know more about these critical values. The Laplace operator $-\Delta$ is a symmetric elliptic operator. The following result for the eigenvalues of such operators can be found in [Eva98, Theorem 1, p. 335].

**Theorem 4.5.4.** All eigenvalues of a symmetric elliptic operator are real and positive.

Hence, a sufficient condition on the arising $\kappa$ is useful, to avoid the problem of the non-invertibility of the single layer potential operator. A condition that will match the wave numbers used in this thesis is presented in the following Proposition.

**Proposition 4.5.5.** A wave number $\kappa \in \mathbb{C}$ with $\text{Im}(\kappa) > 0$ is not a critical value and allows therefore to employ the single layer potential given by (4.23), in order to solve the exterior Dirichlet problem of the Helmholtz equation as defined in (4.19).

**Proof.** Let be $\kappa = a + bi$ with $a, b \in \mathbb{R}$ and $b > 0$. It holds for the squared wave number $\kappa^2 = a^2 - b^2 + 2abi$. Assume, that $\kappa^2$ is real, i.e. $2ab = 0$. As $b > 0$, it follows that $a = 0$, which results in $\kappa^2 = -b^2 < 0$. Therefore, no critical value can occur for such $\kappa$. \(\square\)

\(^1\)For the definition and the properties of the double layer operator $D$ we refer to [SS04, Section 3.1].
4.6 Properties of the Single Layer Potential

First of all, the single layer potential as noted in (2.5) has a distributional kernel $k$. It is given by the fundamental solution of the wave equation. A detailed derivation of it is omitted to [Wla72 Chapter 10]. Its derivation relies on the theory of distributions, the generalized functions. The fundamental solution has to satisfy

$$k_{tt}(x, t) - \Delta k(x, t) = \delta(x)\delta(t) \quad \forall (x, t) \in \mathbb{R}^3 \times \mathbb{R}_{>0}$$  \hspace{1cm} (4.25a)

$$k(x, t) = 0 \quad \forall (x, t) \in \mathbb{R}^3 \times \mathbb{R}_{\leq0},$$  \hspace{1cm} (4.25b)

where the occurring derivatives are derivatives in the sense of distributions and $\delta$ the Dirac distribution. By applying the Fourier transform on the first equation with respect to the variable $x$, an ordinary differential equation in $t$ arises, which can be solved explicitly. The result then is transformed back by the inversion formula for the Fourier transform.

Before starting examining the single layer potential, let us try to get rid of the distributional term in $k$. Hence

$$(S\varphi)(x, t) = \int_0^t \int_{\Gamma} k(x-y, t-\tau)\varphi(y, \tau)\,d\Gamma_y\,d\tau = \int_0^t \int_{\Gamma} \frac{\delta(||x-y|| - (t-\tau))}{4\pi||x-y||}\varphi(y, \tau)\,d\Gamma_y\,d\tau.$$  \hspace{1cm} (4.26)

Furthermore it holds $\tau \leq t$ and therefore

$$||x-y|| - (t-\tau) = 0 \iff \tau = t - ||x-y||,$$  \hspace{1cm} (4.27)

i.e. exactly in the case when $\tau$ is equal to the retarded time $t - ||x-y||$. Formally changing the order of integration and setting $\varphi(\cdot, s) = 0$ for $s < 0$, leads to

$$(S\varphi)(x, t) = \int_{\Gamma} \int_0^t k(x-y, t-\tau)\varphi(y, \tau)\,d\tau\,d\Gamma_y = \int_{\Gamma} \frac{\varphi(y, t - ||x-y||)}{4\pi||x-y||}\,d\Gamma_y,$$  \hspace{1cm} (4.28)

that is the representation deduced in Section 4.3 (cf. equation (4.8)).

Let us continue with the theorems about the properties and regularity of the single layer potential. First it will be shown that the single layer potential satisfies the wave equation and the initial values conditions.

**Theorem 4.6.1.** For $x \in \Omega^c$, $\varphi$ regular enough and vanishing for $t < 0$, the single layer potential $(S\varphi)(x, t)$ solves the wave equation

$$u_{tt}(x, t) - \Delta u(x, t) = 0 \quad \text{in } \Omega^c \times (0, T)$$  \hspace{1cm} (4.29a)

$$u(x, 0) = u_t(x, 0) = 0 \quad \text{on } \Omega^c,$$  \hspace{1cm} (4.29b)

**Proof.** Let us define $u(x, t) := (S\varphi)(x, t)$ on $\Omega^c \times (0, T)$. Remark that for a fixed $x \in \Omega^c$, there exists a $C > 0$, such that $||x-y|| \geq C$ for every $y \in \Gamma$. Hence the integral in the definition of $(S\varphi)(x, t)$ exists as a proper integral, as $\Gamma$ is supposed to be compact. Therefore the integration and differentiation can be interchanged. Using the same argument one can show the continuity on $\Omega^c$. Let us denote by $\varphi_t, \varphi_{tt}$ the derivatives of $\varphi$ in the second argument. Furthermore it holds for $x \neq y$

$$\frac{\partial}{\partial x_i} ||x-y|| = \frac{\partial}{\partial x_i} \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + (x_3-y_3)^2} = \frac{x_i-y_i}{||x-y||} \quad \forall i \in \{1, 2, 3\}.$$  \hspace{1cm} (4.30)
Let us calculate now the derivatives in the time variable of \( u \) using the representation (4.28)

\[
 u_t(x, y) = \int_\Gamma \frac{\varphi_t(y, t - \|x - y\|)}{4\pi \|x - y\|} \, d\Gamma_y,  
\]

(4.31)

\[
 u_{tt}(x, y) = \int_\Gamma \frac{\varphi_{tt}(y, t - \|x - y\|)}{4\pi \|x - y\|} \, d\Gamma_y.  
\]

(4.32)

The next step is to determine the derivatives in space using the same representation

\[
 u_x = \int_\Gamma -\frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|^2} \cdot \frac{x_i - y_i}{\|x - y\|} \, d\Gamma_y,  
\]

(4.33)

\[
 u_{x_i} = \int_\Gamma -\frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|^3} \cdot \frac{x_i - y_i}{\|x - y\|^2} \cdot \varphi_t(y, t - \|x - y\|) \cdot \frac{x_i - y_i}{\|x - y\|} \, d\Gamma_y  
\]

(4.34)

\[
 u_{x_i x_j} = -\int_\Gamma \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|^4} \cdot \frac{x_i - y_i}{\|x - y\|} \cdot \varphi_t(y, t - \|x - y\|) \cdot \frac{x_j - y_j}{\|x - y\|} + \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|^3}  
\]

(4.35)

The last step is to sum now over \( i \). Using

\[
 \sum_{i=1}^3 \frac{(x_i - y_i)^2}{\|x - y\|} = \|x - y\|,  
\]

(4.36)

leads to

\[
 -\Delta u = \int_\Gamma -\frac{\varphi_t(y, t - \|x - y\|)}{4\pi \|x - y\|^2}  
+ 3 \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|^3} - 3 \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|^4}  
- \frac{\varphi_{tt}(y, t - \|x - y\|)}{4\pi \|x - y\|^2}  
+ 3 \frac{\varphi_t(y, t - \|x - y\|)}{4\pi \|x - y\|^3} - 2 \frac{\varphi_t(y, t - \|x - y\|)}{4\pi \|x - y\|^2} \, d\Gamma_y  
\]

(4.37)

\[
 = -\int_\Gamma \frac{\varphi_{tt}(y, t - \|x - y\|)}{4\pi \|x - y\|^2} \, d\Gamma_y.  
\]

(4.38)

Finally that shows that \( u \) satisfies

\[
 u_{tt} - \Delta u = \int_\Gamma \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|} \, d\Gamma_y - \int_\Gamma \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|} \, d\Gamma_y = 0.  
\]

(4.39)

Hence we have to look at the values of \( u \) and \( u_t \) for \( t = 0 \). For the value of \( u \) at \( t = 0 \) it holds

\[
 u(x, 0) = \int_\Gamma \frac{\varphi(y, -\|x - y\|)}{4\pi \|x - y\|} \, d\Gamma_y.  
\]

(4.40)

As \( \|x - y\| > 0 \) and \( \varphi \) is vanishing for \( t < 0 \), it holds \( u(x, 0) = 0 \) for \( x \in \Omega \). Assuming that \( \varphi \) is twice continuously differentiable with respect to the time variable yields to \( \varphi_t = 0 \) for \( t < 0 \). Using the same argument as before ensures \( u_t(x, 0) = 0 \) as

\[
 u_t(x, 0) = \int_\Gamma \frac{\varphi_t(y, -\|x - y\|)}{4\pi \|x - y\|} \, d\Gamma_y.  
\]

(4.41)
To impose Dirichlet boundary data, the behaviour of the single layer potential over the boundary $\Gamma$ has to be determined. As discussed in Footnote 2 on Page 17, the representation of the single layer potential given by (2.5) holds only for densities $\varphi$ which are regular enough. To get a representation of $S$ on the boundary $\Gamma$, denoted by $V$, one also has to restrict to regular $\varphi$. As later on all densities $\varphi$ used for the discretisation in space satisfy $\varphi(\cdot, t) \in S$, where $S$ the boundary element space is defined as $S_{1,0}$ or $S_{0,1}$, which both are a subset of $L^\infty(\Gamma)$ (cf. Section 5.2), we develop a representation of $V$ for $\varphi(\cdot, t) \in L^\infty(\Gamma)$.

The appropriate spaces with respect to the time are introduced in Section 3.4.6. These considerations motivate the choice $\varphi \in C(0,T; L^\infty(\Gamma))$.

**Theorem 4.6.2.** Let $\Gamma$ be the surface of a bounded Lipschitz domain $\Omega^i \subset \mathbb{R}^3$ and $\varphi \in C(0,T; L^\infty(\Gamma))$. The integral representing the single layer potential for such a $\varphi$ given by

$$
(S\varphi)(x, t) = \int_\Gamma \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|} d\Gamma_y,
$$

exists as an improper integral for all $x \in \mathbb{R}^3$ and defines a continuous function $S\varphi$ on $\mathbb{R}^3$.

**Proof.** The proof is based on the proof for the continuity of the single layer potential for the Laplace equation effected in [Néd01, Theorem 3.1.2]. For arguments $t - \|x - y\| < 0$ set the value of $\varphi$ to zero. Using the definition of the function space, we get the existence of a constant $M > 0$ such that

$$
\|\varphi\|_{C(0,T; L^\infty(\Gamma))} = \sup_{t \in [0,T]} \|\varphi(\cdot, t)\|_{L^\infty(\Gamma)} = \sup_{t \in [0,T]} \sup_{x \in \Gamma} |\varphi(x, t)| < M. \quad (4.43)
$$

Therefore, we get a majorant function for the integrand almost everywhere

$$
\left| \frac{\varphi(y, t - \|x - y\|)}{4\pi \|x - y\|} \right| \leq M \cdot \frac{1}{4\pi \|x - y\|} \quad \forall t \in [0, T], \; x, y \in \Gamma \text{ with } x \neq y. \quad (4.44)
$$

Hence the proof is lead back to the case of the single layer potential of the Laplace equation as $\frac{1}{4\pi \|x\|}$ is the fundamental solution of the Laplace equation on $\mathbb{R}^3$. First of all we show the existence as improper integral using a majorant function $f$ that is integrable. To show that the integral is transformed onto a two dimensional domain by using the Lipschitz property of the domain. The mentioned majorant function $f$ also plays the key role in the proof for the continuity, as it allows to interchange the limit, linked to a sequence $(x_n)_{n \in \mathbb{N}}$ converging to $x$, and the integration using the theorem of Lebesgue.

**4.7 Existence and Uniqueness for the Single Layer Potential**

As a last step in the theory of the exact problem, the existence and uniqueness of the problem (2.8) in the formulation using the single layer potential has to be discussed. The results are taken from [BHD86, Proposition 3] respectively [Lmb94, (2.22), (2.24)] and use the functions spaces defined in Section 3.4.6.

**Theorem 4.7.1.** Let $g \in H^{r+2}_0(0, T; H^{1/2}(\Gamma))$ for some $r \in \mathbb{R}$. Then

$$
(V\varphi)(x, t) := \int_0^t \int_\Gamma k(x - y, t - \tau) \varphi(y, \tau) d\Gamma_y d\tau = g(x, t) \quad \forall (x, t) \in \Gamma \times (0, T),
$$

has an unique solution $\varphi \in H_r^0(0, T; H^{-1/2}(\Gamma))$ with

$$
\|\varphi\|_{H_r^0(0, T; H^{-1/2}(\Gamma))} \leq C_T \|g\|_{H^{r+2}_0(0, T; H^{1/2}(\Gamma))}. \quad (4.46)
$$
4.7. EXISTENCE AND UNIQUENESS FOR THE SINGLE LAYER POTENTIAL

For $r > 1/2$, the pointwise estimate

$$\| \varphi(\cdot, t) \|_{H^{-1/2}(\Gamma)} \leq C_T \| g \|_{H^{r+2}(\Gamma; H^{1/2}(\Gamma))},$$

holds for all $t \in [0, T]$.

Proof. The proof can be found in [Lub94]. The most important steps and ideas are listed in this sketch of the proof. Fundamental is the reducing of the wave equation to the Helmholtz equation using the Laplace transform. That also holds for the single layer potential by setting

$$\varphi(y, \tau) = 0 \text{ for } \tau < 0 \text{ and keeping in mind that } \tau > t \text{ holds } \| x - y \| > (t - \tau) > 0.$$

We get the following mapping property for the single layer potential

$$\mathcal{V}(\varphi)(s) = \mathcal{L}\left( \int_0^t \int_{\Gamma} \frac{\delta(\| x - y \| - (t - \tau))}{4\pi \| x - y \|} \varphi(y, \tau) d\Gamma_y d\tau \right)(s)$$

(4.48)

$$= \mathcal{L}\left( \int_{-\infty}^t \int_{\Gamma} \frac{\delta(\| x - y \| - (t - \tau))}{4\pi \| x - y \|} \varphi(y, \tau) d\Gamma_y d\tau \right)(s)$$

(4.49)

$$= \mathcal{L}\left( \int_{\Gamma} \frac{\delta(\| x - y \| - (t - \tau))}{4\pi \| x - y \|} \varphi(y, \tau) d\Gamma_y d\tau \right)(s)$$

(4.50)

$$= \int_{\Gamma} \frac{k(\| x - y \|, t) \ast \varphi(y, t)}{\varphi(y, \tau) d\Gamma_y} ds$$

(4.51)

$$= \int_{\Gamma} k(\| x - y \|, t) \ast \varphi(y, t)(s) d\Gamma_y$$

(4.52)

$$= \int_{\Gamma} \hat{k}(\| x - y \|, s) \hat{\varphi}(y, s) d\Gamma_y =: \mathcal{V}(s) \hat{\varphi}(\cdot, s),$$

(4.53)

with $\hat{k}$ as in the Section 3.3 being the fundamental solution of the Helmholtz problem as we have seen in the Section 4.3. Hence the existence and uniqueness of the single layer potential for the Helmholtz problem, denoted by $\mathcal{V}(s)$, can be considered. As this is discussed in Chapter 4.5 with all necessary references, only the results needed for the transformation back to the wave equation are quoted here. It is provided by [BHD86] and is cited in [Lub94] Proposition 2.3.

Lemma 4.7.2. For $\text{Re}(s) = \sigma > 0$, the single layer potential operator for the Helmholtz problem $\mathcal{V}(s)$ extends by density to an isomorphism

$$\mathcal{V}(s) : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma),$$

(4.54)

which for all $\psi \in H^{-1/2}(\Gamma)$ satisfies coercivity and boundedness

$$\text{Re} \left( \langle s \mathcal{V}(s) \psi, \psi \rangle \right) \geq c \cdot \min(1, \sigma) \| \psi \|_{H^{-1/2}(\Gamma)}^2$$

(4.55)

$$\| \mathcal{V}(s) \psi \| \leq C \cdot |s| \cdot \max(1, \sigma^{-2}) \| \psi \|_{H^{1/2}(\Gamma)}$$

(4.56)

where $c$ and $C$ are positive constants, which depend only on $\Gamma$.

Therefore, we get the following mapping property for the single layer potential

$$\mathcal{V}(s) : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma), \ \varphi \mapsto g$$

(4.57)

where $\varphi$ is the density and $g$ the Dirichlet data. Furthermore a bound for $\mathcal{V}^{-1}(s)$ is needed and is provided by [BHD86] Proposition 2 and quoted in [Lub94] Equation 2.20.

Lemma 4.7.3. For $\text{Re}(s) > \sigma_0 > 0$ the following bound holds

$$\| \mathcal{V}^{-1}(s) \| \leq M(\sigma_0) \cdot |s|^2,$$

(4.58)

where $M(\sigma_0)$ is a positive constant depending on $\sigma_0$ and $\Gamma$. 

The transform of this estimate to the wave equation in the time domain is presented in [Lub94]. In the ‘operational calculus’ the convolution with respect to the time of the form \( k \ast g \) is denoted by \( \hat{k}(\partial_t)g \) to emphasize on Laplace transform \( \hat{k} \) of \( k \) instead of \( k \) itself. Furthermore, using only \( \hat{k}, \hat{k}(\partial_t)g \) is given by the following Definition.

**Definition 4.7.4.** Let \( \hat{k}(s) \) be an analytic function for \( \text{Re}(s) > \sigma_0 \) that is bounded for large \( |s| \) by

\[
|\hat{k}(s)| \leq M \cdot |s|^\mu, \quad (4.59)
\]

for \( \mu \in \mathbb{R} \). Let \( \sigma > \sigma_0, \mathbb{N} \ni m > \mu + 1 \) and \( g \in C^m(\mathbb{R}) \). Then \( \hat{k}(\partial_t)g \) is defined by

\[
\left( \hat{k}(\partial_t)g \right)(t) := \int_0^\infty \left( \frac{1}{2\pi i} \int_{\sigma+i\mathbb{R}} e^{s\tau} \hat{k}(s) \frac{d s}{s^m} \right) g^{(m)}(t - \tau) d \tau. \quad (4.60)
\]

The properties of the operator \( \hat{k}(\partial_t) \) include the composition rule

\[
\hat{k}_2(\partial_t) \hat{k}_1(\partial_t)g = \left( \hat{k}_2 \cdot \hat{k}_1 \right)(\partial_t)g \quad (4.61)
\]

and, for \( g \) Laplace transformable, the relation

\[
\mathcal{L} \left( \hat{k}(\partial_t)g \right)(s) = \hat{k}(s) \cdot (\mathcal{L}g)(s) \quad (4.62)
\]

on the appropriate half-plane in \( \mathbb{C} \). The essence of this approach is summarized in the following Theorem.

**Theorem 4.7.5.** If \( \hat{k}(s) \) is an analytic function for \( \text{Re}(s) > \sigma_0 \) that is bounded for large \( |s| \) by

\[
|\hat{k}(s)| \leq M \cdot |s|^\mu, \quad (4.63)
\]

for \( \mu \in \mathbb{R} \), then \( \hat{k}(\partial_t) \) and therefore \( k \ast g \) extends by density to a bounded linear operator

\[
\hat{k}(\partial_t) : H^{r+\mu}_0(0, T) \to H^r_0(0, T). \quad (4.64)
\]

This theory is applied to Banach space valued operators. Using the definition of the integral operator

\[
z(t) : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma), \quad f \mapsto \int_\Gamma \frac{\delta(\|y - y\| - t)}{4\pi \|y - y\|} f(y) d\Gamma_y, \quad (4.65)
\]

the equation (4.45) can be rewritten in terms of time convolution of integral operators on \( \Gamma \)

\[
\int_0^t \int_\Gamma \frac{\delta(\|x - y\| - (t - \tau))}{4\pi \|x - y\|} \varphi(y, \tau) d\Gamma_y d\tau = \left( \int_0^t z(t - \tau) \varphi(\cdot, \tau) d\tau \right)(x) = \left( (z * \varphi)(t) \right)(x) = g(x, t), \quad \text{on } \Gamma \times [0, T], \quad (4.66)
\]

or as equation in \( H^{1/2}(\Gamma) \)

\[
(z * \varphi)(t) = g(\cdot, t), \quad t \in [0, T]. \quad (4.67)
\]

Applying the Laplace transform on the left-hand side of (4.68), we get

\[
\mathcal{L}(z * \varphi)(s) = \hat{\varphi}(s). \quad (4.69)
\]
Hence, for the Laplace transformed operator \( \hat{\mathcal{L}}(s) \) we exactly get the mapping,

\[
f \mapsto \int_{\Gamma} \frac{e^{-s\|y\|}}{4\pi\|y\|} f(y) d\Gamma_y,
\]

which is the single layer potential operator for the Helmholtz problem \( \hat{\mathcal{L}}(s) = \mathcal{V}(s) \). Therefore using the calculus mentioned above one can write the equation (4.45) as

\[
\mathcal{V}(\partial_t) \varphi = g.
\]

Using the mapping properties of \( \mathcal{V}(s) \) and the composition rule presented in (4.61), one gets

\[
\varphi(t) = (\mathcal{V}^{-1}(\partial_t) \mathcal{V}(\partial_t) \varphi)(t) = (\mathcal{V}^{-1}(\partial_t) g)(t).
\]

Taking the bound of Lemma 4.7.3, the result of Theorem 4.7.5 and the mapping properties into consideration, one gets the asserted mapping properties of \( \mathcal{V} \). Furthermore the boundedness given in the latter theorem leads to the stated estimates. For \( r > 1/2 \) Sobolev’s embedding theorem implies that \( \varphi \in H^r_0(0,T; H^{-1/2}(\Gamma)) \) has a continuous representative with respect to the time variable. Therefore a pointwise estimate is possible.
Chapter 5

Numerical Discretisation

In order to calculate the solution of the wave equation given by (2.4), an indirect method using the single layer potential is applied. The goal therefore is to determine the unknown density $\varphi$ numerically, which will allow the calculation $u$ at any given field point in the exterior domain for any given time. Before applying a boundary element method as discretisation in space, the time dependence is discretized by applying the convolution quadrature presented in Section 3.6.

