UNIVERSITÄT PADERBORN

Die Universität der Informationsgesellschaft

On Shape Sensitivity Analysis for 3D Time-dependent Maxwell's Equations

Von der Fakultät für Elektrotechnik, Informatik und Mathematik

der Universität Paderborn

zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

– Dr. rer. nat. –

genehmigte Dissertation

von

Maria Schütte

Paderborn, 2016

Gutachter: Prof. Dr. Andrea Walther
Prof. Dr. Nicolas R. Gauger

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Dr. Andrea Walther for her guidance, support, time, patience, and motivation. She has always provided a very pleasant research enviroment for me. For her unwavering support and many helpful discussions, I am truly grateful. I also want to thank her for the opportunity to present my work in the international research community.

I also want to thank Prof. Dr. Nicolas R. Gauger from the TU Kaiserslautern for his interest in my work and for being the second referee of this thesis.

I would like to express my special gratitude and thanks to Dr. Stephan Schmidt for imparting his knowledge and expertise in the field of Shape Calculus. Without his encouragement and enlightening discussions, this thesis would never have taken shape.

My thanks also go to all members of Prof. Andrea Walther's research group: Mladen Banovic, Sabrina Fiege, Benjamin Jurgelucks, Kshitij Kulshreshtha, and Karin Senske for many fruitful discussions and their constant support. I also want to thank the Paderborner Center for Parallel Computing, PC^2, for giving me the opportunity to run my memory intensive code on the Arminius cluster.

I acknowledge the financial support I received from the Federal Ministry of Education and Research (BMBF) within the project HPC-FLiS under the support code 01 IH 11 009.

Finally, I would like to express my gratitude towards my family and friends for the encouragement und support, which helped me in completion of this thesis. Special thanks to my husband Stefan for his moral support during hard times.

Abstract

The main focus of this thesis lies on the detection of objects with arbitrary geometry in a predefined computational domain. The determination of the size or location of the geometric object is of special interest for many real-life applications such as non-invasive material testing and remote sensing. One way to obtain the desired information is the use of direct/classic methods, where, e.g., the object is placed between sender and receiver. Very often the usage of those methods is impossible due to problem-dependent restrictions. In this thesis, we concentrate on a remote sensing approach where electromagnetic waves serve as a possible alternative. These waves are described by Maxwell's equations. The electromagnetic waves propagate through a computational domain and get reflected at the boundary of the geometric object. The reflected waves are recorded by a user-defined sink within the computational domain or at the sensor boundary. Based on the measured reflection the size resp. the location of the object can be reconstructed. This leads to an inverse electromagnetic scattering problem. Naive approaches like the use of an indicator function are rarely successful and often cause a high amount of computation time, which is widely not acceptable for many applications. To overcome such restrictions, the use of Shape Calculus is an appropriate remedy. These methods promise more accurate computations and less computational effort. They differentiate functions w.r.t. the underlying domain. An explicit boundary gradient formulation in the *Hadamard* sense leads to a significant reduction in memory requirement. A generalized method especially for problems restricted by linear state constraints is derived. Subsequently, Maxwell's equations are examined more closely. Challenges like the full time dependence and the high complexity of the problem are solved in this thesis. Based on the derived theoretical results, a solver is developed to solve the problem numerically. A complete implementation consists of an appropriate simulation routine, calculation of the shape gradient including the adjoint equations, and an optimization routine. Various test cases with up to $1.2 \cdot 10^9$ state unknowns are solved to demonstrate the practicability of the proposed approach.

Keywords: Maxwell's equation, shape optimization, adjoints, inverse problems, parameter identification, Galerkin methods