5.1 Time Discretisation

The first step is to discretise (2.8) in the time variable $t$. Let $\Delta t = \frac{T}{N}$ be the time step for a $N \in \mathbb{N}_{>0}$. The $N + 1$ equidistant discrete times are defined by $t_n = n \cdot \Delta t$, for $n = 0, \ldots, N$.

Recall the integral equation

$$\int_0^t \int_{\Gamma} k(x - y, t - \tau) \varphi(y, \tau) d\Gamma_y d\tau = g(x, t) \quad \forall (x, t) \in \Gamma \times (0, T). \quad (5.1)$$

In this case, in comparison to the convolution of two real valued functions (cf. Section 3.6), the coefficients $\omega_{n}^{\Delta t}$ depend also on the space variable, i.e. they are functions themselves. As $k(x - y, t - \tau) = \frac{8(\|x-y\|^{2}-(t-\tau))}{4\pi\|x-y\|^{4}}$, they just depend on $\|x - y\|$. If the two integrals in (5.1) are commutable, we get (5.1) in convolution form

$$\int_0^t \int_{\Gamma} k(x - y, t - \tau) \varphi(y, \tau) d\Gamma_y d\tau = \int_{\Gamma} \int_0^t k(x - y, t - \tau) \varphi(y, \tau) d\tau d\Gamma_y \quad (5.2)$$

$$\approx \int_{\Gamma} \sum_{j=0}^{n} \omega_{n-j}^{\Delta t}(\|x-y\|) \varphi_{j}^{\Delta t}(y) d\Gamma_y = \sum_{j=0}^{n} \int_{\Gamma} \omega_{n-j}^{\Delta t}(\|x-y\|) \varphi_{j}^{\Delta t}(y) d\Gamma_y. \quad (5.3)$$

Therefore, let us define the discrete convolution operator

$$(V \varphi)_{n} := \sum_{j=0}^{n} \int_{\Gamma} \omega_{n-j}^{\Delta t}(\|x-y\|) \varphi_{j}^{\Delta t}(y) d\Gamma_y, \quad n = 0, \ldots, N, \quad (5.4)$$

which leads to the semi-discrete problem

$$(V \varphi)_{n}(x) = g_{n}(x), \quad n = 1, \ldots, N, \quad x \in \Gamma, \quad (5.5)$$

with $g_{n}(x) = g(x, t_n)$.

Using $d := \|x - y\|$ the coefficients $\omega_{n}^{\Delta t}(\|x-y\|)$ are denoted as functions $\omega_{n}^{\Delta t}(d)$ with respect to $d$. The next step is to calculate these coefficients using the rule developed in Section 3.6.
Due to the considerations in Section 3.7, we use BDF2, an A-stable linear multistep method of order 2, for the discretisation. The ratio of the corresponding generating polynomials (3.7) is given by
\[
\gamma(\zeta) = \frac{1}{2} \zeta^2 - 2\zeta + \frac{3}{2}.
\] (5.6)

As explained in Section 3.6 (cf. (3.83)) the coefficients of the power series of \( \hat{f}(\gamma(\zeta)\Delta t) \) with respect to \( \zeta \) are the weights \( \omega_n^\Delta t \). For the case of \( k \) being the fundamental solution of the wave equation, it is shown in [HK09] that \( \omega_n^\Delta t \) can be represented explicitly by
\[
\omega_n^\Delta t(d) = \frac{1}{n!} \frac{1}{4\pi d} \left( \frac{d}{2\Delta t} \right)^{n/2} e^{-\frac{3d}{2\Delta t}} H_n \left( \sqrt{\frac{2d}{\Delta t}} \right),
\] (5.7)
where \( H_n \) denote the \( n \)-th Hermite polynomial. There are several possibilities how to calculate \( \omega_n^\Delta t(d) \) as discussed in Section 5.3.

5.2 Space Discretisation

To discretise the spatial problem on \( \Gamma \) a Galerkin boundary element method is applied. First of all a surface paneling of \( \Gamma \) is needed.

5.2.1 Surface Paneling

The theory, which is presented here, follows [SS04, Chapter 4.1.2]. We introduce a boundary element mesh, also called paneling, \( G \) on \( \Gamma \). We restrict to plane triangles with straight edges, i.e. \( \Gamma \) is the surface of a polyhedron. Let \( \tau \in G \), then we have a parametrization \( \chi_\tau : \hat{\tau} \to \tau \), where \( \hat{\tau} \) is the unit triangle defined by
\[
\hat{\tau} = \text{conv} \left\{ \left( 0, 0 \right), \left( 1, 0 \right), \left( 1, 1 \right) \right\}.
\] (5.8)

Let us define for a triangle \( \tau \) with vertices \( A, B, C \in \mathbb{R}^3 \) the matrix \( M_\tau \) by
\[
M_\tau := \begin{pmatrix} \mathbf{B} - \mathbf{A} & \mathbf{C} - \mathbf{B} \end{pmatrix}.
\] (5.9)

Then the parametrization \( \chi_\tau \) is affine and is defined by
\[
\chi_\tau(\hat{x}) = \mathbf{A} + M_\tau \cdot \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix}.
\] (5.10)

The matrix \( M_\tau \) is the Jacobian of \( \chi_\tau \). If \( \tau \) is not degenerated, i.e. the vertices are not colinear, the columns of the matrix \( M_\tau \) are linearly independent and therefore \( M_\tau^T M_\tau \) is positive definite. Thus there exist \( \lambda_{\text{min}}, \lambda_{\text{max}} > 0 \) such that
\[
0 < \lambda_{\text{min}} \leq \inf_{v \in \mathbb{R}^2, \|v\| = 1} v^T M_\tau^T M_\tau v \leq \sup_{v \in \mathbb{R}^2, \|v\| = 1} v^T M_\tau^T M_\tau v \leq \lambda_{\text{max}} < \infty.
\] (5.11)

Hence the mapping \( \chi_\tau \) is regular in the sense of [SS04, Definition 4.1.2]. Let us state the definition for a regular paneling \( G \) of \( \Gamma \).

**Definition 5.2.1.** A paneling \( G \) of the boundary \( \Gamma \) is a partitioning of \( \Gamma \) into relatively open, disjoint elements \( \tau \subset \Gamma \) with
5.2. SPACE DISCRETISATION

(i) $\mathcal{G}$ is a covering of $\Gamma$, i.e., $\Gamma = \bigcup_{\tau \in \mathcal{G}} \tau$.

(ii) Each element $\tau \in \mathcal{G}$ is the image of the reference element $\hat{\tau}$ using a regular $\chi_{\tau}$ as stated in [5.11].

As we have seen before, the mappings for the case of plane triangles with straight edges fulfil this definition.

Remark 5.2.2. Our assumptions on $\mathcal{G}$ imply that $\Gamma$ is the surface of a polyhedron. If $\Gamma$ is curved, then an additional approximation arises if $\Gamma$ is approximated by plane triangles or higher order parametric surface elements. For an error analysis we refer to [SS10].

In order to prove error estimates we have to impose additional properties on $\mathcal{G}$.

Definition 5.2.3. A paneling $\mathcal{G}$ of $\Gamma$ is regular if

(i) The intersection of the closure of two different elements $\tau, t \in \mathcal{G}$ either is empty or contains a common vertex or a common edge.

(ii) The parametrizations of common edges are compatible, i.e. if $\tau, t \in \mathcal{G}$, with $\tau \neq t$, have a common edge $e = \tau \cap t$, then it holds

\[ \chi_{\tau}|_{\hat{e}} = \chi_{t} \circ \gamma_{\tau,t}|_{\hat{e}}, \tag{5.12} \]

with $\hat{e} = \chi_{\tau}^{-1}(e)$ and $\gamma_{\tau,t} : \hat{\tau} \to \hat{t}$ an affine bijection.

The latter point is fulfilled in the case of plane triangles with straight lines. Assume that $\mathcal{G}$ is regular, i.e. the properties for the intersection of two panels hold, and only contains plane triangles. To get the properties of the triangulation state

Definition 5.2.4.

(i) The diameter of a triangle $\tau \in \mathcal{G}$ is given by $h_\tau := \sup_{x,y \in \tau} \|x - y\|$.

(ii) The inner width $\rho_\tau$ is defined by the incircle diameter of $\tau$.

(iii) The mesh width $h_\mathcal{G}$ (shorthand: $h$) is defined by $h_\mathcal{G} := \max_{\tau \in \mathcal{G}} h_\tau$.

(iv) The shape-regularity constant $\kappa_\mathcal{G}$ is given by $\kappa_\mathcal{G} := \max_{\tau \in \mathcal{G}} \frac{h_\tau}{\rho_\tau}$.

(v) The constant $q_\mathcal{G}$ describes the quasi-uniformity by $q_\mathcal{G} := \frac{h_\mathcal{G}}{\min_{\tau \in \mathcal{G}} h_\tau}$.

The variable $\rho_\tau, \kappa_\mathcal{G}$ and $q_\mathcal{G}$ will not be tracked explicitly in the convergence estimates but are hidden in the generic constant $C$. They just remind us that the various triangulations should not contain too big differences in terms of the shape of triangles nor degenerate in terms of extremely cute angles of panels.

Definition 5.2.5. $P_\mathcal{G}$ is the set of all vertices of $\mathcal{G}$ defined by

\[ P_\mathcal{G} := \{ P \in \mathbb{R}^3 \mid \exists \tau \in \mathcal{G} \text{ such that } P \text{ is a vertex of } \tau \} \tag{5.13} \]

For quantity issues we define the following cardinalities

Definition 5.2.6.

(i) The number of panels in $\mathcal{G}$ is defined by $N_\tau := |\mathcal{G}|$

(ii) The number of vertices in $\mathcal{G}$ is defined by $N_P := |P_\mathcal{G}|$

Therefore we will number the panels as $\tau_k$ with $1 \leq k \leq N_\tau$ resp. the vertices as $P_k$ with $1 \leq k \leq N_P$ and it holds

\[ \{\tau_k \mid 1 \leq k \leq N_\tau\} = \mathcal{G} \text{ resp. } \{P_k \mid 1 \leq k \leq N_P\} = P_\mathcal{G}. \tag{5.14} \]
5.2.2 Piecewise Constant Boundary Elements

As a first boundary element space \( S_{-1,0} \) we take discontinuous boundary elements. This makes sense, as \( V : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma) \) and the boundary element space will be a subspace of \( H^{-1/2}(\Gamma) \).

**Definition 5.2.7.** Let \( S_{-1,0} \) denote the space of piecewise constant, discontinuous functions

\[
S_{-1,0} := \{ u \in L^\infty(\Gamma) \mid \forall \tau \in \mathcal{G} : u|_{\tau} \circ \chi_{\tau} \in \mathbb{P}_0 \},
\]

with \( \mathbb{P}_0 \) the space of all constant functions on \( \hat{\tau} \).

As \( u \in S_{-1,0} \) implies \( u \in L^\infty(\Gamma) \) and the set of edges and vertices of the boundary element mesh is of measure zero, \( u \) is defined by its values on the interior of the elements \( \tau \in \mathcal{G} \). Therefore it can be written as

\[
u(x) = \sum_{\tau \in \mathcal{G}} u_{\tau} b_{\tau}(x),
\]

with the characteristic function \( b_{\tau} : \Gamma \to \mathbb{R} \)

\[
b_{\tau}(x) := \begin{cases} 1 & x \in \tau \\ 0 & \text{otherwise} \end{cases}.
\]

Therefore we take \( \{ b_{\tau} \mid \tau \in \mathcal{G} \} \) as a basis for \( S_{-1,0} \) and we get the dimension of the boundary element space \( M = N_\tau \).

5.2.3 Piecewise Linear Boundary Elements

As a second boundary element space \( S_{0,1} \) more regular functions are considered.

**Definition 5.2.8.** Let \( S_{0,1} \) denote the space of continuous, piecewise linear functions

\[
S_{0,1} := \{ u \in C^0(\Gamma) \mid \forall \tau \in \mathcal{G} : u|_{\tau} \circ \chi_{\tau} \in \mathbb{P}_1 \},
\]

where \( \mathbb{P}_1 \) is the space containing all polynomials on \( \hat{\tau} \) of the form \( a_0 + a_1 \hat{x}_1 + a_2 \hat{x}_2, a_0, a_1, a_2 \in \mathbb{R} \).

As a basis we choose the continuous, piecewise affine hat functions \( b_m \) for \( 1 \leq m \leq N_P \), which have the value 1 at the vertex \( P_m \), vanish at all other vertices and interpolate linearly between the vertices. Hence the support is given by

\[
\text{supp} b_m = \bigcup_{k \in \iota_m} \tau_k,
\]

with the definition for the set of indices \( \iota_m \) as in Definition 5.2.9.
Definition 5.2.9. For $1 \leq m \leq N_p$, let $\iota_m$ denote the set of indices of all triangles in $G$, which contain $P_m$ as a vertex:

$$\iota_m := \{1 \leq k \leq N_T \mid P_m \in \tau_k \} = \{1 \leq k \leq N_T \mid P_m \text{ is a vertex of } \tau_k \}.$$ (5.20)

For a $k \in \iota_m$, the pullback $b_m \mid_{\tau_k} \circ \chi_{\tau_k}$ is one of the following shape functions on the reference element $\hat{\tau}$, defined by

$$\hat{N}_0 : \hat{\tau} \to \mathbb{R} : \hat{N}_0(\hat{x}) = 1 - \hat{x}_1$$ (5.21)

$$\hat{N}_1 : \hat{\tau} \to \mathbb{R} : \hat{N}_1(\hat{x}) = \hat{x}_1 - \hat{x}_2$$ (5.22)

$$\hat{N}_2 : \hat{\tau} \to \mathbb{R} : \hat{N}_2(\hat{x}) = \hat{x}_2,$$ (5.23)

depending on the local numbering of the vertices of $\tau_k$, but in such a way that $b_m(P_k) = \delta_{mk}$. These shape functions fulfill $\mathbb{P}_1 = \text{span}\{\hat{N}_0, \hat{N}_1, \hat{N}_2\}$ and therefore $b_1, \ldots, b_{N_p}$ build a basis of $S_{0,1}$. The dimension of the boundary element space is then $M = N_p$.

![Figure 5.2: Continuous linear basis function $b_m$](image)

5.2.4 Discretisation for a Given Basis

Let $\{b_1, \ldots, b_M\}$ for $M \in \mathbb{N}_{>0}$ be the basis of the boundary element space $S$ with underlying paneling $G$. This assumption is compatible with the precedent two Sections 5.2.2 and 5.2.3. As characteristic discretization parameter for the time resp. space discretisation we take the time step $\Delta t$ resp. the mesh width $h$. We will derive error estimates with respect to $\Delta t$ and $h$ next. For a timestep $t_n$, $0 \leq n \leq N$, we employ the ansatz

$$\varphi(y, t_n) \approx \varphi_{n,h}^\Delta(t_n, y) := \sum_{m=1}^M \varphi_{n,m} b_m(y),$$ (5.24)

where $\varphi_{n,m}$ denote the coefficients corresponding to $b_m$ at the time $t_n$. Let us define the coefficient vector at a given time or a given node,

$$\varphi_{n,*} := (\varphi_{n,m})_{m=1}^M \in \mathbb{C}^M, \quad \varphi_{*,m} := (\varphi_{n,m})_{n=0}^N \in \mathbb{C}^{N+1},$$ (5.25)

and the matrix containing all values

$$\varphi_{*,*} := (\varphi_{n,m})_{0 \leq n \leq N \atop 1 \leq m \leq M} \in \mathbb{C}^{(N+1) \times M}.$$ (5.26)

In a first step, we discretised the continuous equation

$$V \varphi = g \quad \text{on } \Gamma \times (0, T)$$ (5.27)
in time and obtained (cf. (5.4))

\[(V \varphi)_n := \sum_{j=0}^{n} \int_{\Gamma} \omega_{n-j}^\Delta(||x-y||)\varphi_{j}^\Delta(y)d\Gamma_y, \quad n = 0, \ldots, N.\]  

By replacing \(\varphi_{j}^\Delta(y)\) by the ansatz \(\varphi_{j}^{\Delta, h}(y)\) results

\[
\sum_{j=0}^{n} \int_{\Gamma} \omega_{n-j}^\Delta(||x-y||) \sum_{m=1}^{M} \varphi_{j,m} b_{m}(y)d\Gamma_y = \sum_{j=0}^{n} \sum_{m=1}^{M} \int_{\Gamma} \omega_{n-j}^\Delta(||x-y||)\varphi_{j,m} b_{m}(y)d\Gamma_y = g(x, t_n),
\]

for all \(x \in \Gamma\) and \(0 \leq n \leq N\), which is in general an overdetermined system if being considered pointwise. To obtain a proper formulation (Galerkin discretisation) we impose it in a weak form (cf. Section 3.5). For each time step \(n\) we have to determine \(M\) coefficients. Therefore \(M\) equations have to be obtained and that is made by imposing the weak formulation on \(S\) itself.

\[
\int_{\Gamma} \left( \sum_{j=0}^{n} \sum_{m=1}^{M} \int_{\Gamma} \omega_{n-j}^\Delta(||x-y||)\varphi_{j,m} b_{m}(y)d\Gamma_y \right)\psi(x)d\Gamma_x = \int_{\Gamma} g(x, t_n)\psi(x)d\Gamma_x \quad \forall \psi \in S.
\]

As \(S\) is the span of the basis functions \(b_1, \ldots, b_M\), it suffices to impose (5.30) just for the basis functions. Furthermore the outer integration can be interchanged with the two sums, as they are finite. In general, let \(g_n^\Delta(x)\) equal \(g(x, t_n)\) or be a sufficiently accurate approximation to it.

Thus we arrive at the space and time discrete problem using convolution quadrature in time and a Galerkin discretisation in space

\[
\sum_{j=0}^{n} \sum_{m=1}^{M} \varphi_{j,m} \int_{\Gamma} \omega_{n-j}^\Delta(||x-y||)b_{m}(y)b_{k}(x)d\Gamma_yd\Gamma_x = \int_{\Gamma} g_n^\Delta(x)b_{k}(x)d\Gamma_x
\]

\[\forall 1 \leq k \leq M \text{ and } 0 \leq n \leq N.\]

Define for each \(n \in \{0, \ldots, N\}\) a right-hand side vector \(g_n \in \mathbb{R}^M\) and a system matrix \(A_n \in \mathbb{R}^{M \times M}\) by

\[(A_n)_{k,m} := \int_{\Gamma} \int_{\Gamma} \omega_{n-j}^\Delta(||x-y||)b_{m}(y)b_{k}(x)d\Gamma_yd\Gamma_x \quad \text{and} \quad (g_n)_k := \int_{\Gamma} g_n^\Delta(x)b_{k}(x)d\Gamma_x.\]

The matrices \(A_n\) are fully populated, as their entries are of the form \(\int_{\Gamma} \int_{\Gamma} \ldots\) and \(\omega_n^\Delta\) is a non-local function. Furthermore the inner sum and the condition \(\forall 1 \leq k \leq M\) of equation (5.31) can be written in vector form resulting in

\[
\sum_{j=0}^{n} A_{n-j} \varphi_{j, *} = g_n \quad \forall n \in \{0, \ldots, N\}.
\]

This can be written in matrix form leading to a lower Toeplitz system of dimension \((N+1) \times (N+1)\) with \(M \times M\) matrices as block entries. It has the following form

\[
\begin{pmatrix}
A_0 & 0 & \cdots & 0 \\
A_1 & A_0 & 0 & \cdots \\
\vdots & \ddots & \ddots & \ddots \\
A_{N-1} & \cdots & \cdots & A_0 & 0 \\
A_N & A_{N-1} & \cdots & A_1 & A_0
\end{pmatrix}
\begin{pmatrix}
\varphi_{0,*} \\
\varphi_{1,*} \\
\vdots \\
\varphi_{N-1,*} \\
\varphi_{N,*}
\end{pmatrix} =
\begin{pmatrix}
g_0 \\
g_1 \\
\vdots \\
g_{N-1} \\
g_N
\end{pmatrix}.
\]
5.3 Approximation of the Convolution Weights

Already in [Lub88], Section 7] the author listed three possibilities to calculate the convolution weights, which in fact are functions. In [HKS09] an approximation method is proposed using

cutoff on the representation \( \omega_j^{\Delta t}(d) \), where \( d := \|x - y\| \). In this master thesis

the method proposed in [BS08] is followed. It is based on the approximation of the Cauchy
integral formula by the trapezoidal rule as proposed in [Lul94] equation 3.10]. However in

[BS08] this approximation is not just used as approximation: The discrete Fourier transform
in integral formula by the trapezoidal rule as proposed in [Lub94, equation 3.10]. However in

the method proposed in [BS08] is followed. It is based on the approximation of the Cauchy

The coefficients \( \omega_j^{\Delta t}(d) \) are defined as the coefficients of the power series of \( \hat{k}(\frac{\gamma(\zeta)}{\Delta t}) \) (cf. Section

3.6] Equation 3.83) . Hence we have to determine the power series of \( \hat{k}(d, \frac{\gamma(\zeta)}{\Delta t}) \) with respect
to \( \zeta \), where \( \hat{k} \) is the Laplace transform of the fundamental solution of the wave equation \( k \).

By using the Cauchy integral formula for higher derivatives, we get for \( j \in \mathbb{N} \)

\[
\omega_j^{\Delta t}(d) = \frac{1}{2\pi i} \int_{|\zeta| = \lambda} \hat{k}(d, \frac{\gamma(\zeta)}{\Delta t}) \frac{\zeta^{j+1}}{\zeta+1} \, d\zeta,
\]

(5.35)

for \( 0 < \lambda < 1 \) small enough, such that \( \hat{k}(d, \frac{\gamma(\zeta)}{\Delta t}) \) is holomorphic on \( B_{\lambda+\varepsilon}(0) \), for some
sufficiently small \( \varepsilon > 0 \). Let us do the approximation using the trapezoidal rule and integrating
over the closed curve \( \alpha : [0, 1] \rightarrow \mathbb{C} \), with \( \alpha(t) = \lambda e^{2\pi it} \)

\[
\omega_j^{\Delta t}(d) = \frac{1}{2\pi i} \int_{\alpha} \hat{k}(d, \frac{\gamma(\zeta)}{\Delta t}) \zeta^{j+1} \, d\zeta = \frac{1}{2\pi i} \int_0^1 \hat{k}(d, \frac{\gamma(\lambda e^{2\pi it})}{\Delta t}) \cdot 2\pi i \cdot \lambda e^{2\pi id} \, d\zeta
\]

(5.36)

\[
= \int_0^1 \hat{k}(d, \frac{\gamma(\lambda e^{2\pi it})}{\Delta t}) \cdot \lambda^{-j} e^{-2\pi ij\zeta} \, d\zeta
\]

(5.37)

\[
\approx \frac{\lambda^{-j}}{N+1} \sum_{l=0}^N \hat{k} \left( d, \frac{\gamma(\lambda e^{2\pi il\zeta})}{\Delta t} \right) e^{-2\pi il\zeta} = \frac{\lambda^{-j}}{N+1} \sum_{l=0}^N \hat{k} \left( d, \frac{\gamma(\lambda e^{2\pi il})}{\Delta t} \right) e^{2\pi il},
\]

(5.38)

where in the last step \( 0 \leq l \leq N \) is replaced by \( k = N+1-l \). Therefore it holds \( 1 \leq k \leq N+1 \).

By virtue of \( e^{-2\pi i(N+1)} = 1 = e^{-2\pi i0} \) in the case of \( k = N+1 \), we can go back to \( 0 \leq k \leq N \)
and substituting just the letter \( k \) by \( l \).

For the following we introduce some further notation. The quadrature points on the unit
sphere in \( \mathbb{C} \) are denoted by \( \zeta_{N+1} := \frac{\gamma(\lambda e^{2\pi i})}{\Delta t} \). For the second argument in the Laplace transform
\( \hat{k} \) the values \( \kappa_l := i \frac{\gamma(\lambda e^{2\pi il})}{\Delta t} \) are defined, where once more the transformation \( k = is \) is used.

Furthermore recall the expression of \( \hat{k} \)

\[
\hat{k}(d, s) = \frac{e^{-sd}}{4\pi d},
\]

(5.39)
CHAPTER 5. NUMERICAL DISCRETISATION

With these notation at hand we can write

$$\omega_j^\Delta t(d) \approx \frac{\lambda^{-j}}{N+1} \sum_{l=0}^{N} k \left( d, \frac{\gamma (\lambda_{N+1})}{\Delta t} \right) \zeta^{-lj}_{N+1} = \frac{\lambda^{-j}}{N+1} \sum_{l=0}^{N} k \left( d, \frac{\kappa_l}{\Delta t} \right) \zeta^{-lj}_{N+1}$$

(5.40)

$$= \frac{\lambda^{-j}}{N+1} \sum_{l=0}^{N} e^{-\frac{\pi d}{4\pi d}} \zeta^{-lj}_{N+1} = \frac{\lambda^{-j}}{N+1} \sum_{l=0}^{N} e^{\kappa l d} e^{-lj}.$$  

(5.41)

Let us denote by $G_n(\cdot)$ the fundamental solution of the Helmholtz operator $-\Delta - \kappa^2$ as in Section 4.5, i.e.,

$$G_n(d) = \frac{e^{i\pi d}}{4\pi d}$$

(5.42)

With this notation, we finally get the approximation

$$\omega_j^\Delta t(d) \approx \frac{\lambda^{-j}}{N+1} \sum_{l=0}^{N} G_{\kappa_l}(d) \zeta^{-lj}_{N+1}.$$  

(5.43)

A proposition concerning the quality of that approximation and its compatibility to the discretisation developed in Section 5.2.4 will follow in Section 6.2.

5.4 Decoupling the Systems

Using the results of the last section we are able to rewrite the discrete problem (5.31). As we made a further approximation by employing the trapezoidal rule for approximating the Cauchy integral, let us denote the coefficients by $\tilde{\varphi}_{j,m}$ instead of the $\varphi_{j,m}$ in the case of the unperturbed discrete problem (5.31), for $0 \leq j \leq N, 1 \leq m \leq M$. Let us furthermore extend the definition of the convolution weights $\omega_n^\Delta t$ to negative indices $n = -N, -N + 1, \ldots, -1$ by setting $\omega_n^\Delta t = 0$ for such $n$. Therefore, we can enlarge the first sum in (5.31) up to $N$ for each time step, thus

$$\sum_{j=0}^{N} \sum_{m=1}^{M} \tilde{\varphi}_{j,m} \int_{\Gamma} \int \omega_{n-j}^\Delta t(||x - y||) b_m(y) b_k(x) d\Gamma_y d\Gamma_x = \int_{\Gamma} g_{n}^\Delta t(x) b_k(x) d\Gamma_x$$

(5.44)

\forall 1 \leq k \leq M and 0 \leq n \leq N.

Now the convolution weights are replaced by the approximation deduced in the last section and transformed coefficients $\hat{\varphi}_{l,m} := \sum_{j=0}^{N} \lambda^j \tilde{\varphi}_{j,m} \zeta_{N+1}^j$ are introduced.

$$\sum_{j=0}^{N} \sum_{m=1}^{M} \tilde{\varphi}_{j,m} \int_{\Gamma} \int \omega_{n-j}^\Delta t(||x - y||) b_m(y) b_k(x) d\Gamma_y d\Gamma_x = \int_{\Gamma} g_{n}^\Delta t(x) b_k(x) d\Gamma_x$$

(5.45)

$$= \frac{\lambda^{-n}}{N+1} \sum_{l=0}^{N} M \left( \sum_{m=1}^{M} \lambda^l \tilde{\varphi}_{j,m} \zeta_{N+1}^j \right) \int_{\Gamma} \int G_{\kappa_l}(||x - y||) b_m(y) b_k(x) d\Gamma_y d\Gamma_x$$

(5.46)

$$= \frac{\lambda^{-n}}{N+1} \sum_{l=0}^{N} \left( \sum_{m=1}^{M} \tilde{\varphi}_{l,m} \int_{\Gamma} \int G_{\kappa_l}(||x - y||) b_m(y) b_k(x) d\Gamma_y d\Gamma_x \right) \zeta_{N+1}^l$$

(5.47)

That leads to the equation

$$\frac{\lambda^{-n}}{N+1} \sum_{l=0}^{N} \left( \sum_{m=1}^{M} \tilde{\varphi}_{l,m} \int_{\Gamma} \int G_{\kappa_l}(||x - y||) b_m(y) b_k(x) d\Gamma_y d\Gamma_x \right) \zeta_{N+1}^l = \int_{\Gamma} g_{n}^\Delta t(x) b_k(x) d\Gamma_x$$

(5.48)

\forall 1 \leq k \leq M and 0 \leq n \leq N.
By looking at the terms one recognizes a scaled discrete Fourier transform as considered in Section 5.4. Let us define the following terms for $k = 1, \ldots, M$

\[
\hat{a}_{l,k} := \sum_{m=1}^{M} \hat{\varphi}_{l,m} \int_{\Gamma} G_{\kappa_l}(||x-y||) b_m(y)b_k(x)d\Gamma_y d\Gamma_x, \quad l = 0, \ldots, N \tag{5.49}
\]

\[
a_{n,k} := \frac{\lambda^{-n}}{N+1} \sum_{l=0}^{N} \left\{ \sum_{m=1}^{M} \hat{\varphi}_{l,m} \int_{\Gamma} G_{\kappa_l}(||x-y||) b_m(y)b_k(x)d\Gamma_y d\Gamma_x \right\} \zeta_{N+1}^{-ln}, \quad n = 0, \ldots, N. \tag{5.50}
\]

This definition makes sense as they are coefficients of two vectors linked by the scaled discrete Fourier transform. Therefore equation (5.48) can be written as

\[
a_{n,k} = \int_{\Gamma} \hat{g}_{n}^{\Delta t}(x)b_k(x)d\Gamma_x \quad \forall 1 \leq k \leq M \text{ and } 0 \leq n \leq N. \tag{5.51}
\]

Employing the inverse scaled Fourier transform results in the equivalent equation

\[
\hat{a}_{l,k} = \sum_{j=0}^{N} \lambda^j a_{j,k} \zeta_{N+1}^{lj} = \sum_{j=0}^{N} \lambda^j \int_{\Gamma} \hat{g}_{j}^{\Delta t}(x,b_k(x)d\Gamma_y \zeta_{N+1}^{lj} \tag{5.52}
\]

\[
= \int_{\Gamma} \left( \sum_{j=0}^{N} \lambda^j \hat{g}_{j}^{\Delta t}(x) \zeta_{N+1}^{lj} \right) b_k(x)d\Gamma_x \quad \forall 1 \leq k \leq M \text{ and } 0 \leq l \leq N. \tag{5.53}
\]

By defining $\hat{g}_l^{\Delta t}(x) := \sum_{j=0}^{N} \lambda^j g_j^{\Delta t}(x) \zeta_{N+1}^{lj}$ for $0 \leq l \leq N$, we get the equations in their final form

\[
\sum_{m=1}^{M} \hat{\varphi}_{l,m} \int_{\Gamma} G_{\kappa_l}(||x-y||) b_m(y)b_k(x)d\Gamma_y d\Gamma_x = \int_{\Gamma} \hat{g}_l^{\Delta t}(x)b_k(x)d\Gamma_x, \quad \forall 1 \leq k \leq M \text{ and } 0 \leq l \leq N. \tag{5.54}
\]

Therefore, for each time step $0 \leq l \leq N$ a single Helmholtz problem must be solved. Let us define the system matrix $\hat{\mathbf{A}}_l \in \mathbb{C}^{M \times M}$ and the right-hand side vector $\hat{\mathbf{g}}_l$ for $0 \leq l \leq N$ by

\[
(\hat{\mathbf{A}}_l)_{k,m} = \int_{\Gamma} \int_{\Gamma} G_{\kappa_l}(||x-y||) b_m(y)b_k(x)d\Gamma_y d\Gamma_x \quad \text{and} \quad (\hat{\mathbf{g}}_l)_k = \int_{\Gamma} \hat{g}_l^{\Delta t}(x)b_k(x)d\Gamma_x. \tag{5.55}
\]

With this notation we get the $N + 1$ decoupled linear systems for $l = 0, \ldots, N$

\[
\hat{\mathbf{A}}_l \hat{\varphi}_{l,*} = \hat{\mathbf{g}}_l, \tag{5.56}
\]

each of which represents a Helmholtz problem with the complex wavenumber $\kappa_l$. By solving these systems, a coefficient matrix $\hat{\varphi}_{*,*}$ is deduced. As we are interested in $\hat{\varphi}_{*,*}$, the obtained FFT-transformed solution vector has to be transformed back by using the inverse of the scaled discrete Fourier transform. Thus for the vector $\hat{\varphi}_{*,m}$ it holds

\[
\hat{\varphi}_{n,m} = \frac{\lambda^{-n}}{N+1} \sum_{l=0}^{N} \hat{\varphi}_{l,m} \zeta_{N+1}^{-ln} \quad \forall 0 \leq n \leq N \text{ and } 1 \leq m \leq M. \tag{5.57}
\]

The last question concerns the wavenumbers $\kappa_l$. As encountered in Section 4.3 there are critical values for wavenumbers, where the single layer potential is not invertible. As we have
seen it is sufficient, that all $\kappa_l$ have positive imaginary part $\Im(\kappa_l) > 0$, $\forall l \in \{0, \ldots, N\}$. Let $l \in \{0, \ldots, N\}$

$$\Im(\kappa_l) = \Im \left( i \frac{\gamma \zeta_N^l}{\Delta t} \right) = \frac{1}{\Delta t} \Re \left( \frac{1}{2} \lambda^2 e^{-\frac{4\pi il}{N+1}} - 2\lambda e^{-\frac{2\pi il}{N+1}} + \frac{3}{2} \right)$$

(5.58)

$$= \frac{1}{\Delta t} \left( \frac{1}{2} \lambda^2 \cos \left( \frac{4\pi il}{N+1} \right) - 2\lambda \cos \left( \frac{2\pi il}{N+1} \right) + \frac{3}{2} \right).$$

(5.59)

As $0 \leq \frac{2\pi il}{N+1} \leq \frac{4\pi il}{N+1} < 4\pi$ for $l \in \{0, \ldots, N\}$ we discuss the function

$$f(x) := \frac{1}{2} \lambda^2 \cos 2x - 2\lambda \cos x + \frac{3}{2}, \quad x \in [0, 4\pi].$$

(5.60)

By virtue of the addition theorem for $\cos$ we get

$$f(x) = \lambda^2 \cos^2 x - 2\lambda \cos x + \left( \frac{3}{2} - \frac{1}{2} \lambda^2 \right).$$

(5.61)

The substitution $z = \cos x$ leads to a quadratical polynomial, whose zeros are examined in the following. These can be calculated on the usual way

$$z_{1,2} = \frac{1 \pm \sqrt{\lambda^2 - 1}}{\lambda}.$$  

(5.62)

As $0 < \lambda < 1$ both $z_1, z_2$ are complex, i.e. $\Im(z_1), \Im(z_2) \neq 0$. Therefore $f$ does not have any zeros on $[0, 4\pi]$. Furthermore $f$ is continuous and $f(\pi) = \frac{1}{2} \lambda^2 + 2\lambda + \frac{3}{2} > 0$, and therefore $f > 0$ on $[0, 4\pi]$. That gives us the sufficient condition, that $\Im(\kappa_l) > 0$ for each $0 \leq l \leq N$.

![Figure 5.3: Frequencies $\kappa_l$ for $\lambda = \Delta t^{2.5/N}$ with $T = 1.9$, $N = 127$](image1)

![Figure 5.4: Squared frequencies $\kappa_l^2$ for $\lambda = \Delta t^{2.5/N}$ with $T = 1.9$, $N = 127$ (blue); range of possible critical values (red)](image2)

### 5.5 Approximation of the Wave Function

As we are finally interested in the wave function $u$ and not in the density $\varphi$, we have to calculate $u$ using the discrete density $\tilde{\varphi}(x)$. Let $x \in \Omega$ be a given point in the exterior domain
5.5. APPROXIMATION OF THE WAVE FUNCTION

Ω. Using our ansatz we get

$$u(x, t) = \int_0^t \int \frac{\delta(\|x - y\| - (t - \tau))}{4\pi\|x - y\|} \varphi(y, \tau)d\Gamma_y d\tau$$

$$= \int \frac{1}{4\pi\|x - y\|} \varphi(y, t - \|x - y\|)d\Gamma_y$$

$$= \sum_{\tau \in G} \int \frac{1}{4\pi\|x - y\|} \varphi(y, t - \|x - y\|)d\Gamma_y.$$  (5.63)

For the time \( t \) we use the equidistant discrete times arising in the boundary element method \( t_l, 0 \leq l \leq N \). Suppose that \( b_m, 1 \leq m \leq M \) is a basis of the approximating space \( S \) and \( \tilde{\varphi}_{n,m} \) are the coefficients of the numerical approximation of \( \varphi \), i.e. \( \tilde{\varphi}(x, t_n) = \sum_{m=1}^M \tilde{\varphi}_{n,m} b_m(x) \), hence

$$u(x, t_l) = \sum_{\tau \in G} \int \frac{1}{4\pi\|x - y\|} \varphi(y, t_l - \|x - y\|)d\Gamma_y.$$  (5.64)

As the retarded time \( t^* := t_l - \|x - y\| \) in general does not coincide with a discrete time point, we use a linear interpolation in time to evaluate \( \varphi \) at \( t^* \). With \( n := \left\lfloor \frac{t^*}{\Delta t} \right\rfloor \), it holds \( t^* \in [n\Delta t, (n+1)\Delta t] \). Define \( t^- := n\Delta t \) and \( t^+ := (n+1)\Delta t \). Using linear interpolation we get

$$\varphi(y, t_l - \|x - y\|) \approx \varphi(y, t^-) + (\varphi(y, t^+) - \varphi(y, t^-)) \frac{t^* - t^-}{t^+ - t^-}$$

$$= \varphi(y, t^-) + (\varphi(y, t^+) - \varphi(y, t^-)) \frac{t^* - n\Delta t}{\Delta t}$$

$$= \varphi(y, t^-) \left( 1 - \left( \frac{t^*}{\Delta t} - n \right) \right) + \varphi(y, t^+) \left( \frac{t^*}{\Delta t} - n \right)$$

$$= \sum_{m=1}^M \left( \tilde{\varphi}_{n,m} \left( 1 - \left( \frac{t^*}{\Delta t} - n \right) \right) + \tilde{\varphi}_{n+1,m} \left( \frac{t^*}{\Delta t} - n \right) \right) \cdot b_m(y).$$  (5.65)

Merging these approximations leads to

$$u(x, t_l) \approx \sum_{m=1}^M \left[ \left( \tilde{\varphi}_{n,m} \left( 1 - \left( \frac{t^*}{\Delta t} - n \right) \right) + \tilde{\varphi}_{n+1,m} \left( \frac{t^*}{\Delta t} - n \right) \right) \sum_{\tau \in G} \int \frac{1}{4\pi\|x - y\|} b_m(y)d\Gamma_y \right]$$

$$= \sum_{m=1}^M \left[ \left( \tilde{\varphi}_{n,m} \left( 1 - \left( \frac{t^*}{\Delta t} - n \right) \right) + \tilde{\varphi}_{n+1,m} \left( \frac{t^*}{\Delta t} - n \right) \right) \int_{\text{supp} b_m} \frac{b_m(y)}{4\pi\|x - y\|} d\Gamma_y \right].$$  (5.66)
Chapter 6

Theory for the Discretised Problem

After the formulation of the time and space discretisation in the last chapter, the convergence of the method is examined. First of all, a discussion of the different discretisation schemes will be presented for finally merging the results to a theorem proving the convergence of the entire method.

6.1 Boundary Element Space

For the space discretisation we use the boundary element spaces \( S_{-1,0} \) and \( S_{0,1} \) defined in Section 5.2. As the exact solution \( \varphi(x,t) \) exists in \( H^{-1/2}(\Gamma) \) with respect to \( x \) (cf. Theorem 4.7.1), we are interested in the accuracy of the approximation employing these boundary element spaces. The appropriate result is presented in [DL90 Chapter 8] and adapted by [Lub94 Chapter 5.2].

**Theorem 6.1.1.** Let \( G \) be a regular paneling with mesh width \( h \) and \( S_{m-1,m} \) the space of piecewise constant resp. piecewise linear functions on \( G \) for \( m = 0 \) resp. \( m = 1 \). Then it holds

\[
\begin{align*}
(i) \quad & S_{m-1,m} \subset L^2(\Gamma) \\
(ii) \quad & \inf_{\varphi_S \in S_{m-1,m}} \| \varphi_S - \varphi \|_{H^{-1/2}(\Gamma)} \leq C \cdot h^{m+\frac{3}{2}} \cdot \| \varphi \|_{H^{m+1}(\Gamma)} \quad \forall \varphi \in H^{m+1}(\Gamma).
\end{align*}
\]

In order to validate the MATLAB implementation (cf. Chapter 8) an approximation of the \( L^2(\Gamma) \)-error is calculated, so that the convergence rate in the \( L^2(\Gamma) \)-norm should also be known. The result of Theorem 6.1.1 is obtained by employing a duality argument onto the error in the \( L^2(\Gamma) \)-norm, whereby the convergence rate is increased by 1/2. Let us resume these results in a short table.

<table>
<thead>
<tr>
<th>( m = 0 )</th>
<th>( H^{-1/2}(\Gamma) )-norm</th>
<th>( L^2(\Gamma) )-norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m = 1 )</td>
<td>( h^{5/2} )</td>
<td>( h^2 )</td>
</tr>
</tbody>
</table>

Table 6.1: Convergence rate for \( S_{-1,0} \) and \( S_{0,1} \) in different norms by assuming full regularity

6.2 Approximation of the Convolution Weights

In Chapter 5.3 the convolution weights \( \omega_j(d) \) are approximated using the Cauchy integral formula for higher derivatives. As parameter for the accuracy of the approximation the real number \( \lambda \) is introduced. Depending on that parameter the following proposition is shown in [BS08 Proposition 4.1].
Proposition 6.2.1. Let $N \in \mathbb{N}$, $d > 0$, $\Delta t = \frac{T}{N}$ and $\lambda < e^{-\Delta t}$ be given. There exists a constant $C > 0$ independent of the parameters, such that

$$
|\omega_j^N(d) - \frac{\lambda^{-j}}{N+1} \sum_{l=0}^{N} G_{\kappa_l}(d) \zeta_{N+1}^l| \leq C e^{2T} \frac{\lambda^{N+1}}{d}, \quad \text{for } j = -N, -N + 1, \ldots, N. \tag{6.1}
$$

The proof is based on the theory of discrete Fourier transform, mainly on the 'aliasing formula' [BS08 Theorem A.1], which gives a connection between the exact and the discrete Fourier coefficient.

6.3 Convergence of the Method

First of all, we consider the semi-discrete problem formulated in (5.4) using the BDF2 scheme. In [Lub94] Chapter 3 the author develops error estimates for the convolution quadrature applied to convolutions of the form $\hat{k}(\partial_t) g$. Starting with pointwise error bounds, using just the properties of the linear multistep method and the bound (4.63) for the Laplace transform $\hat{k}$, the theory leads to $l^2$-error bounds. In [Lub94] Chapter 4 the theory is applied to the wave equation using the appropriate bound for $\mathcal{V}^{-1}(s)$. [Lub94] Theorem 5.1 gives us the result for an $A$-stable linear multistep method of order 2, that fulfills furthermore an additional condition ([Lub94 Equation (3.11)]). As BDF2 has all these properties (cf. Section 3.7), we can state the above cited theorem in the case using BDF2.

Theorem 6.3.1. Let the BDF2 method be employed for the time discretization. Consider the problem on $[0, T]$ with equidistant discrete times $n\Delta t$ where $\Delta t = \frac{T}{N}$ for $N \in \mathbb{N}$. For $g \in H_0^5(0, T; H^{1/2}(\Gamma))$ the solution of the semi-discrete problem (5.4) denoted by $\varphi_n^\Delta t$ satisfies

$$
\left( \Delta t \sum_{n=0}^{N} \|\varphi_n^\Delta t(\cdot) - \varphi(\cdot, n\Delta t)\|^2_{H^{-1/2}(\Gamma)} \right) \leq C_{\Delta t_0} \cdot \Delta t^2 \cdot \|g\|_{H_0^5(0, T; H^{1/2}(\Gamma))} \tag{6.2}
$$

uniformly for $0 < \Delta t \leq \Delta t_0$ and a constant $C_{\Delta t_0}$ independent of $\Delta t$ and $g$.