Zusammenfassung

Gegenstand der Arbeit ist die Detektion beliebiger geometrischer Objekte in einem vorgegebenen Berechnungsgebiet. Untersuchenswert ist beispielsweise die Größe und der Ort eines geometrischen Objektes. Dies ist für viele Anwendungen wie z.B. zerstörungsfreie Materialprüfung oder im Bereich der Fernerkundung wichtig. Die Bestimmung gewünschter Eigenschaften ist aufgrund problemspezifischer Einschränkungen oftmals nicht unmittelbar möglich. Um dennoch gewisse Größen des Problems identifizieren zu können, konzentriert sich diese Arbeit auf eine Alternative, welche die Verwendung von elektromagnetischen Wellen beinhaltet. Diese werden durch Maxwell-Gleichungen beschrieben. Diese Wellen propagieren durch ein Gebiet und werden an dem geometrischen Objekt reflektiert. Aufgrund dieser reflektierten Daten werden Rückschlüsse auf den Ort bzw. die Größe des Objektes gezogen. Daher handelt es sich nicht um ein direktes Problem sondern um ein inverses Problem. Naheliegende Ansätze wie die Verwendung von Indikatorfunktionen führen nur bedingt zum Ziel und verursachen zudem eine hohe Rechenzeit, die je nach Anwendung inakzeptabel ist. Diese Problematik motiviert die Verwendung eines Shape Calculus Ansatzes. Die darauf basierenden Methoden versprechen sowohl eine präzisere Rekonstruktion als auch einen geringeren Rechenaufwand. Methoden aus diesem Bereich differenzieren nach einer Form eines geometrischen Objektes. Die *Hadamard*-Randdarstellung des Gradienten führt zu einer erheblichen Speicherplatzeinsparung. Ein generalisierter Ansatz zur Behandlung von Problemen, welche durch lineare Zustandsbedingungen restringiert sind, wurde entwickelt. Im Speziellen wurde der Ansatz auf die Maxwell-Gleichungen übertragen. Sowohl die volle Zeitabhängigkeit als auch die hohe Komplexität des Problems stellen Herausforderungen dar, welche in der Arbeit bewältigt wurden. Auf Basis der in der Theorie erarbeiteten Ergebnisse wurde ein Löser entwickelt, um das Problem numerisch zu lösen. Eine vollständige Implementierung umfasst eine Vorwärtssimulation, die Auswertung des Formgradienten, wobei die adjungierten Gleichungen einfließen, und eine Optimierungsroutine. Verschiedenartige Testfälle mit bis zu $1.2 \cdot 10^9$ unbekannten Zustandsvariablen wurden mittels der vorgeschlagenen Methode gelöst und zeigen die Durchführbarkeit dieser Methode.

Stichworte: Maxwell-Gleichungen, Formoptimierung, adjungierte Gleichungen, inverse Probleme, Parameteridentifikation, Galerkin-Methoden

Contents

List of Figures

List of Tables

Nomenclature

Symbol	SI Unit	Description
μ	$V \cdot s$	magnetic permeability
μ_0	$V \cdot s$	vacuum permeability; $\mu_0 = 1.2566370614 \cdot 10^{-6}$
ϵ	F/m	electric permittivity
ϵ_0	F/m	vacuum permittivity; $\epsilon_0 = 8.854187817 \cdot 10^{-12}$
σ	S/m	electric conductivity

Notation

Ω	geometric domain
$\Gamma = \partial\Omega$	boundary of Ω
$\nabla \times F$	curl F
	(*curl* operator of vector field F)
∇v	grad $v = (\frac{\partial v}{\partial x_1}, \ldots, \frac{\partial v}{\partial x_d})$
	(*gradient* operator of scalar function v of $x \in \mathbb{R}^d$)
$\nabla \cdot F$	div $F = \sum_{i=1}^{d} \frac{\partial F_i}{\partial x_i}$
	(*divergence* operator of vector field $F = (F_1, \cdots, F_d)$ of $x \in \mathbb{R}^d$)
Δv	$\nabla \cdot \nabla v = \nabla^2 v = \sum_{i=1}^{d} \frac{\partial^2 v}{\partial x_i^2}$
	(*Laplace* operator of scalar function v of $x \in \mathbb{R}^d$)
$\mathcal{L}(\Omega, u, \lambda_i)$	Lagrange function depending on state u and adjoint variables λ_i
V	velocity/vector field
$T_\varepsilon(\Omega)[V]$	transformation/perturbation of geometric domain in direction V
$\Omega_\varepsilon = T_\varepsilon(\Omega)$	perturbed domain