As next step, in [Lub94] Chapter 5.2 the error caused by the discretisation in space by the Galerkin method is analysed. The result is a stability assertion and an error bound based Theorem 5.1.1. The final step in [Lub94] is merging the results, to get the convergence rate of the fully discretised problem (5.31). This is stated in [Lub94] Theorem 5.4 and is formulated in [HKS09] Theorem 4.2 especially for the case of BDF2.

Theorem 6.3.2. For smooth compatible data $g$, i.e. $g \in H_0^5(0, T; H^{1/2}(\Gamma))$, the fully discrete method defined by (5.31), i.e. Galerkin discretisation in space and operational convolution quadrature in time, is unconditionally stable and provides a unique solution $\varphi_n^{\Delta t, h} \in S_{m-1, m}$ for $m \in \{0, 1\}$ and $0 \leq n \leq N$, which satisfies the error estimate

$$
\|\varphi(\cdot, t_n) - \varphi_n^{\Delta t, h}(\cdot)\|_{H^{-1/2}(\Gamma)} \leq C_g (\Delta t^2 + h^{m+\frac{3}{2}}). \tag{6.3}
$$

In [HKS09] the authors continue to extend the convergence theory for perturbations in space discretization due to the approximation of the convolution weights and panel clustering. The idea is to replace the weights for certain parameters $d$ by zero. The condition for that is

$$
|\omega_n^\Delta t(d)| \leq \frac{\varepsilon}{4\pi d}, \tag{6.4}
$$

with $\varepsilon > 0$ the cutoff parameter. The convergence theory is developed in [HKS09] Chapter 4.2 and is once more achieved by examining the issue in the Laplace transformed case. The
6.3. CONVERGENCE OF THE METHOD

The final result (cf. [HKS09, Corollary 4.8]) is to choose $\varepsilon \sim h^{7m/2+25/4}$ to retain the original convergence rate.

The approximation of $\omega_n^{\Delta t}$, developed in Section 5.3 and analysed in Section 6.2, is presented in [BS08]. There the convergence theory is based on the result provided by 6.3.2 but with a slightly modified proof, and the error estimate given by Proposition 6.2.1. The accuracy parameter is $\lambda$ instead of $\varepsilon$. Putting all these results together allows to proof the main result [BS08, Theorem 5.1].

**Theorem 6.3.3.** Let the exact solution $\varphi(\cdot, t)$ be in $H^{m+1}(\Gamma)$ for any $t \in [0, T]$, the data $g \in H_0^2(0, T; H^{1/2}(\Gamma))$ and the boundary element space $S = S_{m-1,m}$ for $m \in \{0, 1\}$. For a given $\sigma > 0$ and for $\lambda < 1$ such that $\lambda^{N+1} \leq C_\sigma e^{-(2+\sigma)T}(1 - e^{-\sigma \Delta t})h \Delta t^2$, the solution $\varphi_n^{\Delta t,h}$ for the problem (5.54) exists and satisfies the error estimate

$$\|\varphi_n^{\Delta t,h} - \varphi(\cdot, t_n)\|_{H^{-1/2}(\Gamma)} \leq C_g(t_n)(\lambda^{N+1} e^{2T} h^{-1} \Delta t^{-5} + \Delta t^2 + h^{m+3/2}),$$

where $C_\sigma$ depends on $\sigma$ and $C_g$ depends on $\sigma$ and the right-hand side $g$.

To resume this result in a short, recipe-style way, we state

**Corollary 6.3.4.** Assume the same conditions of the last theorem and let

$$h^{m+3/2} \lesssim \Delta t^2, \quad \lambda^{N+1} \sim h^{7m/2+25/4},$$

then the optimal rate of convergence is achieved,

$$\|\varphi_n^{\Delta t,h} - \varphi(\cdot, t_n)\|_{H^{-1/2}(\Gamma)} \leq C \Delta t^2,$$

with $C$ depending on $g$. 

Chapter 7

MATLAB Implementation

In this chapter a MATLAB implementation is presented, that can be found in [Hub10]. Consider the exterior Dirichlet problem given by $P$ in Chapter 2 in the case of three dimensional space, using the discretisation developed in Chapter 5. Assume the following details:

(i) Let $\mathcal{G}$ be a regular triangulation of $\Gamma$ consisting of $N_\tau \in \mathbb{N}$ triangles denoted by $\tau_i$, $i = 1, \ldots, N_\tau$. Let $N_P$ be the number of vertices denoted by $P_i \in \mathbb{R}^3$, $i = 1, \ldots, N_P$. Analogously let us define $N_E$ as the number of edges denoted by $E_i$, $i = 1, \ldots, N_E$.

(ii) Let $T > 0$ a finite time, discretised in $N \in \mathbb{N}$ time steps with a distance of $\Delta t$.

\[ N_\tau, N_P, N_E \text{ are of the same order. For } N_\tau \text{ even it holds } N_V = \frac{1}{2} N_\tau + 2 \quad N_E = \frac{3}{2} N_\tau. \] (7.1)

Therefore, if a data structure has storage costs of order $O(N_\tau)$, $O(N_V)$ or $O(N_E)$ it can be seen as order $O(M)$ independent of the choice $M = N_\tau$ or order $M = N_V$.

7.1 Algorithm

First, we will explain the principal structure of the algorithm. It will become clear that the decoupling of the Toeplitz system to decoupled Helmholtz problems, as deduced in Section 5.4, is a huge advantage. Our MATLAB implementation is adapted to parallel computer architecture as follows: The program code is run on several started MATLAB sessions. One of them is the 'Master' session, the other ones are 'Slaves'. The main routine is run on the 'Master' session in contrast to the 'Slave' sessions, where a slave routine of the 'Multicore' package\(^1\) provided by Markus Buehren is started. After the preprocessing, which provides

\(^1\)The package can be found on the homepage of MATLAB Central \url{http://www.mathworks.com/matlabcentral/fileexchange/13775} (13.10.2010)
the necessary info for solving the different Helmholtz problems, the 'Multicore' package distributes the different decoupled Helmholtz problems to the different 'Slave' sessions. This happens by saving for each single Helmholtz problem a file, including all needed parameters and data into a defined directory. In that way, as many Helmholtz problems, as MATLAB sessions are opened, are solved parallelly. The calculated solution again is saved into that directory, from where the 'Master' session collects them and puts them together, as if all the work has been done by the 'Master' session.

![Flow Chart](image)

Figure 7.2: Schematic application flow

To illustrate the structure, a flow chart with the most important points and the responsible
MATLAB routines is presented in Figure 7.2. In Section 7.3 later in this Chapter, each routine is described in details.

7.2 Generating a Mesh

As the whole method is based on a surface paneling of the boundary $\Gamma$, one essential part is to get such data and integrate it into the implementation. The tool used in this project is Gmsh\textsuperscript{2}. It is a 3D finite element grid generator equipped with a CAD engine. Furthermore it allows doing finite element calculations, like building system matrices. However, as we are using the boundary element method, only the meshing routine for surfaces is used. More detailed information is provided in [GR09] and the entire documentation can be found on the project homepage (cf. Footnote 2 on Page 53). One important feature is the possibility of defining the geometry in a script. In that way even more complicated objects can be constructed exactly and quite easily. Furthermore, for each vertex a characteristic length can be fixed, that will define the fineness of the mesh around that vertex, cf. Figure 7.3.

Figure 7.3: Surface mesh of the unit sphere produced by Gmsh

Generated meshes can be saved as .msh-file. To import it into the data structure described in Section 7.3, a MATLAB routine is written, called 'readMeshFile.m' (the description follows in Section 7.5).

7.3 Data Structures for Mesh Handling and Global Parameters

The structure called mesh contains all information about the given triangulation $\mathcal{G}$. In the following list the fields of this structure and their link to the mathematical problem and notation are stated. If their storage costs depend on $M$ or $N$, the complexity in dependence of $M$ and $N$ are listed.

(i) $nN$: Single variable containing the number of unique vertices.

$$nN \in \mathbb{N} \quad \text{with} \quad nN = N_P$$

(ii) node: A $nN \times 3$ matrix with the coordinates of the vertices saved in its rows. Storage costs of order $O(M)$.

$$\text{node} \in \mathbb{R}^{N_P \times 3} \quad \text{with} \quad (\text{node})_{i,j} = (P_i)_j, \quad 1 \leq i \leq N_P, \quad j \in \{1, 2, 3\}$$

\textsuperscript{2}The software and documentation can be found on http://geuz.org/gmsh (33.10.2010)
(iii) \( nT \): Single variable containing the number of unique triangles.

\[ nT \in \mathbb{N} \quad \text{with} \quad nT = N_T \quad (7.4) \]

(iv) \( \text{triangle} \): A \( nT \times 3 \) matrix with indices of the vertices saved in its rows. Storage costs of order \( \mathcal{O}(M) \).

\[ \text{triangle} \in \mathbb{N}^{N_T \times 3} \quad \text{with} \]

\[ \tau_i = \text{conv} \{ \mathbf{P}(\text{triangle})_{i,1}, \mathbf{P}(\text{triangle})_{i,2}, \mathbf{P}(\text{triangle})_{i,3} \} \quad (7.5) \]

(v) \( nE \): Single variable containing the number of unique edges.

\[ nE \in \mathbb{N} \quad \text{with} \quad nE = N_E \quad (7.7) \]

(vi) \( \text{edgeind} \): A \( nE \times 2 \) matrix with indices of the starting and end vertex of a unique edge. Storage costs of order \( \mathcal{O}(M) \).

\[ \text{edgeind} \in \mathbb{N}^{N_E \times 2} \quad (7.8) \]

For a \( i \in \{1, \ldots, N_E\} \), assume that \( \mathbf{E}_i \) is the \( j \)-th edge of a triangle \( \tau_k \), \( 1 \leq j \leq 3 \), \( 1 \leq k \leq N_T \). Then it holds

\[ j = 1 : \{(\text{edgeind})_{i,l} | l = 1, 2\} = \{(\text{triangle})_{k,2}, (\text{triangle})_{k,3}\} \quad (7.9) \]

\[ j = 2 : \{(\text{edgeind})_{i,l} | l = 1, 2\} = \{(\text{triangle})_{k,3}, (\text{triangle})_{k,1}\} \quad (7.10) \]

\[ j = 3 : \{(\text{edgeind})_{i,l} | l = 1, 2\} = \{(\text{triangle})_{k,1}, (\text{triangle})_{k,2}\} \quad (7.11) \]

Furthermore

\[ (\text{edgeind})_{i,1} < (\text{edgeind})_{i,2} \quad \forall i \in \{1, \ldots, N_E\} \quad (7.12) \]

(vii) \( \text{edge} \): A \( nE \times 3 \) matrix with coordinates of the edges saved in its rows. Storage costs of order \( \mathcal{O}(M) \).

\[ \text{edge} \in \mathbb{R}^{N_E \times 3} \quad (7.13) \]

Assume that edge \( \mathbf{E}_i \) consists of vertices \( \mathbf{P}_j \) and \( \mathbf{P}_k \) with \( 1 \leq j < k \leq N_P \) and \( i \in \{1, \ldots, N_E\} \). Then

\[ (\text{edge})_{i,l} = (\mathbf{P}_k)_l - (\mathbf{P}_j)_l, \quad 1 \leq l \leq 3 \quad (7.14) \]

(viii) \( \text{triangletoedge} \): A \( nT \times 3 \) matrix with indices of the edges of a triangle. The sign of the entry shows the orientation of the edge. Storage costs of order \( \mathcal{O}(M) \).

\[ \text{triangletoedge} \in \mathbb{Z}^{N_T \times 3} \quad (7.15) \]

For \( 1 \leq i \leq N_T \) and \( 1 \leq j \leq 3 \) let \( x = (\text{triangletoedge})_{i,j} \) be an entry. Then it holds

\[ j = 1 : (\text{edge})_{x,i,*} = \begin{cases} (\text{triangle})_{i,2} & (\text{triangle})_{i,3} \quad x > 0 \\ (\text{triangle})_{i,3} & (\text{triangle})_{i,2} \quad x < 0 \end{cases} \quad (7.16) \]

\[ j = 2, 3 \quad \text{analogously to the cases in (7.10) and (7.11)} \quad (7.17) \]
7.3. DATA STRUCTURES FOR MESH HANDLING AND GLOBAL PARAMETERS

(ix) area: A nT array with the area of the triangles saved in it. Storage costs of order $O(M)$.

$$\text{area} \in \mathbb{R}^{N_T} \text{ with } (\text{area})_i = |\tau_i|, \ \forall i \in \{1, \ldots, N_T\}$$ (7.18)

Assume $A, B \in \mathbb{R}^3$ are two different edges of the triangle $\tau$, available in $\text{edge}$. Then the area is calculated by

$$|\tau| = \frac{1}{2} \|A \times B\|,$$ (7.19)

where $\|\cdot\|$ is the Euclidean norm and $\times$ the cross product in $\mathbb{R}^3$.

(x) edgelength: A nE array with Euclidean length of the edges saved in it. Storage costs of order $O(M)$.

$$\text{edgelength} \in \mathbb{R}^{N_E} \text{ with } (\text{edgelength})_i = \|E_i\|, \ \forall i \in \{1, \ldots, N_E\}$$ (7.20)

(xi) diameter: A nT array with the diameter of the triangles saved in it. Storage costs of order $O(M)$.

$$\text{diameter} \in \mathbb{R}^{N_T} \text{ with } (\text{diameter})_i = \max_{x, y \in \tau_i} \|x - y\|, \ \forall i \in \{1, \ldots, N_T\}$$ (7.21)

In the case of flat triangles with straight edges, the diameter is equal to the largest distance between two vertices of the triangle. Let $1 \leq j_1 < j_2 < j_3 \leq N_E$ be the indices of the edges of a triangle $\tau_i, i \in \{1, \ldots, N_T\}$. Then

$$(\text{diameter})_i = \max\{\|E_{j_1}\|, \|E_{j_2}\|, \|E_{j_3}\|\}$$ (7.22)

(xii) hmin, hmax: Single variables containing the minimal resp. maximal diameter of the triangulation $\mathcal{T}$. Using the considerations made for $\text{diameter}$ it holds

$$h_{\text{min}}, h_{\text{max}} \in \mathbb{R} \text{ with } h_{\text{min}} = \min_{1 \leq i \leq N_E} \|E_i\| \text{ resp. } h_{\text{max}} = \max_{1 \leq i \leq N_E} \|E_i\|$$ (7.23)

(xiii) enumeration: A nT $\times$ nT sparse matrix encoding the relation of two triangles as a flag. Storage costs of order $O(M^2)$.

Furthermore enumeration is symmetric and it holds

$$\text{enumeration} \in \mathbb{N}^{N_T \times N_T} \text{ with } 1 \leq i, j \leq N_T$$ (7.24)

$$(\text{enumeration})_{i,j} = (\text{enumeration})_{j,i} = \begin{cases} 0 & \text{the distance between } \tau_i \text{ and } \tau_j \text{ is positive} \\ 1 & \tau_i \text{ and } \tau_j \text{ have a common vertex} \\ 2 & \tau_i \text{ and } \tau_j \text{ have a common edge} \\ 3 & \tau_i \text{ and } \tau_j \text{ are identical} \end{cases}$$ (7.25)

(xiv) index: A nT $\times$ nT sparse matrix saving the local number of a common object. Storage costs of order $O(M^2)$.

$$\text{index} \in \mathbb{N}^{N_T \times N_T} \text{ with } 1 \leq i, j \leq N_T$$ (7.26)

$$(\text{index})_{i,j} = \begin{cases} 0 & (\text{enumeration})_{i,j} = 0 \\ k \in \{1, 2, 3\} & (\text{enumeration})_{i,j} = 1 \text{ and } k \text{ is the local number of the common vertex w.r.t. } \tau_i \\ k \in \{1, 2, 3\} & (\text{enumeration})_{i,j} = 2 \text{ and } k \text{ is the local number of the common edge w.r.t. } \tau_i \\ 0 & (\text{enumeration})_{i,j} = 3 \end{cases}$$ (7.27)

This matrix is not symmetric, however the positioning of non-zero entries is symmetric.
(xv) **distnode**: A \( nN \times nN \) matrix with the distance between the nodes saved in it. Storage costs of order \( \mathcal{O}(M^2) \).

\[
\text{distnode} \in \mathbb{R}^{N_P \times N_P} \quad \text{with} \quad (\text{distnode})_{i,j} = \| \mathbf{P}_i - \mathbf{P}_j \| \quad \forall i, j \in \{1, \ldots, N_P\}
\]

(7.28)

(xvi) **distance**: A \( nT \times nT \) matrix with an estimate of the distance between two triangles saved in it. Storage costs of order \( \mathcal{O}(M^2) \).

\[
\text{distance} \in \mathbb{R}^{N_\tau \times N_\tau} \quad \text{with} \quad (\text{distance})_{i,j} = \max_{x \in \tau_i, y \in \tau_j} \| x - y \| \quad \forall i, j \in \{1, \ldots, N_\tau\}
\]

(7.29)

As the right-hand side in (7.29) is relatively time consuming in the computation, we employ an upper bound instead.

\[
(\text{distance})_{i,j} = \max_{\text{A is vertex of } \tau_i, \text{B is vertex of } \tau_j} \| \mathbf{A} - \mathbf{B} \| \quad \forall i, j \in \{1, \ldots, N_\tau\}
\]

(7.30)

(xvii) **edgetotriangle**: A \( nE \times 2 \) matrix with the indices of its adjacent triangles stored in its rows. Storage costs of order \( \mathcal{O}(M) \).

\[
\text{edgetotriangle} \in \mathbb{N}^{N_E \times 2}
\]

(7.31)

Assume, that \( \mathbf{E}_i \) is a joint edge of two triangles \( \tau_j, \tau_k \), for \( i \in \{1, \ldots, N_E\} \), \( j, k \in \{1, \ldots, N_\tau\} \) and \( j \neq k \), then

\[
((\text{edgetotriangle})_{i,1}, (\text{edgetotriangle})_{i,2}) = \{j, k\}
\]

(7.32)

(xviii) **midpoint**: A \( nT \times 3 \) matrix with the coordinates of the midpoint of every triangle saved in its rows. Storage costs of order \( \mathcal{O}(M) \).

\[
\text{midpoint} \in \mathbb{R}^{N_\tau \times 3}
\]

(7.33)

Assume, that \( \mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^3 \) are the vertices of a triangle \( \tau_i \), \( i \in \{1, \ldots, N_\tau\} \). Hence

\[
(\text{midpoint})_{i,l} = \frac{1}{3} (\mathbf{A}_l + \mathbf{B}_l + \mathbf{C}_l) \quad \forall l \in \{1, 2, 3\}
\]

(7.34)

\[\rightarrow\] Overall the structure \text{mesh} uses storage of the order \( \mathcal{O}(M^2) \).

The parameters of the discretisation, which are independent of the mesh geometry, are saved in a structure called \text{param}. This structure contains

(i) **T**: A single variable that represents the final time.

\[
T \in \mathbb{R} \quad \text{with} \quad T = T
\]

(7.35)

(ii) **N, N1**: Two single variables saving the number of time steps.

\[
N, N1 \in \mathbb{N} \quad \text{with} \quad N = N, N1 = N + 1
\]

(7.36)

(iii) **dt**: A single variable that represents the length of a time step.

\[
dt \in \mathbb{R} \quad \text{with} \quad dt = \Delta t
\]

(7.37)
(iv) **filename**: A string containing the name of the mesh file generated by Gmsh.

(v) **order**: A flag saving the boundary element space.

\[
\text{order} \in \mathbb{N} \quad \text{with} \quad \text{order} = \begin{cases} 
0 & \text{approximation using piecewise constant elements} \\
1 & \text{approximation using piecewise linear elements}
\end{cases}
\]  

(7.38)

(vi) **dof**: A single variable representing the number of degree of freedom with respect to the space discretisation.

\[
dof \in \mathbb{N} \quad \text{with} \quad \text{dof} = \begin{cases} 
N_r & \text{approximation using piecewise constant elements} \\
N_P & \text{approximation using piecewise linear elements}
\end{cases}
\]  

(7.39)

(vii) **zeta**: A single complex variable used for computing the wavenumbers.

\[
\text{zeta} \in \mathbb{C} \quad \text{with} \quad \text{zeta} = \zeta_{N+1} = e^{-\frac{2\pi i}{N+1}}
\]  

(7.40)

(viii) **lambda**: A single variable representing a parameter for approximating the convolution weights used in Section 5.3

\[
\lambda \in (0, 1) \quad \text{with} \quad \lambda = \lambda
\]  

(7.41)

(ix) **kappal**: A \( \mathbb{N} \) array consisting of all wavenumbers for the decoupled systems. Storage costs of order \( \mathcal{O}(N) \).

\[
\text{kappal} \in \mathbb{C}^{N+1} \quad \text{with} \quad (\text{kappal})_{l+1} = i \frac{\gamma (\lambda \zeta_{N+1}^l)}{\Delta t}, \quad l \in \{0, \ldots, N\}
\]  

(7.42)

with \( \gamma \) the quotient of the generating polynomials of BDF2.

(x) **testCase**: A single flag for the choice of implemented test cases, which determine the right-hand side function \( g \).

\[
\text{testCase} \in \mathbb{N} \quad \text{with}
\begin{align*}
0 & \text{ test case with } e(x) = Y_0^0(x) \text{ (cf. Section 8.1)} \\
1 & \text{ test case with } e(x) = Y_1^1(x) \text{ (cf. Section 8.1)} \\
2 & \text{ simple version of a plane wave (cf. Section 8.5)} \\
3 & \text{ extended version of a plane wave (cf. Section 8.5)} \\
4 & \text{ spherical wave (cf. Section 8.5)}
\end{align*}
\]  

(7.43)

(xi) **intlogh**: A single flag for the choice of the number of Gauss points for the numerical quadrature.

\[
\text{intlogh} \in \{\text{true}, \text{false}\} \quad \text{with} \quad \text{intlogh} = \begin{cases} 
\text{true} & \text{using the rules of Section 7.4} \\
\text{false} & \text{use the same number of Gauss points for all combinations}
\end{cases}
\]  

(7.44)

\(---> \) Overall the structure **param** uses storage of the order \( \mathcal{O}(N) \).