Γ_ε	updated boundary
$dF(\Omega; V)$	shape derivative along a vector field V
Df	Jacobian of f
$DH \cdot n$	$\left[\frac{\partial H_i}{\partial x_j}\right] \cdot n = \begin{bmatrix} \frac{\partial H_1}{\partial x_1} & \frac{\partial H_1}{\partial x_2} & \frac{\partial H_1}{\partial x_3} \\ \frac{\partial H_2}{\partial x_1} & \frac{\partial H_2}{\partial x_2} & \frac{\partial H_2}{\partial x_3} \\ \frac{\partial H_3}{\partial x_1} & \frac{\partial H_3}{\partial x_2} & \frac{\partial H_3}{\partial x_3} \end{bmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \begin{pmatrix} \langle \nabla H_1, n \rangle \\ \langle \nabla H_2, n \rangle \\ \langle \nabla H_3, n \rangle \end{pmatrix} = \begin{pmatrix} \frac{\partial H_1}{\partial n} \\ \frac{\partial H_2}{\partial n} \\ \frac{\partial H_3}{\partial n} \end{pmatrix}$
$g(\Gamma)$	shape gradient
γ	trace operator $\gamma : \mathcal{H}^1(\Omega) \mapsto L_2(\Gamma)$, linear and bounded
$\mathcal{H}^{-1}(\Omega)$	dual space of $\mathcal{H}_0^1(\Omega)$
div_Γ	tangential divergence on Γ
∇_Γ	tangential gradient on Γ
$\dot{F}(\Omega; V)$	material derivative of $F(\Omega)$ in Ω in direction of vector field V
$F'(\Omega; V)$	shape derivative of $F(\Omega)$ in Ω in direction of vector field V
$F'(\Gamma; V)$	boundary shape derivative of $F(\Gamma)$ in Γ in direction of vector field V
$\mathcal{D}(\Omega)$	space of infinitely continuously differentiable functions with compact support in Ω
$\mathcal{D}^{-1}(\Omega)$	(dual space of $\mathcal{D}(\Omega)$) space of distributions on Ω

1 Introduction

Non-invasive analyses of interior structures and material properties are daily challenges in various research fields such as aircraft design, biology, medical diagnostics, stealth technology, design of sun cream and networks [2, 9, 69]. Another field, which is related to this topic, is the production of optimized nanoscale particles, where beside the chemical composition, other properties, like the size or the free surface area of the particles have influence and have to be identified [30, 32].

A typical problem setting is the following: A given test setup defines the geometrical resp. computational domain. Within this domain the position and/or size of a geometric object is of special interest. While detecting the targeted object, specific restrictions have to be taken into account. In many cases the measurements are only available on a part of the geometrical boundary of the geometrical domain. This scenario can be formulated as an optimization problem constrained by problem-dependent boundary conditions.

Hence, a description of the problem mainly comprises three components:

- a mathematical model to describe physical states,
- a computer-based simulation,
- an optimization of a user-defined target function.

Different methods can be applied to solve this kind of problem. One class of methods are the direct methods, e.g., light-based methods or special set-ups, where the object of interest is placed between sender and receiver. These strategies are not always feasible for several reasons, e.g., inappropriate geometric shapes of the object or due to physical settings. The considered application in this thesis includes a remote sensing approach, where direct methods are also useless. As alternative electromagnetic measurements might be used, leading to an inverse electromagnetic scattering problem. Details for further reading can be found in [16]. In the context of inverse scattering problems, the target function can be formulated as the mismatch between given, measured data and data simulated in the same test setup. Here, we assume that the data is obtained at a certain part of the surface. A general target function is then given by

$$J(F_{sim}(t)) = \frac{1}{2} \int_{\Gamma_{in/out}} \int_{t_0}^{t_f} \|F_{sim}(t) - F_{meas}(t)\|_*^2 \; dt \; dS,$$

where $F_{sim}(t)$ denotes the simulated data and $F_{meas}(t)$ the measured data over a time interval $[t_0, t_f]$. The measured data is defined by the user and, of course, stays constant during the whole shape identification process. The difference between $F_{sim}(t)$ and $F_{meas}(t)$ is evaluated in a suitable norm $\|\cdot\|_*$. The boundary part denoted by $\Gamma_{in/out}$ represents the surface where the scattering data is obtained. One possible test setup is shown in Fig. 1.1, with Γ_i, $i \in \mathcal{I}$, where $\mathcal{I}$ is an index-set of different boundary conditions and Γ_{incl} the boundary of the inclusion. The variable t lies in the time interval $[t_0, t_f]$ with $0 \leq t_0 < t_f$.

A source is located on the surface $\Gamma_{in/out}$, i.e., a problem-dependent pulse enters the domain over a part of or completely over $\Gamma_{in/out}$. The recorded values are stored in a sink on $\Gamma_{in/out}$ or a sink in the interior. In our application the source serves as sink as well.

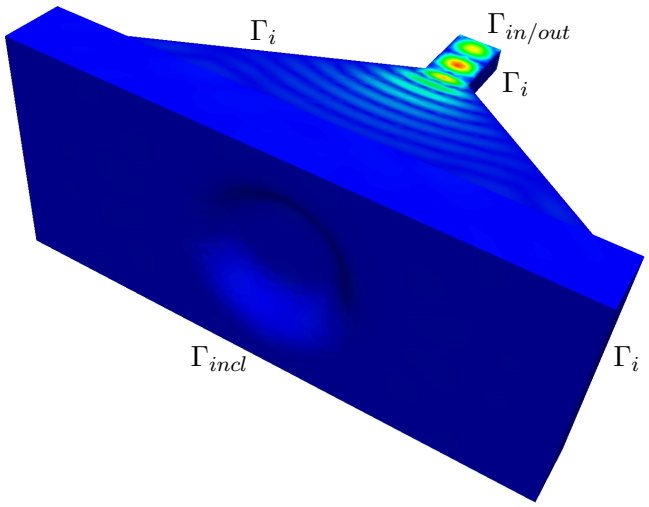

Figure 1.1: One possible test setup of a horn antenna.

The definition of the excitation of the incoming pulse (force) lies in the responsibility of the user. Several setups for the excitation are interesting to analyze: E.g., point source (dipole antenna), plane waves (horn antenna) and a Gaussian pulse.

Naive approaches like Finite-Differences in Time Domain (FDTD) methods [74, 77] in combination with an appropriate indicator function do not lead to desired results due to accuracy reasons and a high amount of computation time [35]. A comparatively new approach in this context is the field of shape optimization. So far, there are rather few contributions on this topic, which probably might be explained by the high complexity of the Maxwell's equations.

The considered problem setting in this thesis is the following: We want to detect the position and/or size of a geometric object, a so-called inclusion or obstacle, in a domain Ω. The boundary of the domain is denoted by $\partial\Gamma = \cup_i \Gamma_i$. On the parts Γ_i of the boundary various boundary conditions, e.g., reflective or non-reflective behaviour on the surface may apply. A sketch of the initial state is shown in the first picture of Fig. 1.2. To determine the shape gradient, a forward simulation as well as an adjoint computation is performed. The solutions to these equations are required for the evaluation of the shape gradient. This shape gradient multiplied by a velocity field V and a normal component n determines the directional derivative of

the given obstacle in direction V. Shape Calculus is used to determine the directional derivative when the domain Ω is subject to a deformation field V. Following the necessary optimality conditions, we conduct a steepest descent algorithm, until a geometry is found with vanishing directional derivatives for all desired directions V. One iteration is illustrated in Fig. 1.2.

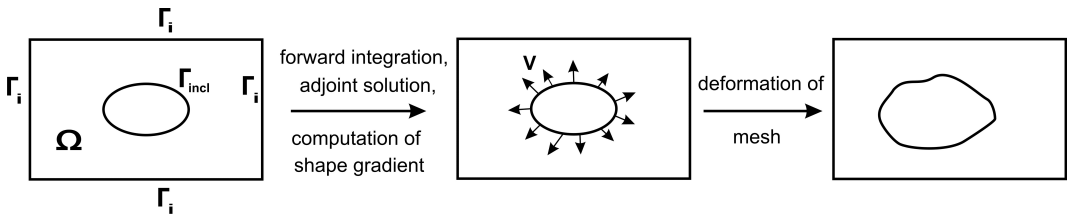

Figure 1.2: Illustration of one possible deformation process of a geometric object. The first picture shows the initial state; the second one shows the velocity field arising from the gradient calculation; the last one sketches the deformation of the obstacle.

1.1 Contributions and Structure of this Thesis

The field of shape optimization is a relatively new field. Especially in the context of electromagnetics, it is of high relevance and hardly analyzed for the time-dependent case. More often time-harmonic formulations of electromagnetic waves are considered, e.g., in [78]. First studies on shape sensitivities are done in [36, 58]. Other studies on shape differentiability properties incorporate the usage of pseudo-homogeneous kernels [17]. However, taking non-stationarity of the problem into account, causes further problems. As discussed in [1, 74] solutions of the inverse problem might not be sufficient.
More recently, works like [53] cover the time-dependence and focus on finding optimal controls for the solution of the Maxwell problem. Concerning the shape sensitivities of the Maxwell problem, investigations by Zolésio [13, 80] are noteworthy. In these works regularity results are the center of interest.

This work focuses on an inverse electromagnetic optimization problem constrained by Maxwell's equations and mixed boundary conditions. For this complex problem, there exists no theoretical work regarding existence or uniqueness of the solution,

yet. A comprehensive analysis of this problem needs further extensive investigations and is not part of the work. We assume the problem to be smooth enough and perform a formal derivation of the shape gradient. The numerical part supports the obtained results.