Both described structures are declared as global variables. Therefore they will not be part of any input or output of the routines described in Section 7.5.
7.4 Numerical Integration

The numerical integration for calculating the system matrices and the right-hand side vector for the decoupled systems (5.56) is based on the integration routines developed in [SS04, Chapter 5]. The fundamental solution of the Helmholtz equation is an example of the class of kernel functions, for which these routines are developed [SS04, Example 5.1.17].

As the entry of a system matrix, according to (5.55), can be written as

\[
(\hat{A}_l)_{k,m} = \int_\Gamma \int_\Gamma G_{\kappa l}(\|x-y\|)b_m(y)b_k(x)d\Gamma_y d\Gamma_x = \sum_{\tau \in G} \sum_{t \in G} \int_{\tau} \int_{t} \ldots d\Gamma_y d\Gamma_x,
\]

where \(0 \leq l \leq N\) and \(1 \leq k,m \leq M\), the methods have to be developed just for the integration over a pair of triangles \(\tau \times t\) for \(\tau,t \in G\). As

\[
G_{\kappa l}(\textbf{x,y}) = \frac{e^{i\kappa\|\textbf{x-y}\|}}{4\pi\|\textbf{x-y}\|},
\]

is singular if \(\tau \cap \bar{t} \neq \emptyset\) or nearly singular for a small distance between \(\tau\) and \(t\), special routines have to be developed to calculate the matrix entries efficiently and exactly enough. As mentioned above, this is done in [SS04]. Here only the most important points are recalled:

(i) Using the transformations \(\chi_\tau\) resp. \(\chi_t\), the integration over a pair of triangles \(\tau \times t\) is pulled back to the reference element \(\hat{\tau} \times \hat{t}\), where the quadrature routine is developed.

(ii) Relative coordinates \(z = x - y\) are introduced to fix the singularity \(\|x-y\|\) at the origin. Then an \(\varepsilon\) neighbourhood is cut out and a 'good', in the sense of appropriate to the properties of the integral kernel, decomposition of the integration domain is chosen. After changing the integration order, the integration domain is transferred to the four dimensional simplex. Finally the integration domain is transformed on \([0,1]^4\). Furthermore it is shown, that in each former singular case, the integrand can be analytically extended onto a complex neighbourhood of \([0,1]^4\).

(iii) The resulting regularizing coordinate transforms require only some mild assumptions on the parametrization of the two triangles.

The assumptions mentioned in Point (iii) are the following.

**Assumption 7.4.1.** Let be \(\tau,t \in G\) and \(\chi_\tau\) resp. \(\chi_t\) their parametrizations. It holds:

(i) In the case of identical panels: \(\chi_\tau = \chi_t\).

(ii) In the case of a common edge: \(\chi_\tau(s,0) = \chi_t(s,0)\) for all \(s \in [0,1]\).

(iii) In the case of a common vertex: \(\chi_\tau(0,0) = \chi_t(0,0)\).

To assert that the triangles are arranged in exactly that manner is one of the most important and most sensitive parts of the implementation.

Furthermore, let us define the integrand in local coordinates

\[
k_{loc}(\hat{\textbf{x}},\hat{\textbf{y}}) = \hat{b}_m(\hat{\textbf{y}})\hat{b}_m(\hat{\textbf{x}})k(\chi_\tau(\hat{\textbf{x}}),\chi_t(\hat{\textbf{y}}))g_\tau(\hat{\textbf{x}})g_t(\hat{\textbf{y}}),
\]

where \(g_\tau\) resp. \(g_t\) denotes the Gramian Determinant of \(\chi_\tau\) resp. \(\chi_t\).

\[
g_\tau(\hat{\textbf{x}}) = \sqrt{\det(D\chi_\tau(\hat{\textbf{x}}))^T(D\chi_\tau(\hat{\textbf{x}}))}.
\]
Under these assumptions we have the following regularizing transforms:

- **Identical Panels**:

\[
I_{\tau \times t} = \int_{(0,1)^4} \xi^3 \eta_1^2 \eta_2 \left\{ k_{\text{loc}} \left( \xi \left( \begin{array}{c} 1 \\ 1 - \eta_1 + \eta_1 \eta_2 \\ 1 - \eta_1 \eta_2 \eta_3 \\ 1 - \eta_1 \end{array} \right) \right) + k_{\text{loc}} \left( \xi \left( \begin{array}{c} 1 \\ 1 - \eta_1 \eta_2 \eta_3 \\ 1 - \eta_1 \eta_2 \\ 1 - \eta_1 \end{array} \right) \right) + k_{\text{loc}} \left( \xi \left( \begin{array}{c} 1 \\ 1 - \eta_1 \eta_2 \eta_3 \\ 1 - \eta_1 \eta_2 \\ 1 - \eta_1 \end{array} \right) \right) \right\} d\eta_1 d\eta_2 d\eta_3 d\xi.
\]

- **Common Edge**:

\[
I_{\tau \times t} = \int_{(0,1)^4} \xi^3 \eta_1^2 k_{\text{loc}} \left( \frac{\xi}{\xi(1 - \eta_1 \eta_2)} \left( \frac{\xi \eta_1 \eta_3}{\xi(1 - \eta_1 \eta_2)} \right) + \xi^3 \eta_1^2 \left( \frac{\xi}{\xi(1 - \eta_1 \eta_2)} \left( \frac{\xi \eta_1 \eta_3}{\xi(1 - \eta_1 \eta_2)} \right) \right) \right) + k_{\text{loc}} \left( \frac{\xi}{\xi(1 - \eta_1 \eta_2)} \left( \frac{\xi \eta_1 \eta_3}{\xi(1 - \eta_1 \eta_2)} \right) \right) \right\} d\eta_1 d\eta_2 d\eta_3 d\xi.
\]

- **Common Vertex or Positive Distance**:

\[
I_{\tau \times t} = \int_{(0,1)^4} \xi^3 \eta_2 \left\{ k_{\text{loc}}(\xi, \xi \eta_1, \xi \eta_2, \xi \eta_3) + k_{\text{loc}}(\xi \eta_2, \xi \eta_2 \eta_3, \xi, \xi_1) \right\} d\eta_1 d\eta_2 d\eta_3 d\xi.
\]

As we now have an analytic integrand \( f : [0,1]^4 \to \mathbb{C} \), we can employ a tensorised Gauss quadrature on it. That means, that the exact integral

\[
I(f) := \int_{(0,1)^4} f(x) dx,
\]

is approximated by a quadrature of the form

\[
Q^n(f) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \sum_{l=1}^{n_4} \omega_{i,n_1} \omega_{j,n_2} \omega_{k,n_3} \omega_{l,n_4} f(\xi_{i,n_1}, \xi_{j,n_2}, \xi_{k,n_3}, \xi_{l,n_4}),
\]

with \( n = (n_1, n_2, n_3, n_4)^T \in \mathbb{N}^4 \). Furthermore \( \omega_{i,n} \) resp. \( \xi_{i,n} \) for \( n \in \mathbb{N} \) and \( 1 \leq i \leq n \) are the weights resp. nodes for the Gauss quadrature of order \( n \) on \( [0,1] \). They can be calculated once for different \( n \) to a high precision and can be saved in a file, for using it in each quadrature.

In [SS04, Chapter 5] the analysis of the quadrature error and the question, how \( n \) has to be chosen that the quadrature error does not spoil the convergence rate of a Galerkin boundary element method is developed. The practical results are presented in [SS04, Chapter 5.3.4]. In the case of the Helmholtz problem, where the single layer potential defines a boundary integral operator from \( H^{-1/2}(\Gamma) \) to \( H^{1/2}(\Gamma) \) and induces an integral equation of negative order, we get the following result.
**Proposition 7.4.2.** For the singular cases, i.e. identical panel, common edge and common vertex, the quadrature order have to be chosen as $n = (n_1, n_2, n_2, n_2)^T$. That means there is one order for the variable $\xi$ and one order for the variables $\eta_1, \eta_2, \eta_3$. In the regular case, i.e. the panels $\tau, t$ have positive distance $d_{\tau,t} > 0$, the quadrature order is all the same in each variable. However there are two different cases arising. Let $\chi_{\tau,t} := \max\{d_{\tau,t}/h, 1\}$.

1. In the near field case, where $\chi_{\tau,t} \sim 1$, i.e. the integrand is nearly singular, the quadrature order is denoted by $n_{\text{near}}$.

2. In the far field case, where $d_{\tau,t}$ is in another scale than $h$, i.e. $\chi_{\tau,t} \sim h^{-1}$, the quadrature order is denoted by $n_{\text{far}}$.

The integral order have to be chosen in the following way

<table>
<thead>
<tr>
<th>$p$</th>
<th>$n_1$</th>
<th>$n_2$</th>
<th>$n_{\text{near}}$</th>
<th>$n_{\text{far}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>$</td>
<td>3C_1</td>
<td>\log h</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>$</td>
<td>4C_1</td>
<td>\log h</td>
</tr>
</tbody>
</table>

Table 7.1: The rules for the proper choice of the integration order

where $p = 0$ resp. $p = 1$ in the case of piecewise constant elements resp. piecewise linear elements and $C_1, C_2$ independent of $p, h, d_{\tau,t}$.

This proposition gives us a recipe, how to choose the integration order in the different cases. For the constants $C_1, C_2$ and the deciding value for $\chi_{\tau,t}$, where the near field is differed from the far field, testing is necessary.

The basic integration routines are provided by the package of Dursun Akay (cf. [Aka10]). The only point where the implementation differs from the introduced methods is the handling of the Gramian Determinant. As we just have plane triangles with straight edges, we choose the parametrization as proposed in [5.10]. Thus

$$\chi_{\tau}(\tilde{x}) = A + M_{\tau} \cdot \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}.$$

(7.55)
Therefore, it holds \( D \chi_\tau(\hat{x}) = M_\tau \), that is independent of \( \hat{x} \). Hence \( g_\tau(\hat{x}) \) is a constant. More precisely, the columns of \( M_\tau \) contain the coordinates of two different edge-vectors of the triangle \( \tau \). Let us denote them as \( E \) and \( F \). Therefore we have

\[
g^2_\tau = \det \left( \begin{pmatrix} -E & - \end{pmatrix} \cdot \begin{pmatrix} E & F \end{pmatrix} \right) = \det \left( \begin{pmatrix} |E|^2 & \langle E, F \rangle \\ \langle F, E \rangle & |F|^2 \end{pmatrix} \right) 
\]

(7.56)

\[
ge^2_\tau = |E|^2 |F|^2 - \langle E, F \rangle^2, \quad (7.57)
\]

where \( \langle \cdot, \cdot \rangle \) is the Euclidean scalar product in \( \mathbb{R}^3 \) and \( |\cdot| \) the Euclidean norm. Using Lagrange’s identity leads to

\[
g^2_\tau = |E|^2 |F|^2 - \langle E, F \rangle^2 = \langle E \times F, E \times F \rangle = |E \times F|^2 \quad (7.58)
\]

\[
ge_\tau = 4|\tau|, \quad (7.59)
\]

where \( \times \) is the outer product on \( \mathbb{R}^3 \) and \( |\tau| \) the measure of \( \tau \). Thus, instead of calculating \( g_\tau \), one can determine the area of \( \tau \) and use the representation

\[
g_\tau = 2|\tau|. \quad (7.60)
\]

### 7.5 Routines

The program flow of the implementation is presented in Section 7.1. In this section the single routines are documented. Instead of giving a detailed description of the implementation, the main principles used for coding are presented at the beginning. Apart from these principles the implementation is straightforward and further details can be found in the commented code. Technical difficulties arise especially in the handling of the mesh information. However, that issue is presented in details in the Section 7.5.1.

Everywhere in the discretisation, finite sums or equations appear which have to be satisfied for various finite parameters. The \texttt{for}-loops are a straightforward way to implement such sums. However MATLAB offers a powerful way to speed up the program by avoiding these loops. This concept is called vectorization. The MATLAB documentation gives a few simple illustrating examples for that concept. The fundamental idea is to transform a non-recursive loop into a vector operation. The entries of the vector represent the different steps of the loop. The gain of speed is in many cases striking. Furthermore, various MATLAB routines are already programmed in a vectorized way, e.g. the fast Fourier transform \texttt{fft}, that can apply the transform on different vectors simultaneously.

Strongly linked to the vectorization is the concept of linear indexing. As each \( m \times n \) matrix \( A \) can be identified by a vector with \( m \cdot n \) entries, vectorization also can be applied to matrix operations. In MATLAB linear indexing is done in the following way

\[
A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix},
\]

(7.61)

is identified with the vector

\[
(a_{11} \cdots a_{m1} \ a_{12} \cdots a_{m2} \cdots a_{1n} \cdots a_{mn}).
\]

Hence the entry in the matrix at the index \( (i, j) \) is identified with entry in the vector with index \( i + (j - 1) \cdot m \). Therefore the same concepts as for vectors can be applied to matrices. The implementation makes use of these concepts as often as possible. As mentioned, the impact on the performance is substantial. However, the readability of the code suffers from that, as it is not clear at first glance, which values are assigned to which variable.
7.5.1 Arranging the Pairs of Triangles

In the preprocessing phase of the algorithm, i.e. before the different Helmholtz problems are distributed to different processors, the set of triangles for the integration is prepared. As stated in Assumption 7.4.1, the parametrizations of the triangles play an important role, as the regularizing transformations depend on the geometry and the explicit parametrizations. Hence for an integration \( \int_\tau \int_t \ldots \) for \( \tau, t \in \mathcal{G} \), where \( \tau \cap t \neq \emptyset \), the local numbering of the vertices of the triangles \( \tau \) and \( t \) has to be arranged properly in order to fulfill Assumption 7.4.1. Due to the fact that for each Helmholtz problem the same integration domains \( \tau \times t \) arise, and to be able to effect the integration in a vectorized way, all possible triangle combinations are prepared in advance.

Using the fact that all system matrices defined by (5.55) are symmetric but non-hermitian, due to the symmetry of the integration kernels respectively the imaginary entries on the diagonal, only pairs of triangles of the form \( \tau_i \times \tau_j \) with \( 1 \leq i \leq j \leq N_\tau \) have to be considered. That does not reduce the computational complexity order, but halves the time for the computation and the memory requirement. The symmetry is taken into account by assigning the calculated integrals to the system matrices \( \hat{A}_l \), i.e. the values are assigned to \((\hat{A}_l)_{k,m}\) as well as \((\hat{A}_l)_{m,k}\).

The result of this preparation are two \( L \times 3 \) matrices \( \text{triangle1} \) and \( \text{triangle2} \), saved in the structure \text{quadrature}, where \( L = N_\tau^2 + N_\tau \). In the \( l \)-th row of these two matrices the local numbering of the vertices of the triangles \( \tau, t \in \mathcal{G} \) are saved, arranged in the way that Assumption 7.4.1 is fulfilled. This arrangement is once more done in vectorized form in the file \text{vectTriOrder_Gauss.m} \. The structure \text{quadrature} has therefore storage costs of order \( O(M^2) \).

7.5.2 Calculating the System Matrices

The fundamental part of the algorithm is to calculate the entries of the system matrices \( \hat{A}_l \). They are given by (5.55) as

\[
(\hat{A}_l)_{k,m} = \int_\Gamma \int_\Gamma G_{\kappa_{l}}(\|x - y\|)b_{m}(y)b_{k}(x)d\Gamma_{y}d\Gamma_{x},
\]  

(7.63)

for \( 0 \leq l \leq N \) and \( 1 \leq k, m \leq M \). Let us look at the calculation of these entries for the different approximation spaces defined in Section 5.2.

(i) Recall that in the case of piecewise constant functions the basis functions \( b_m \) are the characteristic functions for the triangles \( \tau_m \). Therefore, the integral over \( \Gamma \times \Gamma \) reduces to

\[
(\hat{A}_l)_{k,m} = \int_{\tau_k} \int_{\tau_m} G_{\kappa_{l}}(\|x - y\|)d\Gamma_{y}d\Gamma_{x},
\]

(7.64)

i.e., an integral over a pair of triangles. That leads to the algorithm

```matlab
for 1 \leq k, m \leq M 
\quad (\hat{A}_l)_{k,m} = \int_{\tau_k} \int_{\tau_m} G_{\kappa_{l}}(\|x - y\|)d\Gamma_{y}d\Gamma_{x} 
end
```

To avoid the double for-loop, the quadrature package is built to handle the calculation of several entries at once by using the built-in vectorization.
(ii) The case of piecewise linear, continuous functions is more subtle. Let us have a look at
the support of a basis function. For this purpose recall the definition of the index set \( t_k \)
given in Definition 5.2.9
\[
\tau_k := \{ 1 \leq m \leq N_T | P_k \in \tau_m \} = \{ 1 \leq m \leq N_T | P_k \text{ is a vertex of } \tau_m \},
\]
(7.65)
for \( 1 \leq k \leq N_P \) so that
\[
\text{supp } b_k = \bigcup_{i \in \tau_k} \tau_i.
\]
(7.66)
This allows to reduce the integration over \( \Gamma \times \Gamma \) to
\[
(\hat{A}_l)_{k,m} = \int_{\text{supp } b_k} \int_{\text{supp } b_m} G_{\kappa_l}(\|x-y\|) b_m(y)b_k(x) d\Gamma_y d\Gamma_x
\]
(7.67)
\[
= \sum_{i \in \tau_k} \sum_{j \in \tau_m} \int_{\tau_i} \int_{\tau_j} G_{\kappa_l}(\|x-y\|) b_m(y)b_k(x) d\Gamma_y d\Gamma_x,
\]
(7.68)
which is again in the form, where the integration domain is a pair of triangles.

The next step is the development of an efficient implementation. We explain how to
avoid the explicit generation of the index sets \( \tau_k \) for \( 0 \leq k \leq N_P \) by changing the point
of view for this calculation. Up to now, our loop is organized so that the element \((\hat{A}_l)_{k,m}\)
are generated independently. An alternative view is based on the generation of 'element
matrices for pairs of panels'. For two triangles \( \tau \) and \( t \) the integration \( \int_{\tau} \int_{t} \ldots \) arises in
the calculation of nine entries, namely for every \( k, m \) such that \( P_k \) is a vertex of \( \tau \) and
\( P_m \) is a vertex of \( t \). Hence, the following algorithm results

```matlab
for \( \tau, t \in \mathcal{G} \)
- \text{for } k, m \in \{1, \ldots, M\} \text{ such that } P_k \text{ resp. } P_m \text{ is a vertex of } \tau \text{ resp. } t
- (\hat{A}_l)_{k,m} = (\hat{A}_l)_{k,m} + \int_{\tau} \int_{t} G_{\kappa_l}(\|x-y\|) b_m(y)b_k(x) d\Gamma_y d\Gamma_x
- end
end
```

The double for-loop is resolved analogously to \((\hat{1})\) by the vectorized quadrature package.

### 7.5.3 Calculating the Right-Hand Side Vectors

In order to get the decoupled linear systems of equations given by \((5.56)\), the right-hand side
vectors have to be built. The discretisation gives us in Equation \((5.55)\)
\[
(\hat{g}_l)_k = \int_{\Gamma} \delta_{l}^{\Delta t}(x)b_k(x)d\Gamma_x,
\]
(7.69)
where \( 0 \leq l \leq N \) and \( 1 \leq k \leq M \) and the transformed right-hand side function is defined by
\[
\delta_{l}^{\Delta t}(x) := \sum_{j=0}^{N} \lambda^j \phi_j^{\Delta t}(x) \zeta_{N+1}^l,
\]
(7.70)
with \( \phi_j^{\Delta t}(x) \) an approximation of \( g(x, j \cdot \Delta t) \). Assume that \( g \) is known explicitly, e.g. defined
in MATLAB by a function. Hence
\[
(\hat{g}_l)_k = \int_{\Gamma} \left( \sum_{j=0}^{N} \lambda^j \phi_j^{\Delta t}(x) \zeta_{N+1}^l \right) b_k(x)d\Gamma_x
\]
(7.71)
\[
= \sum_{j=0}^{N} \lambda^j \left( \int_{\text{supp } b_k} g(x, t_j)b_k(x)d\Gamma_x \right) \zeta_{N+1}^l =: \sum_{j=0}^{N} \lambda^j b_{j+k} \zeta_{N+1}^l,
\]
(7.72)
CHAPTER 7. MATLAB IMPLEMENTATION

gives a matrix \( b \in \mathbb{C}^{(N+1) \times M} \) of coefficients calculated by an integration

\[
b_{j+1,k} := \int_{\text{supp} b_k} g(x, t_j) b_k(x) d\Gamma_x, \quad 0 \leq j \leq N, 1 \leq k \leq M. \tag{7.73}
\]

The last step, the scaled Fourier transform is applied to the columns of \( b \) (cf. Section 7.5.4).

For the calculation of \( b \) we distinguish once more the two cases of basis functions.

(i) For \( b_k \) a piecewise constant basis functions holds \( \text{supp} b_k = \tau_k \) and \( b_k \big|_{\tau_k} \equiv 1 \). That leads to

\[
b_{j,k} = \int_{\tau_k} g(x, t_j) d\Gamma_x, \tag{7.74}
\]

and the algorithm

```
for 0 \leq j \leq N
    for 1 \leq k \leq M
        \( b_{j+1,k} = \int_{\tau_k} g(x, t_j) d\Gamma_x \)
    end
end
```

(ii) For \( b_k \) a piecewise linear, continuous basis function the set of indices \( \iota_k \) is defined as in Section 7.5.2, Part (ii). Using the same notation leads to

\[
b_{j,k} = \sum_{i \in \iota_k} \int_{\tau_i} g(x, t_j) b_k(x) d\Gamma_x. \tag{7.75}
\]

Again this can be reorganized by computing panelwise right-hand sides in analogy to the algorithm presented in Section 7.5.2, Part (ii).

```
for 0 \leq j \leq N
    for \( \tau \in G \)
        for \( k \in \{1, \ldots, M\} \) such that \( P_k \) is a vertex of \( \tau \)
            \( b_{j+1,k} = b_{j+1,k} + \int_{\tau} g(x, t_j) b_k(x) d\Gamma_x \)
        end
    end
end
```

Remark 7.5.1. In the case of a right-hand side function \( g \) separated in time and space variable, i.e. of the form \( g(t) e(x) \), the matrix \( b \) can be computed as dyadic product of two vectors \( g \in \mathbb{C}^{(N+1) \times 1} \) and \( e \in \mathbb{C}^{1 \times M} \) where

\[
g_j = g(t_j), \quad e_k = \int_{\text{supp} b_k} e(x) b_k(x) d\Gamma_x. \tag{7.76}
\]

The calculation of the integration is implemented analogously to the non-separated case discussed before.
7.5.4 Implementing the Scaled Fourier Transform

In Section 3.1 the scaled Fourier transform with a parameter \( \lambda > 0 \) is introduced. It appears in the calculation of the right-hand side vector (cf. Equation (7.72)) and the coefficients \( \tilde{\varphi}_{l,m} \) (cf. Equation (5.46)). In both cases, the transform can be applied to the columns of a matrix and has to be carried out with respect to the indices for the time. Therefore, the implementation of the transform between the columns of two matrices \( \mathbf{A}, \mathbf{\hat{A}} \in \mathbb{C}^{(N+1) \times M} \) has to be considered. The implementation uses the built-in functions \texttt{fft} and \texttt{ifft}, both operating on the columns of a matrix. Define furthermore a matrix \( \Lambda \in \mathbb{R}^{(N+1) \times (N+1)} \) by

\[
\Lambda_{i,j} = \begin{cases} 
\lambda^{i-1} & i = j \\
0 & i \neq j 
\end{cases}, \tag{7.77}
\]

which is realized in MATLAB as a sparse matrix. Using the connection between the scaled and the usual discrete Fourier transform presented in Remark 3.1.3, we get the following implementations

\[
\mathbf{\hat{A}} = \texttt{fft}(\Lambda \cdot \mathbf{A})
\]

\[
\mathbf{A} = \Lambda^{-1} \texttt{ifft}(\mathbf{\hat{A}})
\]

7.5.5 Calculating the Solution of the Wave Equation

In Section 5.5 it is explained how to compute an approximated solution of the wave equation in 'field points' of the exterior domain by using the numerically computed density \( \tilde{\varphi} \). The following representation holds (cf. Equation (5.72))

\[
u(x,t_j) \approx \sum_{m=1}^{M} \left[ \left( \tilde{\varphi}_{n,m} \left( 1 - \frac{t^*}{\Delta t} - n \right) \right) + \tilde{\varphi}_{n+1,m} \left( \frac{t^*}{\Delta t} - n \right) \right] \int_{\text{supp}_m} \frac{b_m(y)}{4\pi \|x-y\|} d\Gamma_y, \tag{7.78}
\]

with \( n := \left\lfloor \frac{t_j}{\Delta t} \right\rfloor \), \( t^* := t_j - \|x-y\| \) and \( t_j = j \cdot \Delta t \). The aim is to develop a vectorizable algorithm for both, piecewise constant as well as piecewise linear, basis functions is developed in order to evaluate \( u \) at a field point \( x \in \Omega^e \) at all timesteps \( t_j \).

(i) For piecewise constant elements we get

\[
u(x)_{j+1,m} = \left( \tilde{\varphi}_{n,m} \left( 1 - \frac{t^*}{\Delta t} - n \right) \right) + \tilde{\varphi}_{n+1,m} \left( \frac{t^*}{\Delta t} - n \right) \int_{\text{supp}_m} \frac{1}{4\pi \|x-y\|} d\Gamma_y, \tag{7.79}
\]

for \( 0 \leq j \leq N \), \( 1 \leq m \leq M \) such that \( \nu(x) \in \mathbb{C}^{(N+1) \times M} \). Therefore the approximation of \( \nu(x,t_j) \) is just the sum over the \((j+1)\)-th row.

\[
\begin{align*}
\text{for } 0 \leq j \leq N \\
\quad \text{for } 1 \leq k \leq M \\
\quad \quad \nu(x)_{j+1,k} = \left( \tilde{\varphi}_{n,k} \left( 1 - \frac{t^*}{\Delta t} - n \right) \right) + \tilde{\varphi}_{n+1,k} \left( \frac{t^*}{\Delta t} - n \right) \int_{\text{supp}_m} \frac{1}{4\pi \|x-y\|} d\Gamma_y \\
\quad \text{end} \\
\nu(x,t_j) \text{ is approximated by } \sum_{k=1}^{M} \nu(x)_{j+1,k} \\
\text{end}
\end{align*}
\]
(ii) Once more the only difference in the case of linear elements is the handling of the support of $b_m$, as $\text{supp } b_m = \bigcup_{i \in m} \Gamma_m$.

$$u(x)_{j+1,m} = (\tilde{\varphi}_{n,m}(1 - (t^* - n)) + \tilde{\varphi}_{n+1,m}(t^* - n)) \sum_{i \in m} \int_{\Gamma_i} \frac{b_m(y)}{4\pi\|x-y\|} d\Gamma_y. \tag{7.80}$$

Using the same consideration as in the computation of the system matrices in Section 7.5.2 resp. right-hand side vectors in Section 7.5.3 one gets the algorithm for the matrix $u(x)$, analogously to the precedent Point (i), where $u(x, t_j)$ again is approximated by the sum over the $(j+1)$-th row.

```matlab
for 0 \leq j \leq N
    for \tau \in G
        for k \in \{1, \ldots, M\} such that P_k is a vertex of \tau
            u(x)_{j+1,k} = u(x)_{j+1,k} + (\tilde{\varphi}_{n,k}(1 - (t^* - n)) + \tilde{\varphi}_{n+1,k}(t^* - n)) \int_{\Gamma} \frac{b_k(y)}{4\pi\|x-y\|} d\Gamma_y
        end
    end
end
u(x, t_j) is approximated by \sum_{k=1}^{M} u(x)_{j+1,k}
```

### 7.5.6 Description of the Routines

After this detailed presentation of the fundamental steps of the program, the routines used for the calculation of the coefficients of the unknown density are listed. If an input or output parameter is not documented any further, then it is a part of a data structure described in Section 7.3. Furthermore the whole source code is commented and provides additional information. For input or output parameter, whose storage costs depend on $M$ or $N$, the order of their storage costs is indicated.

(i) **BEM_main_multicore.m**: This is the main program, that has to be run on the 'Master' session. It controls the whole course of actions, which are necessary to get the wished solution.

**input:**

(a) $T$: endtime
(b) $N$: number of discrete time step
(c) file: name of the mesh-data file generated by Gmsh
(d) $\text{order}$: polynomial order of approximation (0: piecewise constant elements; 1: continuous, piecewise linear elements)
(e) testCase: choice of testcase 0, 1, 2, 3, 4 (cf. Section 7.3)
(f) intlogh: choice of the integration order (true: following Proposition 7.4.2 false: constant (defined in vectTriOrder_Gauss.m))

**output:**

(a) meshb: built and used mesh structure, storage costs of order $O(M^2)$
(b) paramb: built and used parameter structure, storage costs of order $O(N)$
(c) phi: a $(N+1) \times M$ matrix containing the coefficients for the approximation of the unknown density $\tilde{\varphi}_{n,m}$ introduced in Section 5.4 storage costs of order $O(MN)$
(d) err: a single number in $\mathbb{R}$ representing the measure for the error (cf. Section 8.2)

(ii) **calcError.m**: To get information about the convergence rates, this routine determines the error between the approximated and the analytical solution using discrete $L^2$-norm
7.5. ROUTINES

of Section 8.2

input:
(a) sol: a \((N+1) \times N_\tau\) matrix with the values of the analytical solution at the midpoints of the triangle, storage costs of order \(O(MN)\)
(b) phi: cf. description of \texttt{BEM\_main\_multicore.m}, storage costs of order \(O(MN)\)

output:
(a) err: cf. description of \texttt{BEM\_main\_multicore.m}

(iii) \texttt{completeMeshData.m}: To get all the information which are saved in the mesh structure, this routine does all the needed computations. Some part are taken or inspired by the package MESH2D by Darren Engwirda\(^3\). Especially the routine \texttt{connectivity.m} was very helpful.

input: none
output: none

(iv) \texttt{getRightSide.m}: This routine calculates the right-hand side vector \(\hat{g}_l\) for each \(0 \leq l \leq N\) as introduced in equation (5.56). It works for right-hand side functions \(g(x,t)\) (just pass one argument) as well as for right-hand side functions of tensor form \(g(t)e(x)\)

input:
(a) \(g\): a function handle for \(g(x,t)\) resp. the time dependent part \(g(t)\)
(b) \(e\): a function handle for the space dependent part \(e(x)\)

output:
(b): a \(M \times (N+1)\) matrix containing the vector \(\hat{g}_l\) in the \(l\)-th column, storage costs of order \(O(MN)\)

(v) \texttt{getSolution.m}: In different test cases (0 and 1), it is possible to calculate the analytical solution. This routine determines the value of the analytical solution at the midpoints of the triangles at all different time steps.

input:
(a) \(dg\): a function handle for the derivative of time dependent part written as \(g'(t)\)
(b) \(e\): a function handle for the space dependent part \(e(x)\)

output:
(a) sol: a \((N+1) \times N_\tau\) matrix with the values of the analytical solution at the midpoints of the triangle, storage costs of order \(O(MN)\)

(vi) \texttt{getSystemMatrix.m}: This routine calculates the system matrix for one Helmholtz problem using the formula in equation (5.55) and the ’Quadrature’ package. Further details on the structure of this routine is provided at the end of the Section 7.4

input:
(a) \(t\): a single natural number giving the index of the time step \(0 \leq t \leq N\)

output:
(a) \(A\): a \(M \times M\) matrix representing the system matrix, storage costs of order \(O(M^2)\)

(vii) \texttt{IFFTlambda.m}: To calculate the coefficients of the unknown density \(\tilde{\varphi}\) for the wave equation out of the coefficients \(\hat{\varphi}\) for the different Helmholtz problems, this routine implements the formula (5.57) in vectorized form.

input:
(a) \(\text{phihat}\): a \((N+1) \times M\) matrix containing the coefficients of the unknown densities \(\tilde{\varphi}_{n,m}\) for the Helmholtz problems given by (5.56), storage costs of order \(O(MN)\)

\(^3\) The package can be found on the homepage of MATLAB Central \url{http://www.mathworks.com/matlabcentral/fileexchange/25555-mesh2d-automatic-mesh-generation} (13.10.2010)
output:

(a) \( \phi \): a \((N+1) \times M\) matrix containing the coefficients for the approximation of the unknown density \( \tilde{\phi}_{n,m} \) for the Equations (5.44), storage costs of order \( O(MN) \)

(viii) \texttt{readMeshFile.m}: This is the routine that reads the .msh-file generated by Gmsh and fills the data into the mesh structure.

\textit{input:}

(a) \texttt{filename}: a string containing the name of the .msh-file

\textit{output:} none

(ix) \texttt{solveTimeStep.m}: This is the main routine that handles the solving of a single Helmholtz problem in the different MATLAB sessions. The arising system of linear equations is solved using the built-in function \texttt{linsolve}, that uses an \texttt{LU} factorization.

\textit{input:}

(a) \texttt{l}: a single natural number giving the index of the time step \( 0 \leq l \leq N \)
(b) \texttt{b}: a \( M \times 1 \) vector representing the right-hand side vector for the \( l \)-th Helmholtz problem, storage costs of order \( O(M) \)

\textit{output:}

(a) \texttt{phihat}: a \((N+1) \times M\) matrix containing the coefficients of the unknown densities \( \tilde{\phi}_{n,m} \) for the Helmholtz problems given by (5.56), storage costs of order \( O(MN) \)

(x) \texttt{vectTriOrder_Gauss.m}: This routine generates vectorized integration data prepared for the 'Quadrature'-package and determines the needed integration order using the rules presented in Table 7.1.

\textit{input:}

(a) \texttt{order}: the polynomial order of approximation defining the used boundary element functions

\textit{output:} none

To calculate the coefficients of the right-hand side vectors and the system matrices, a 'Quadrature' package is employed. That package was developed by Dursun Akay (cf. \cite{Akay10}) in collaboration with the author. It consists of the following files:

\texttt{evalInt.m, GauLeg.mat, gaussQuad.m, gaussQuad_X_W.m, gaussQuadMain.m, gaussQuadMainPlot.m, gaussQuadStep.m, getIndexMat.m, getJacobian.m, getTriangleMat.m, getTriangleVertices.m, getVariable.m, k3.m, linTransform.m}

Not all of them are described any further, as in the final implementation optimized and more specific versions of several routines are applied.

Let us describe the optimized functions in more detail, however we skip the input and output arguments. They are structured into several sections. The first one is executed during the preprocessing phase by calling the file 'getVariableOnce.m'.

(i) \texttt{GauLeg.mat}: A list containing high precision weights and nodes for Gauss quadrature on \([0,1]\) up to the order 100.

(ii) \texttt{gaussQuad_X_W.m} Here the integration nodes and weights are loaded from \texttt{GauLeg.mat} for given integration order.

(iii) \texttt{getIndexMat.m}: That is a routine providing a special matrix of indices.
7.5. ROUTINES

(iv) getOptIndex.m: This routine loads and provides the information about the occurring integration weights and nodes.

(v) getTriangleMat.m: Herein the matrices for the affine transformations $\chi_\tau$ for a whole set of triangles $G$ are built.

(vi) getVariableOnce.m: This is the main routine for the preparation of the integration process. It gets all prepared data from the main program and provides the data concerning the integration to solve the different Helmholtz problems.

The second part contains the routines for calculating the right-hand side vectors using the algorithms developed in Section 7.5.3.

(i) gaussQuadMain_2d.m: That is the main routine handling the integration of the form $\int_\tau \cdots$.

(ii) k3_2d.m: Here the integrand is evaluated at the chosen nodes and then multiplied by the basis functions.

(iii) linTransform.m: In that routine the nodes are projected onto the occurring triangle $\tau$ using $\chi_\tau$.

In the third and last part, the routines for the quadrature used for calculating the system matrices following the algorithms of Section 7.5.2 are listed.

(i) evalComEdge.m: Herein the quadrature in the case of common edges is handled.

(ii) evalComVert.m: Herein the quadrature in the case of common vertices or distant triangles is handled.

(iii) evalIdentical.m: Herein the quadrature in the case of identical triangles is handled.

(iv) gaussQuadMain_4d.m: That is the main routine to handle the quadrature of integrals of the form $\int_\tau \int_t \cdots$.

(v) k3_4d.m: Here the integrand is evaluated at the chosen nodes and then multiplied by the basis functions.

(vi) linTransform.m: In that routine the nodes are projected onto the occurring triangle $\tau$ using $\chi_\tau$.

Finally a description of the files for the calculation of the solution of the wave equation $u$ follows. For the calculation of the values, there is no restriction for the chosen field points $x \in \Omega$. However, the points have to be arranged properly in a single plane in order to get a plot of the cross section of the wave.

(i) calcWaveFunction_multicore.m: This routine handles the calculation of the solution of the wave equation. It distributes the work on arbitrary many MATLAB session as it is known from BEM_main_multicore.m. In the source code, the field points, where the wave function shall be calculated, are defined.

**input:**

(a) phi: the coefficients for the unknown density $\varphi$ resulted from BEM_main_multicore.m

(b) mesht: the corresponding mesh structure

(c) paramt: the corresponding param structure

**output:**
(a) $u$: the value of the solution of the wave equation $u$ at the equidistant discrete timesteps and the above chosen points

(b) film: the resulting film generated by MATLAB for the use in the function movie

c) graph: a structure containing all information about the chosen field points in the outer space

(ii) `gaussQuadMain_Wave.m`: That is the routine executing the integration over one triangle $\tau$ in the algorithms developed in Section 7.5.5.

(iii) `generateGraphicPoints.m`: This routine creates all information about the field points.

   - **input**: none
   - **output**: none

(iv) `plotWaveFunction.m`: This function generates the plots for the different timesteps and merges them together into a film.

   - **input**:
     - (a) $u$: the discrete values of the wave function
     - (b) graph: cf. description of `calcWaveFunction_multicore.m`

   - **output**:
     - (a) film: cf. description of `calcWaveFunction_multicore.m`

To evaluate the solution of the wave equation in a cuboid and present it in a more 3D-style there are functions with the suffix _slice, which currently are not parallelized. Furthermore there are a few functions to complete the whole package. They allow solving the problem $P$ (cf. Equation (2.4)) for a given incoming wave on $\mathbb{R}^3$ and plotting then the total wave $u_{tot}$. That allows the full animation of the scattering of a certain wave at an obstacle.

(i) `calcTotalWave.m`: This routine adds the value of the incoming wave to the value of the scattered wave at the field points.

   - **input**:
     - (a) $u$: a result of `calcWaveFunction_multicore.m`
     - (b) param: a result of `BEM_main_multicore.m`
     - (c) graph: a result of `calcWaveFunction_multicore.m`

   - **output**:
     - (a) $u$: the sum of the values of the incoming and scattered wave at the field points

(ii) `plotWaveFunction_total.m`: That is the final routine generating the film of the total wave including the scatterer.

   - **input**:
     - (a) $u$: a result of `calcTotalWave.m`
     - (b) graph: a result of `calcWaveFunction_multicore.m`
     - (c) mesh: a result of `BEM_main_multicore.m`

   - **output**:
     - (a) film: cf. description of `calcWaveFunction_multicore.m`

The program flow for calculating the total wave, that is caused by the scattering of an incoming wave at a scatterer, is listed in Figure 7.5.
7.6 Storage Costs

After having introduced the necessary global data structure in Section 7.3, input and output arguments of the routines in Section 7.5 and the data structure for the quadrature package in Section 7.5.1 we discuss the storage costs of the whole implementation. The following order of storage costs appear in the implementation, focusing on the data structures and not on their field variables:

(i) $O(M^2)$: This order arises as storage costs of the structure `mesh`, the system matrices $\mathbf{A}_l$ and structure `quadrature` providing the information for the quadrature package (cf. Section 7.5.1).

(ii) $O(MN)$: Any matrix containing the coefficients $\hat{\varphi}_{l,m}$ or $\tilde{\varphi}_{l,m}$, the matrix containing all the right-hand side vectors $\mathbf{g}_l$ for all decoupled systems as well as storing the value of the analytical solution in all barycenters for all times uses storage of the order $O(MN)$.

(iii) $O(N)$: The structure `param` only depends linearly on $N$.

Overall that results in storage costs for the whole implementation on a sequential computer of the order

$$O(MN) + O(M^2). \tag{7.81}$$

In contrast to the storage costs of order $O(M^2N)$ for solving the block Toeplitz system in (5.34) arising in the direct approach, our approach provides a major improvement with respect to the storage costs at the expense of solving $N$ systems of linear equations. As these $N$ systems of linear equations are independent, they can be solved on a parallel computer.
Assuming that \( P \) processors are available, the distributed memory requirement increases to \( O(PM^2) \) whereas the needed time is reduced by the factor \( P \). In our tests a typical choice is \( P = 6 \).

For the final time \( T \) applied in this thesis, \( N \) is much smaller than \( M \), as it can be seen in the parameters chosen for the tests in Section 8.3. According to the choice \( \Delta t^2 \sim h^{m+3/2} \) and taking into account that \( \Delta t \sim \frac{1}{N} \) respectively \( h \sim \frac{1}{\sqrt{M}} \), a typical choice is \( M \sim N \frac{8}{m+3} \). As \( 8 > 2m + 3 \) for \( m \in \{0, 1\} \), the order of the storage costs observed during the tests behaves as \( O(M^2) \), which dominates the order \( O(MN) \).

The philosophy to prepare in a preprocessing phase all necessary information, e.g. about the mesh or the arranged integration domains, causes the need to store this information in data structures. To apply the quadrature in its current, vectorized version further information, concerning the integration order or the parametrization of the triangles, has to be provided and is necessary to be stored. As the information for the quadrature routine scale with \( O(M^2) \) various matrices in the structure \texttt{quadrature}, which also need the same order of storage, are kept in memory without being used after the preprocessing phase, as they do not change the overall order of the storage costs. However, they provide useful information about the mesh which simplifies the debugging of the code.
Chapter 8

Numerical Tests

All numerical tests using the presented implementation are run on a parallel computer, called 'Baxter', built of 64 processors (Intel Xeon X7550) and equipped with approximately 500 gigabyte RAM. As the computer also is in use for other projects, at the most 20 and normally about 7 MATLAB sessions were started at the same time.

8.1 Test Case with known Analytical Solution

To validate the MATLAB program, we need a right-hand side function $g$, where the analytical solution $\phi$ of $V\phi = g$ is known explicitly. Therefore, we take data $g$ separable in time and space. Let us use the notation

$$g(t, x) = g(t)e(x). \quad (8.1)$$

As scatterer we choose the unit sphere $S^2$ in $\mathbb{R}^3$. For $e(x)$ we use an eigenfunction of $V$. These are the spherical harmonics $Y^m_l(\theta, \varphi)$, which are introduced in Section 3.8. As test case, we choose the spherical harmonic for $l = m = 1$. In spherical coordinates it holds

$$Y^1_1(\theta, \varphi) = -\sqrt{\frac{3}{8\pi}} \sin(\theta)e^{i\varphi}. \quad (8.2)$$

We set as $e(x)$ the spherical harmonic $Y^1_1$ in Cartesian coordinates. In [SV11] it is shown, that for $t \in [0, 2)$ the density $\phi$ solving the problem (2.8) is given by

$$\phi(t, x) = \left(2g'(t) + 2\int_0^t \sinh(\tau)g'(t - \tau)d\tau\right) \cdot e(x). \quad (8.3)$$

For $t \geq 2$ there also exists a formula for the exact solution $\phi$. However it is not implemented, as several integrals have to be evaluated. Similar results concerning the choice $e(x) = Y^0_0(x)$ are presented in [ibid.] as well, where the formula for $t > 0$ is much simpler, whereas this test case also is implemented. However, as $Y^0_0 = \frac{1}{2\sqrt{\pi}}$ is constant, there is no dependence with respect to $x$. That is the reason why this test case does not appear in any convergence test.

8.2 Error Measure

The result of Theorem [6.3.3] shows that we have to measure the error for each timestep $t_n$, $0 \leq n \leq N$ in the $H^{-1/2}(\Gamma)$-norm. As this norm is very time consuming to be evaluated numerically, we assume higher smoothness of $\phi$ and measure the error in a discrete $L^2(\Gamma)$-norm. The arising integral over $\Gamma$ is split into a sum of integrals over a single panel $\tau$. The approximation of the error then is done one these single panels $\tau$. Therefore, the difference at
the midpoint of $\tau$ is multiplied by the area on $\tau$, which corresponds to a one-point quadrature formula. On that way we get

$$\left\| \tilde{\varphi}_{\Delta t,h}^{n} - \varphi(\cdot,t_n) \right\|_{L^2(\Gamma)}^2 = \sum_{\tau \in G} \int_{\tau} \left| \tilde{\varphi}_{\Delta t,h}^{n}(y) - \varphi(y,t_n) \right| dy$$ \hspace{1cm} (8.4)

$$\approx \sum_{m=1}^{M} |\tau_m| \left| \tilde{\varphi}(N_{\tau_m},t_n) - \varphi(N_{\tau_m},t_n) \right|^2 =: e_n^2, \quad (8.5)$$

where $N_{\tau_m}$ is the barycenter of $\tau_m$. Finally we define the error $\tilde{e}_{L^2(\Gamma)}$ as the maximum of these errors, i.e. it is an error pointwise taken in time.

$$\tilde{e}_{L^2(\Gamma)} := \max_{0 \leq n \leq N} e_n. \quad (8.6)$$

Let the vertices of $\tau$ be $A, B, C$ with its global indices $1 \leq i_1 < i_2 < i_3 \leq N_P$. Then it holds

$$N_{\tau} = \frac{1}{3}(A + B + C). \quad (8.7)$$

For calculating $\tilde{\varphi}(N_{\tau_m},t_n)$, which arises in Equation (8.5), we have to distinguish two cases, depending on the approximation space $S$ introduced in Section 5.2.

- For piecewise constant elements, the value of the approximation at the barycenter of $\tau$ simply is the value of the appropriate coefficient.

$$\tilde{\varphi}(N_{\tau_m},t_n) = \tilde{\varphi}_{n,m} \quad (8.8)$$

- In the case of piecewise linear elements, the value at the barycenter of $\tau$ is the average of the values at the vertices of $\tau$.

$$\tilde{\varphi}(N_{\tau_m},t_n) = \frac{1}{3}(\tilde{\varphi}_{n,i_1} + \tilde{\varphi}_{n,i_2} + \tilde{\varphi}_{n,i_3}) \quad (8.9)$$

### 8.3 Convergence Tests

The goal of the numerical tests on the convergence is to provide information about the sharpness of the error bound given by Theorem 6.3.3. However, it cannot be expected that exactly the predicted convergence rate will result in these tests, as the mentioned error bound is an upper bound for the asymptotic behaviour in the limit $\Delta t, h \to 0$. In return, these tests also help to validate the implementation. If the convergence rate resulted in the tests is within a certain range of the rates predicted by Theorem 6.3.3, one has some evidence about the correctness of the implementation.

<table>
<thead>
<tr>
<th>$N_{\tau}$</th>
<th>$N_P$</th>
<th>$h_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>48</td>
<td>0.7134</td>
</tr>
<tr>
<td>332</td>
<td>168</td>
<td>0.4271</td>
</tr>
<tr>
<td>616</td>
<td>310</td>
<td>0.3120</td>
</tr>
<tr>
<td>1192</td>
<td>598</td>
<td>0.2163</td>
</tr>
<tr>
<td>1706</td>
<td>855</td>
<td>0.1805</td>
</tr>
<tr>
<td>2568</td>
<td>1286</td>
<td>0.1478</td>
</tr>
<tr>
<td>3386</td>
<td>1695</td>
<td>0.1292</td>
</tr>
<tr>
<td>5788</td>
<td>2896</td>
<td>0.0986</td>
</tr>
<tr>
<td>7890</td>
<td>3947</td>
<td>0.0854</td>
</tr>
<tr>
<td>9062</td>
<td>4533</td>
<td>0.0783</td>
</tr>
<tr>
<td>11074</td>
<td>5539</td>
<td>0.0706</td>
</tr>
<tr>
<td>13346</td>
<td>6675</td>
<td>0.0662</td>
</tr>
</tbody>
</table>

Table 8.1: Mesh data for the unit sphere
8.3. CONVERGENCE TESTS

We generate a set of meshes approximating the unit sphere in $\mathbb{R}^3$. The relevant data is listed in Table 8.1 where $N_\tau$ denotes the number of triangles, $N_P$ the number of nodes and $h_{\text{max}}$ the maximal triangle diameter.

We choose $T = 1$ as the final time. The right-hand side function is given by $g(t) \cdot e(x)$ with

$$g(t) = t^6 e^{-2t}, \quad e(x) = Y_1^1(x),$$

motivated by the results of Section 8.1. This right-hand side is an element of the space $H^5(0,T; H^{1/2}(\Gamma))$ and admits therefore a solution for the exact as well as for the discretised problem, as encountered in Theorem 4.7.1 resp. Theorem 6.3.3. Furthermore we take $\lambda = \Delta t^{\frac{3}{2}}$ as proposed in [BS08]. The other parameters are determined by performing tests, which lead to the following values.

- Determine the switch between near field and far field (cf. Proposition 7.4.2)

$$\frac{d_{r,t}}{h_{\text{max}}} > \frac{1.2}{h_{\text{max}}} \implies \tau \text{ is in the far field of } t \quad (8.11)$$

$$\frac{d_{r,t}}{h_{\text{max}}} \leq \frac{1.2}{h_{\text{max}}} \implies \tau \text{ is in the near field of } t \quad (8.12)$$

- Constants $C_1, C_2$ for determining the integration order (cf. Proposition 7.4.2)

$$C_1 = 0.5, \quad C_2 = 0.9 \quad (8.13)$$

- Integration order for calculating the right-hand side vectors $\hat{g}$ (cf. Equation 5.55):

$$n_{1,g} = n_{2,g} = 8 \text{ in both dimensions} \quad (8.14)$$

This choice leads to a negligible error caused by numerical calculation of the right-hand side vectors.

To illustrate the consequences of the choice of these variables, the resulting parameters for different meshes are listed in the next table. The number $r_{\text{n/f}}$ is a measurement for the partition of the pairs of triangles $\tau \times t$, where $\tau$ and $t$ have positive distance, into far field and near field. It is computed as the number of pairs of triangles in the near field divided by the number of pairs of triangles in the far field. Resulting from the choice of $C_1$ and $C_2$, the integration order $n_2$ and $n_{\text{near}}$ are computed using the formula presented in Proposition 7.4.2. The additional subscript "constant" and "linear" denote the case of the basis functions.

<table>
<thead>
<tr>
<th>$N_\tau$</th>
<th>$r_{\text{n/f}}$</th>
<th>$n_{2,\text{constant}}$</th>
<th>$n_{\text{near,constant}}$</th>
<th>$n_{2,\text{linear}}$</th>
<th>$n_{\text{near,linear}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>1.33</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>332</td>
<td>0.93</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>616</td>
<td>0.82</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1192</td>
<td>0.75</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>1706</td>
<td>0.71</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2568</td>
<td>0.69</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3386</td>
<td>0.67</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>5788</td>
<td>0.64</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>7890</td>
<td>0.63</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>9062</td>
<td>0.63</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>11074</td>
<td>0.62</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>13346</td>
<td>0.62</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 8.2: Quadrature orders and relation between triangle combinations in the near field and far field for the proposed constants.
8.3.1 Piecewise Constant Elements

As first test we examine the dependence of the error with respect to decreasing $\Delta t$ and therefore increasing $N$. We fix the spatial boundary mesh at $N_r = 1192$. The results then are listed in Table [8.3]. The integration orders are chosen as $3$ in every direction to eliminate an error depending on the choice of near and far field.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$N + 1$</th>
<th>$\Delta t$</th>
<th>$\epsilon$</th>
<th>rate wrt. $\Delta t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1192</td>
<td>4</td>
<td>0.3333</td>
<td>0.1156</td>
<td>-</td>
</tr>
<tr>
<td>1192</td>
<td>6</td>
<td>0.2000</td>
<td>0.0541</td>
<td>1.49</td>
</tr>
<tr>
<td>1192</td>
<td>8</td>
<td>0.1429</td>
<td>0.0291</td>
<td>1.84</td>
</tr>
<tr>
<td>1192</td>
<td>10</td>
<td>0.1111</td>
<td>0.0179</td>
<td>1.94</td>
</tr>
<tr>
<td>1192</td>
<td>12</td>
<td>0.0909</td>
<td>0.0120</td>
<td>1.98</td>
</tr>
<tr>
<td>1192</td>
<td>14</td>
<td>0.0769</td>
<td>0.0086</td>
<td>1.99</td>
</tr>
<tr>
<td>1192</td>
<td>16</td>
<td>0.0667</td>
<td>0.0066</td>
<td>1.89</td>
</tr>
<tr>
<td>1192</td>
<td>18</td>
<td>0.0588</td>
<td>0.0052</td>
<td>1.90</td>
</tr>
<tr>
<td>1192</td>
<td>20</td>
<td>0.0526</td>
<td>0.0049</td>
<td>0.56</td>
</tr>
<tr>
<td>1192</td>
<td>22</td>
<td>0.0476</td>
<td>0.0049</td>
<td>0.01</td>
</tr>
<tr>
<td>1192</td>
<td>24</td>
<td>0.0435</td>
<td>0.0049</td>
<td>0.00</td>
</tr>
<tr>
<td>1192</td>
<td>26</td>
<td>0.0400</td>
<td>0.0049</td>
<td>0.00</td>
</tr>
<tr>
<td>1192</td>
<td>28</td>
<td>0.0370</td>
<td>0.0049</td>
<td>0.00</td>
</tr>
<tr>
<td>1192</td>
<td>30</td>
<td>0.0345</td>
<td>0.0049</td>
<td>0.00</td>
</tr>
<tr>
<td>1192</td>
<td>32</td>
<td>0.0323</td>
<td>0.0049</td>
<td>0.00</td>
</tr>
<tr>
<td>1192</td>
<td>34</td>
<td>0.0303</td>
<td>0.0049</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 8.3: Convergence test for constant elements and increasing $N$

Figure 8.1: Plot of the error for the test with respect to $\Delta t$

The results behave like the theoretical upper estimate. We have chosen the mesh width $h$ for the spatial mesh small enough, so that the convergence rate of the total error is dominated by the temporal convergence rate with respect to the (large) timesteps. The error is decreasing by the rate, which the theory predicts, until the total error is dominated by the error caused by space discretisation. Afterwards, the error remains constant.

As second test, we make use of the recipe which was formulated in Corollary [6.3.4]. It proposes to simultaneously decrease $\Delta t$ and $h_{\text{max}}$ in the form $h_{\text{max}}^{3/2} \sim \Delta t^2$ for piecewise constant elements. The choice of $M$ and $N$ are presented in Table [8.4]. The integration orders are
8.3. CONVERGENCE TESTS

taken from Table 8.2

<table>
<thead>
<tr>
<th>$M$</th>
<th>$N + 1$</th>
<th>$\Delta t$</th>
<th>$e$</th>
<th>rate wrt. $\Delta t$</th>
<th>rate wrt. $h_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>4</td>
<td>0.3333</td>
<td>0.2463</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>332</td>
<td>5</td>
<td>0.2500</td>
<td>0.0819</td>
<td>3.83</td>
<td>2.15</td>
</tr>
<tr>
<td>616</td>
<td>6</td>
<td>0.2000</td>
<td>0.0517</td>
<td>2.06</td>
<td>1.46</td>
</tr>
<tr>
<td>1192</td>
<td>8</td>
<td>0.1429</td>
<td>0.0291</td>
<td>1.70</td>
<td>1.56</td>
</tr>
<tr>
<td>1706</td>
<td>9</td>
<td>0.1250</td>
<td>0.0220</td>
<td>2.12</td>
<td>1.57</td>
</tr>
<tr>
<td>2568</td>
<td>10</td>
<td>0.1111</td>
<td>0.0179</td>
<td>1.73</td>
<td>1.62</td>
</tr>
<tr>
<td>3386</td>
<td>11</td>
<td>0.1000</td>
<td>0.0144</td>
<td>2.05</td>
<td>1.60</td>
</tr>
<tr>
<td>5788</td>
<td>13</td>
<td>0.0833</td>
<td>0.0101</td>
<td>1.93</td>
<td>1.30</td>
</tr>
<tr>
<td>7890</td>
<td>14</td>
<td>0.0769</td>
<td>0.0086</td>
<td>2.06</td>
<td>1.15</td>
</tr>
<tr>
<td>9062</td>
<td>15</td>
<td>0.0714</td>
<td>0.0075</td>
<td>1.87</td>
<td>1.60</td>
</tr>
<tr>
<td>11074</td>
<td>16</td>
<td>0.0667</td>
<td>0.0065</td>
<td>2.05</td>
<td>1.37</td>
</tr>
<tr>
<td>13346</td>
<td>17</td>
<td>0.0625</td>
<td>0.0057</td>
<td>1.92</td>
<td>1.93</td>
</tr>
</tbody>
</table>

Table 8.4: Convergence test for constant elements and simultaneously increasing $M$ and $N$

![Figure 8.2: Plot of the error for the test with respect to $\Delta t$](image1)

![Figure 8.3: Plot of the error for the test with respect to $h_{\text{max}}$](image2)

Let us first look at the convergence with respect to $\Delta t$, i.e. Figure 8.2. The theory predicts an asymptotic convergence rate of quadratic order. The results show for the considered example that the numerically observed convergence rate approaches the theoretical order of 2 as $\Delta t$ tends to 0.

A look at the convergence rate with respect to $h_{\text{max}}$, i.e. Figure 8.3 also provides a satisfactory result. For the $H^{-1/2}(\Gamma)$-norm an asymptotic rate of $h_{\text{max}}^{3/2}$ is predicted. Taking into account that the measurements are made with a $L^2(\Gamma)$-norm, which reduces the rate to $h_{\text{max}}^1$, and additionally this $L^2(\Gamma)$-norm is approximated panelwise by a one point quadrature, one cannot expect to exactly obtain the theoretical convergence rate. Therefore one has to state that the convergence rate with respect to $h_{\text{max}}$ overall behaves as expected. It is even more clear in the plot, that the convergence rate is between the predicted rates.
8.3.2 Piecewise Linear Elements

The same tests are also run for the case of piecewise linear functions. Again for the first test \( h_{\text{max}} \) is fixed and \( \Delta t \) is decreased. \( M \) denotes now the number of nodes. The spatial boundary mesh is fixed again, like for the test presented in Table 8.3 at \( N = 1192 \). The results are presented in Table 8.5 and Figure 8.4. Again the integration orders are equal for each pair of triangles, however in this case they are set to 4.

<table>
<thead>
<tr>
<th>( M )</th>
<th>( N + 1 )</th>
<th>( \Delta t )</th>
<th>( e )</th>
<th>rate wrt. ( \Delta t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>598</td>
<td>4</td>
<td>0.3333</td>
<td>0.1154</td>
<td>-</td>
</tr>
<tr>
<td>598</td>
<td>6</td>
<td>0.2000</td>
<td>0.0549</td>
<td>1.45</td>
</tr>
<tr>
<td>598</td>
<td>8</td>
<td>0.1429</td>
<td>0.0298</td>
<td>1.82</td>
</tr>
<tr>
<td>598</td>
<td>10</td>
<td>0.1111</td>
<td>0.0184</td>
<td>1.91</td>
</tr>
<tr>
<td>598</td>
<td>12</td>
<td>0.0909</td>
<td>0.0125</td>
<td>1.94</td>
</tr>
<tr>
<td>598</td>
<td>14</td>
<td>0.0769</td>
<td>0.0092</td>
<td>1.81</td>
</tr>
<tr>
<td>598</td>
<td>16</td>
<td>0.0667</td>
<td>0.0071</td>
<td>1.83</td>
</tr>
<tr>
<td>598</td>
<td>18</td>
<td>0.0588</td>
<td>0.0057</td>
<td>1.82</td>
</tr>
<tr>
<td>598</td>
<td>20</td>
<td>0.0526</td>
<td>0.0047</td>
<td>1.71</td>
</tr>
<tr>
<td>598</td>
<td>22</td>
<td>0.0476</td>
<td>0.0040</td>
<td>1.67</td>
</tr>
<tr>
<td>598</td>
<td>24</td>
<td>0.0435</td>
<td>0.0034</td>
<td>1.57</td>
</tr>
<tr>
<td>598</td>
<td>26</td>
<td>0.0400</td>
<td>0.0030</td>
<td>1.43</td>
</tr>
<tr>
<td>598</td>
<td>28</td>
<td>0.0370</td>
<td>0.0028</td>
<td>1.13</td>
</tr>
<tr>
<td>598</td>
<td>30</td>
<td>0.0345</td>
<td>0.0028</td>
<td>0.12</td>
</tr>
<tr>
<td>598</td>
<td>32</td>
<td>0.0323</td>
<td>0.0027</td>
<td>0.10</td>
</tr>
<tr>
<td>598</td>
<td>34</td>
<td>0.0303</td>
<td>0.0027</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 8.5: Convergence test for linear elements and increasing \( N \)

Figure 8.4: Plot of the error for the test with respect to \( \Delta t \)

The rates indicated in the table and the plot show again a similar result as for piecewise constant functions in Table 8.3. At the beginning, the rate is increasing towards the predicted rate of 2. However, the transition to the error dominated by the spatial error is not as sharp as in the last test presented in Table 8.3. Already for \( N > 12 \) a light decrease of the rate can be observed. The definitive transition happens for \( N > 30 \), where the error remains constant.

In the second test case, \( M \) and \( N \) are increased simultaneously by using the rule \( h_{\text{max}}^{5/2} \sim \Delta t^2 \). Our computer facilities do not allow the calculation these tests on meshes with more than
8.4. COMPLEXITY TESTS

8000 triangles in a reasonable time. Therefore the maximal mesh contains 7890 triangles, for which the solving of one single Helmholtz problem takes 10 hours. Again the integration orders are chosen increasingly, like presented in Table 8.2.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$N + 1$</th>
<th>$\Delta t$</th>
<th>$e$</th>
<th>rate wrt. $\Delta t$</th>
<th>rate wrt. $h_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>3</td>
<td>0.5000</td>
<td>0.3663</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>168</td>
<td>4</td>
<td>0.3333</td>
<td>0.1142</td>
<td>2.88</td>
<td>2.27</td>
</tr>
<tr>
<td>310</td>
<td>5</td>
<td>0.2500</td>
<td>0.0850</td>
<td>1.03</td>
<td>0.94</td>
</tr>
<tr>
<td>598</td>
<td>7</td>
<td>0.1667</td>
<td>0.0391</td>
<td>1.91</td>
<td>2.12</td>
</tr>
<tr>
<td>855</td>
<td>8</td>
<td>0.1429</td>
<td>0.0296</td>
<td>1.81</td>
<td>1.54</td>
</tr>
<tr>
<td>1286</td>
<td>10</td>
<td>0.1111</td>
<td>0.0182</td>
<td>1.95</td>
<td>2.45</td>
</tr>
<tr>
<td>1695</td>
<td>12</td>
<td>0.0909</td>
<td>0.0122</td>
<td>1.98</td>
<td>2.95</td>
</tr>
<tr>
<td>2896</td>
<td>16</td>
<td>0.0667</td>
<td>0.0066</td>
<td>1.97</td>
<td>2.26</td>
</tr>
<tr>
<td>3947</td>
<td>19</td>
<td>0.0556</td>
<td>0.0046</td>
<td>2.00</td>
<td>2.53</td>
</tr>
</tbody>
</table>

Table 8.6: Convergence test for linear elements and simultaneously increasing $M$ and $N$

Figure 8.5: Plot of the error for the test with respect to $\Delta t$

Figure 8.6: Plot of the error for the test with respect to $h_{\text{max}}$

Again the predicted convergence rate with respect to $\Delta t$ is achieved. The rate approaches the asymptotic theoretical order of 2. The error with respect to $h_{\text{max}}$ behaves similarly as in the test case for piecewise constant elements presented in Table 8.4. The theory predicts a rate of $h_{\text{max}}^{5/2}$ in the $H^{-1/2}(\Gamma)$-norm, and therefore $h_{\text{max}}^2$ in the $L^2(\Gamma)$ norm. The obtained rates lie overall in this given range, which the plot also shows.

In summary, the implementation is validated by these test results and there is evidence that the theoretical asymptotic convergence rate is sharp.

8.4 Complexity Tests

In this section we will discuss the question of the complexity of the implemented method. As $N + 1$ different Helmholtz problems are solved and the system matrices have the dimension $M \times M$, one expects complexity of $O(NM^2)$ due to the fact that no technique for reducing that complexity is applied in this implementation.

First of all, let us have a look at the time consumption of the various MATLAB routines. The
following information can be obtained by starting the tool ‘Profiler’ in MATLAB. It keeps track of the time spent in the lines of the MATLAB code. A condensed presentation of the result for a relatively small run of the programmed implementation is given in the following plot.

![Figure 8.7: Analysis of the time spent in the different routines](image)

The results show clearly that more than 90 percent of the needed time is used for calculating the entries of the system matrices by applying the implemented quadrature. We point out that the evaluation of the fundamental solution is the dominant time consuming step. Analysing this fact leads to the justification that evaluating the exponential function in MATLAB takes relatively long. Beyond the quadrature routines, the method `solveTimeStep.m` needs a moderate amount of time, as there the temporary file with all information in it is loaded and the arising system of linear equations is solved.

As next step, the complexity with respect to \( N \) is examined for both case of basis functions. One has to keep in mind that for the piecewise constant elements another mesh is used as for the calculation applying the piecewise linear elements. Therefore, the elapsed time cannot be compared to each other.

<table>
<thead>
<tr>
<th>( M )</th>
<th>( N + 1 )</th>
<th>( t_{\text{constant}} ) in min</th>
<th>rate</th>
<th>( M )</th>
<th>( N + 1 )</th>
<th>( t_{\text{linear}} ) in min</th>
<th>rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1192</td>
<td>2</td>
<td>1.51</td>
<td>-</td>
<td>168</td>
<td>2</td>
<td>1.14</td>
<td>-</td>
</tr>
<tr>
<td>1192</td>
<td>6</td>
<td>4.17</td>
<td>0.92</td>
<td>168</td>
<td>6</td>
<td>2.97</td>
<td>0.87</td>
</tr>
<tr>
<td>1192</td>
<td>10</td>
<td>6.34</td>
<td>0.82</td>
<td>168</td>
<td>10</td>
<td>4.77</td>
<td>0.93</td>
</tr>
<tr>
<td>1192</td>
<td>14</td>
<td>8.96</td>
<td>1.03</td>
<td>168</td>
<td>14</td>
<td>6.72</td>
<td>1.02</td>
</tr>
<tr>
<td>1192</td>
<td>18</td>
<td>11.54</td>
<td>1.01</td>
<td>168</td>
<td>18</td>
<td>8.77</td>
<td>1.06</td>
</tr>
<tr>
<td>1192</td>
<td>22</td>
<td>13.40</td>
<td>0.74</td>
<td>168</td>
<td>22</td>
<td>10.53</td>
<td>0.91</td>
</tr>
<tr>
<td>1192</td>
<td>26</td>
<td>16.52</td>
<td>1.25</td>
<td>168</td>
<td>26</td>
<td>12.25</td>
<td>0.91</td>
</tr>
<tr>
<td>1192</td>
<td>30</td>
<td>18.86</td>
<td>0.92</td>
<td>168</td>
<td>30</td>
<td>14.49</td>
<td>1.17</td>
</tr>
<tr>
<td>1192</td>
<td>34</td>
<td>21.20</td>
<td>0.94</td>
<td>168</td>
<td>34</td>
<td>16.16</td>
<td>0.87</td>
</tr>
<tr>
<td>1192</td>
<td>38</td>
<td>23.74</td>
<td>1.02</td>
<td>168</td>
<td>38</td>
<td>18.30</td>
<td>1.12</td>
</tr>
<tr>
<td>1192</td>
<td>42</td>
<td>25.87</td>
<td>0.86</td>
<td>168</td>
<td>42</td>
<td>19.85</td>
<td>0.82</td>
</tr>
</tbody>
</table>

Table 8.7: Complexity test for increasing \( N \)
The result shows the expected dependence of \( \mathcal{O}(N) \) with respect to \( N \). The rate oscillates slightly, as these measured times are not exactly reproducible in a re-run.

The analysis of the dependence of \( M \) is more interesting. Particularly the differences between applying constant integration orders in all directions or applying variable integration orders depending on the pair of triangle. These two test cases are marked with the subscript 'intconst' resp. 'intvar'.

<table>
<thead>
<tr>
<th>( M )</th>
<th>( N + 1 )</th>
<th>( t_{\text{intvar}} ) in min</th>
<th>rate</th>
<th>( t_{\text{intconst}} ) in min</th>
<th>rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>2</td>
<td>0.05</td>
<td>-</td>
<td>0.07</td>
<td>-</td>
</tr>
<tr>
<td>332</td>
<td>2</td>
<td>0.07</td>
<td>0.27</td>
<td>0.24</td>
<td>0.93</td>
</tr>
<tr>
<td>616</td>
<td>2</td>
<td>0.11</td>
<td>0.59</td>
<td>0.77</td>
<td>1.87</td>
</tr>
<tr>
<td>1192</td>
<td>2</td>
<td>0.43</td>
<td>2.09</td>
<td>1.48</td>
<td>0.99</td>
</tr>
<tr>
<td>1706</td>
<td>2</td>
<td>0.69</td>
<td>1.34</td>
<td>2.91</td>
<td>1.89</td>
</tr>
<tr>
<td>2568</td>
<td>2</td>
<td>1.52</td>
<td>2.93</td>
<td>6.74</td>
<td>2.06</td>
</tr>
<tr>
<td>3386</td>
<td>2</td>
<td>2.57</td>
<td>1.91</td>
<td>11.31</td>
<td>1.87</td>
</tr>
<tr>
<td>5788</td>
<td>2</td>
<td>15.47</td>
<td>3.34</td>
<td>32.28</td>
<td>1.96</td>
</tr>
<tr>
<td>7890</td>
<td>2</td>
<td>28.36</td>
<td>1.96</td>
<td>60.56</td>
<td>2.03</td>
</tr>
<tr>
<td>9062</td>
<td>2</td>
<td>37.77</td>
<td>2.07</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11074</td>
<td>2</td>
<td>54.31</td>
<td>1.81</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13346</td>
<td>2</td>
<td>78.21</td>
<td>1.95</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 8.8: Complexity test for increasing \( M \) and piecewise constant basis functions
For constant integration orders the order is chosen as 3. One has to keep in mind that the variable integration orders are partly larger than 3 for $M > 2568$ (cf. Table 8.2).

In the case of constant integration order the dependency is as expected quadratically. The application of variable integration orders complicates the analysis of the complexity as the rate is sometimes much larger than the expected order of 2. This is caused by the change of the integration order from one mesh to the next, as it can be observed in Table 8.2. Everytime the integration order increases in the near field, the rate grows beyond 2, and behaves normally again for the following few test results. Interesting is the difference between the needed time for the different choices of integration order. The test presented here cannot give detailed information about this behaviour as the variable integration order increases simultaneously with $M$ and constant integration order spoils the convergence rate for increasing $M$. However, constant integration orders chosen as the maximum of the variable integration order would lead to problems, which cannot be calculated within a usable time.

The same test is executed for the case of piecewise linear elements. The results are presented in Table 8.9, Plot 8.12 and Plot 8.13. We have chosen 4 as the constant integration order.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$N + 1$</th>
<th>$t_{\text{intvar}}$ in min</th>
<th>rate</th>
<th>$t_{\text{intconst}}$ in min</th>
<th>rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>2</td>
<td>0.06</td>
<td>-</td>
<td>0.22</td>
<td>-</td>
</tr>
<tr>
<td>168</td>
<td>2</td>
<td>0.18</td>
<td>0.86</td>
<td>1.13</td>
<td>2.40</td>
</tr>
<tr>
<td>310</td>
<td>2</td>
<td>0.72</td>
<td>2.29</td>
<td>3.52</td>
<td>2.85</td>
</tr>
<tr>
<td>598</td>
<td>2</td>
<td>2.38</td>
<td>1.81</td>
<td>12.28</td>
<td>1.90</td>
</tr>
<tr>
<td>855</td>
<td>2</td>
<td>11.67</td>
<td>4.45</td>
<td>24.48</td>
<td>1.93</td>
</tr>
<tr>
<td>1286</td>
<td>2</td>
<td>25.77</td>
<td>1.94</td>
<td>56.90</td>
<td>2.07</td>
</tr>
<tr>
<td>1695</td>
<td>2</td>
<td>45.34</td>
<td>2.05</td>
<td>99.23</td>
<td>2.01</td>
</tr>
</tbody>
</table>

Table 8.9: Complexity test for increasing $M$ and piecewise linear basis functions
8.4. COMPLEXITY TESTS

Figure 8.12: Plot of the elapsed time with respect to $M$ applying variable integration orders

The results are principally the same as in the test presented in Table 8.8. One has to take into account that for meshes with more than 2568 triangles the variable integration orders are partly larger than 4 (cf. Table 8.2). With constant integration order, the dependence is as predicted $O(M^2)$. For the variable choice of the orders there are again jumps at the places where the integration order is increased in the near field.

Overall one can state that the implementation scales with $O(NM^2)$ by applying constant integration orders. Choosing the integration order as proposed in the proposition 7.4.2 complicates the analysis, as the elapsed time grows, if the integration order is increased. However, it is important to state that for problems with more than 1000 triangles applying sufficiently large constant integration orders would take by far too much time to calculate and is just useless, as the exactness of the solution is not improved relevantly. In the case where a mesh is not uniform in $h$, i.e. large differences of the diameter of the triangles of the mesh exist, additional tests on the choice of the integration order have to be done.

Analysing the source code of the implementation reveals two routines which would scale with a higher complexity order:

(i) For solving the decoupled Helmholtz system given by (5.56), the direct solver provided by the built-in function \texttt{linsolve} is used. As this solver scales with $O(M^3)$ and $N$ systems of linear equations have to be solved, the complexity of this part of the program is $O(NM^3)$, which is higher than $O(NM^2)$ with respect to $M$.

(ii) The built-in function \texttt{fft} implementing the Fast Fourier Transform scales with $O(N \log N)$ and is execute $O(M)$ times. Therefore in this part results a complexity of $O(MN \log N)$, which is higher than $O(NM^2)$ with respect to $N$.

As the test results presented in the Figures and Plots of this section show that in the range of the employed $M$ and $N$ the behaviour of $O(NM^3)$ cannot be observed. Therefore, for the employed $M$ and $N$ the term of order $O(NM^2)$ dominates the term of order $O(NM^3)$ due to the constant in front of the terms, i.e. it holds for the terms $C_1 NM^2$ and $C_2 NM^3$ and $C_1, C_2 \in \mathbb{R}$ that $C_1 \gg C_2$. The same considerations can be applied to Point (i).
8.5 Acoustic Tests

To check the plausibility of the calculated solution of the wave equation and to perform some tests on the scattering properties of various objects, plane and spherical waves are considered in the case where the speed of sound is \( c = 1 \). Therefore, we go back to the formulation of the problem in Chapter 2. As explained there, we consider in Equation (2.2) the case of an incoming wave \( u_{\text{inc}} \), that gives us then the right-hand side function \( g = \frac{1}{2} u_{\text{inc}} \) on \( \Gamma \). As the convergence of the method is proved for the case, where the incoming wave has not reached the obstacle at \( t = 0 \) and thus \( g \) and sufficiently many of its derivatives are vanishing at \( t = 0 \), both, the classical spherical as well as the plane wave, are modulated by a Gaussian (see Ban09).

Let us look first at the case of an incoming plane wave. The classical formula for a plane wave is

\[
\tilde{u}_{\text{inc}}(x,t) = \cos (\omega \cdot (t - \langle \alpha, x \rangle)),
\]

where \( \omega \in \mathbb{R} \) is the frequency, \( \alpha \in \mathbb{R}^3 \) with \( \| \alpha \| = 1 \) the direction of motion and \( \langle \cdot, \cdot \rangle \) the Euclidean scalar product on \( \mathbb{R}^3 \). This is modulated by a Gaussian with parameters for defining the peak \( A \in \mathbb{R} \) and the speed of decay \( \sigma \in \mathbb{R} \). Thus we get the incoming wave

\[
u_{\text{inc}}(x,t) := \cos (\omega \cdot (t - \langle \alpha, x \rangle)) \cdot \exp \left( -\left( \frac{t - \langle \alpha, x \rangle - A}{\sigma} \right)^2 \right).
\]

Let us show, that this function satisfies the problem \( P_{\text{inc}} \) with vanishing inhomogeneity \( f \) on the full space. Therefore, let us introduce the following notation

\[
r := \omega \cdot (t - \langle \alpha, x \rangle), \quad s := -\left( \frac{t - \langle \alpha, x \rangle - A}{\sigma} \right)^2.
\]

Using the notation \( x = (x_1, x_2, x_3)^T, \alpha = (\alpha_1, \alpha_2, \alpha_3)^T \) and \( i \in \{1, 2, 3\} \), it holds for their derivatives

\[
r_t = \omega, \quad r_{x_i} = -\omega \alpha_i, \quad r_{tt} = 0, \quad r_{x_i x_i} = 0,
\]

respectively for \( s \)

\[
s_t = -\frac{2}{\sigma^2} \frac{t - \langle \alpha, x \rangle - A}{\sigma}, \quad s_{x_i} = 2 \alpha_i \frac{t - \langle \alpha, x \rangle - A}{\sigma^2}, \quad s_{tt} = -\frac{2}{\sigma^2}, \quad s_{x_i x_i} = -\frac{2 \alpha_i^2}{\sigma^2}.
\]

Therefore we get

\[
\begin{align*}
(u_{\text{inc}})_{t} &= -\sin(r) \exp(s) r_t + \cos(r) \exp(s) s_t, \\
(u_{\text{inc}})_{tt} &= -\sin(r) \exp(s) r_t^2 + \cos(r) \exp(s) r_{tt} - \sin(r) \exp(s) s_t r_t + \cos(r) \exp(s) s_{tt} + \cos(r) \exp(s) s_t r_t \\
&= \left( \frac{-r_t^2 + s_t^2 + s_{tt}}{s} \right) \cos(r) \exp(s) + \left( -2 r_t s_t - r_{tt} \right) \sin(r) \exp(s),
\end{align*}
\]

and analogously

\[
\begin{align*}
(u_{\text{inc}})_{x_i} &= -\sin(r) \exp(s) r_{x_i} + \cos(r) \exp(s) s_{x_i} \\
(u_{\text{inc}})_{x_i x_i} &= \left( 3 r_{x_i}^2 + s_{x_i}^2 + s_{x_i x_i} \right) \cos(r) \exp(s) + \left( -2 r_{x_i} s_{x_i} - r_{x_i x_i} \right) \sin(r) \exp(s)
\end{align*}
\]

and

\[
(\Delta u_{\text{inc}}) = \left( \sum_{i=1}^{3} -r_{x_i}^2 + s_{x_i}^2 + s_{x_i x_i} \right) \cos(r) \exp(s) + \left( \sum_{i=1}^{3} -2 r_{x_i} s_{x_i} - r_{x_i x_i} \right) \sin(r) \exp(s).
\]
8.5. ACOUSTIC TESTS

Hence we just examine the coefficients $a_1, a_2, b_1, b_2$ and by virtue of $\|\alpha\| = 1$ we get

$$b_1 = -\omega^2 \|\alpha\|^2 + \|\alpha\|^2 \cdot \left( 2 \frac{t - \langle \alpha, x \rangle - A}{\sigma^2} \right)^2 - \|\alpha\|^2 \frac{2}{\sigma^2}$$

(8.26)

$$= -r_t^2 + s_t^2 + s_{tt} = a_1$$

(8.27)

$$b_2 = 4\omega \|\alpha\|^2 \cdot \frac{t - \langle \alpha, x \rangle - A}{\sigma^2} = a_2.$$  

(8.28)

Therefore it holds

$$(u_{\text{inc}})_{tt} - \Delta u_{\text{inc}} = 0.$$  

(8.29)

The second example is a spherical wave of the form

$$\tilde{u}_{\text{inc}}(r, t) = \frac{1}{r} \cos (\omega(t - r)),$$  

(8.30)

where $r = \|x\|$ and $\omega \in \mathbb{R}$ is the frequency. Again that is modulated by a Gaussian resulting in

$$u_{\text{inc}}(r, t) := \frac{1}{r} \cos (\omega(t - r)) \cdot \exp \left( - \left( \frac{t - A}{\sigma} \right)^2 \right),$$  

(8.31)

with $A, \sigma \in \mathbb{R}$. To show that this wave satisfies the homogeneous scalar wave equation the Laplace operator in spherical coordinates $\frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial u_{\text{inc}}}{\partial r} \right)$ is calculated. The rest follows as in the case of the plane wave. By setting $r = \|x - x_{\text{origin}}\|$, one can choose the origin of the spherical wave, for $x_{\text{origin}} \in \mathbb{R}^3$. In that way the source of the spherical wave can be placed anywhere in space.

As an illustrating example, a Gaussian impulse is scattered at the unit sphere approximated by 92 triangles for $T = 20$ and $N + 1 = 80$. The solution of the wave equation then is calculated on the plane containing the points $(x_1, x_2, x_3)^T \in \mathbb{R}^3$ with $x_3 = 0$ and $-10 \leq x_1, x_2 \leq 10$. It results a film with 80 frames, whereof four are presented now.

Figure 8.14: Frame number 10 of the film  

Figure 8.15: Frame number 20 of the film
In frame number 10, the Gaussian impulse has not reached the obstacle yet (it is hidden beyond the wave), whereas in frame number 20 some parts of the wave are already diffracted. In the next picture one sees the reflected parts going back and the notch in the impulse that the obstacle has caused. Furthermore, a sharp smaller wave follows the impulse in that notch. In frame number 50 one sees the further propagation of the diffracted wave. During the thesis other objects, namely a cuboid and a model of the ancient theater of Epidaurus, are used as scatterers. Some of these results are presented on [Hub10], where the film containing Figures 8.14 - 8.17 is available as well.
Chapter 9

Conclusion

In this thesis, we studied the numerical solution of the wave equation in unbounded domains of the three dimensional space by following the approach presented in [BS08]. In order to discretise in time, the convolution quadrature developed in [Lub88a] and [Lub88b] is applied based on the linear multistep method BDF2. With respect to the space, the discretisation is achieved by a Galerkin boundary element method using the space $S_{m-1,m}$, including the case of piecewise constant ($m = 0$) as well as piecewise linear ($m = 1$) basis functions. Furthermore, the convolution weights arising in the convolution quadrature are approximated in such a way that a decoupling into different Helmholtz problems was possible. After the presentation of the theory about the exact equation, the existence and convergence results for the discretised problem were presented by quoting the existing literature. The presentation of physical properties of waves as well as the motivation of the single layer potential, by a result of classical physics, justifies and illustrates the chosen approach.

Based on the provided theory, a complete MATLAB implementation, including the calculation of the solution in the exterior domain, has been programmed. Using the decoupling into several Helmholtz problems, a method for running the vectorized code on several processors parallelly has been established. The implemented choice of the integration orders is another important step for reducing drastically the needed amount of time for the calculations. These mentioned implementation details allow the solution of the wave equations on much finer meshes. Not affected by these efforts is the overall complexity of $O(M^2N)$. For the validation of the implementation, convergence tests using a right-hand side, for which the analytical solution is known, have been performed. The asymptotic convergence rates given by the theory, which are $\Delta t^2$ with respect to the timestep $\Delta t$ and $h^{m+3/2}$ with respect to the spatial mesh width $h$, have been verified as sharp by measuring the error in a discrete $L^2(\Gamma)$-norm denoted by $\tilde{e}_{L^2(\Gamma)}$. Furthermore, the visualization of the solution in the exterior domain allows to prove the plausibility of the solutions calculated for the scattering of a wave package at a sphere resp. cuboid.

Let us resume the most important parts provided by the MATLAB implementation:

- A package to handle surface triangulations in the three dimensional space has been built. Based on a triangulation generated by Gmsh, which only contains the nodes and the triangles, a whole data structure, including e.g. edges, neighbours and distances between triangles, is generated, which involves all necessary data for executing a vast variety of operations on that mesh.

- As the time dependence included in the wave equation only affects the wave numbers of the decoupled Helmholtz problems, the major part of the implementation can be easily applied on exterior Helmholtz problems in the three dimensional space.
• The quadrature package, developed by Dursun Akay and the author, is applicable for any similar integrand fitting into the framework presented in [SS04] Chapter 5. As quadrature routines are largely decoupled from the other parts of the implementation, there are just few difficulties to apply them on other examples. The presented implementation is rather an example for the use of that quadrature package.

• The convergence tests provide a rule of thumb for the choice of $\Delta t$ and $h$ to attain the optimal convergence rate by a simultaneous decrease of $\Delta t$ and $h$.

Running the different tests on increasingly finer meshes and smaller timesteps revealed the bottlenecks of the current implementation. As mentioned in the discussion of the complexity tests, the evaluation of the exponential function causes a relatively large amount of time. That problem could be reduced by using an approximation of the exponential function, implemented in a file in the program language C and included into MATLAB. However there are differences regarding the limiting factors between the case of piecewise constant and piecewise linear basis functions. Generally in the case of piecewise constant functions the necessary integration orders are smaller. This allows applying meshes with more than 10'000 triangles, which then needs a temporary file with all informations about the mesh and the integration exceeding several gigabytes. This amount of needed space could be reduced by changing the integration package concerning the provided data, as at the moment there is a high redundancy caused by the choice of the integration orders. In the application of the piecewise linear basis functions that problem is dominated by the needed amount of RAM and time. Even with coarser meshes, a single Helmholtz problem cannot be solved within an appropriate period of time on the computer facilities which have been available. In addition not many processors can work parallelly on different Helmholtz problems, as the needed amount of RAM would even be too large for the machine 'Baxter', where the tests were run. Some optimizations to reduce these difficulties would be possible by revising the quadrature package in combination with a stronger linkage to the routine assembling the system matrix.

Furthermore, in [BS08] Chapter 4.2 a possibility for the reduction of the number of Helmholtz problems is presented for special right-hand side functions $g$. In order to decrease the overall complexity for larger $M$ and $N$ applied in the tests, an iterative solver for the systems of linear equations can be applied. Beside these suggestions for an optimization of the implementation avoiding a fundamental change of the method, there exist possibilities of far-reaching modifications to reduce the overall complexity. One possibility is the implementation of a fast multipole technique or the use of a panel-clustering algorithm for solving a single Helmholtz problem. In order to reduce storage costs as well one might reorganize the data structure too, as in the current implementation the storage costs are of order $O(M^2) + O(MN)$. Such propositions can be found e.g. in [BS08] Chapter 4.3 or [SS04], Chapter 7. Furthermore, one could use an iterative solver to solve the arising systems of linear equations. Beside the further development of the method itself, an enhanced visualization of the solution in the exterior domain should be developed, in order to be able to show the propagation of the waves for field points chosen in the exterior volume domain $\Omega$.

All these improvements would increase the usability of the method for applying it to real-world problems. As even in the current stage of the implementation plausible results concerning the scattering of incoming waves at simple three dimensional objects are possible, there is enough evidence that the method is suitable for real problems. In any case, the handling of different boundary conditions and the possibility of modelling material properties will be necessary in order to be able to calculate relevant problems. However, as in the boundary element method exactly the surface of the scatterer, which is essential for the scattering behaviour of an obstacle, is discretised by a mesh, it seems possible that the proposed approach could provide the possibilities to fulfil these requirements.
Bibliography


BIBLIOGRAPHY