To the knowledge of the author, there is no comparable work published yet which focuses on the derivation of the shape gradient of the 3D time-dependent Maxwell's equations in Hadamard form. Volume-based expressions for the shape gradient are also possible and used but the explicit expression of the shape gradient on the boundary leads to a significant reduction of memory requirements for the considered application. Additionally, the explicit expression is necessary in the optimization part when using gradient-based methods. To obtain the shape gradient for the Maxwell's equations, results for the Navier-Stokes equations [38, 60, 71] are extended. Likewise numerical analysis of this problem has not been done before. There one has to face other difficulties like the complexity of the fully time-dependent system. One challenge is the determination of six unknowns for every node of the discretization for every timestep. This leads to a high computational effort especially for fine discretized grids. Several test settings are analyzed to demonstrate the applicability of the proposed approach.
The theoretical derivation of the shape gradient for the Maxwell problem as well as the numerical realization are completely new and presented in this thesis.

The structure of the thesis is the following: In Chapter 2, the governing partial differential equations (PDEs), the Maxwell's equations, are introduced. Fundamental theorems and fundamentals of functional analysis are also given to provide all theoretical ingredients for the following sections. In addition to the Maxwell's equations appropriate solution methods are introduced. Chapter 3 starts with an often applied ansatz using an indicator function to solve the considered problem. Despite the simplicity of the described approach, many restrictions make the approach impractical. This fact motivated the author to look for possible alternatives. One promising approach, which turned out to be a good choice, is Shape Calculus. Shape Calculus provides several methods to obtain a shape gradient for the shape optimization. These methods are briefly explained and basic definitions needed for the derivation of the shape gradient for the Maxwell's equations are given. The chapter ends with a section deriving a general formulation for the shape gradient

concerning problems constrained by linear PDEs. A general problem formulation for the considered class of optimization problems forms the last part of the section. Chapter 4 covers the main theoretical results of this thesis: First, a formulation of a Steger-Warming upwind flux is studied to deal with possible discontinuous jumps in the material coefficients and to stabilize the convective problem. Then, appropriate boundary conditions and a reasonable target function are defined to obtain our concrete problem formulation. Corresponding adjoint equations complete the required ingredients for the shape gradient. Next, the derivation of the shape gradient follows and forms one of the main results of this thesis. Several challenges like the appearance of the *curl* operator or the additional dependence of the target function on the normal component have to be solved. In the end, a gradient expression in Hadamard form is derived. Based on this explicit expression, numerical analysis w.r.t. shape optimization is possible. Numerical results are presented in Chapter 5. The forward integration of the state equation is realized using the open source tool dolfin/FEniCS, see, e.g., [45, 46]. FEniCS is a domain specific language and can automatically generate parallel C/C++ code to solve variational problems. A gradient-based optimization algorithm is used for the shape optimization part. The achieved results are very promising and provide a good basis for further research. In Chapter 6, main results are summarized and further questions of interest are formulated.

2

Forward Simulation of 3D Maxwell's Equations in Time Domain

In this chapter, a brief overview of the underlying equations of the simulation problem is presented. Electromagnetic waves, which propagate through the domain are described by Maxwell's equations. These fundamental equations of electromagnetics and appropriate methods to solve these equations numerically are explained.

James Clerk Maxwell (1831 - 1879) was a pioneer in the field of electromagnetics. He formulated the relation of magnetic and electric fields to their sources, charge densities and current densities in two pairs of coupled PDEs. These equations are highly relevant in many applications such as telecommunications, computings, biomedicine and radar technology. In the field of radar technology, antennas, microwave sources and waveguides can be mentioned. In [6] a historical reproduction of Maxwell's work, which was originally published before 1923, can be found.

2.1 Formulation of Maxwell's Equations

Let us consider a bounded domain $\Omega \subseteq \mathbb{R}^3$ and a time interval $(0,T)$. The cylindrical evolution domain is denoted by $Q = (0,T) \times \Omega$, the lateral boundary by $(0,T) \times \Gamma_{in/out}$, and the boundary of an inner inclusion by $(0,T) \times \Gamma_{incl}$.

The time-dependent Maxwell's equations are linear hyperbolic partial differential equations and stated in the differential form as follows

Faraday's Law. This law relates the electric field strength $E(t,x) =: E$ and the time-dependent change of the magnetic flux density $B(t,x) =: B$.

$$\frac{\partial B}{\partial t} = -\nabla \times E \tag{2.1}$$

Ampere's Law. The magnetic field strength $H(t,x) =: H$ is induced by the charge J and the time-changing electric current density $D(t,x) =: D$.

$$\nabla \times H = \frac{\partial D}{\partial t} + J \tag{2.2}$$

From these two laws, one can recognize the immediate correlation of E and H in the propagation process: A varying D is giving rise to H and a varying H effects E.

Gauss' Law for the Electric Field. This law states that wherever the electric current density exists, the divergence of it is equal to the electric charge density ρ. Therefore, the electric charge density can be interpreted as a source or sink for the electric charge.

$$\nabla \cdot D = \rho \tag{2.3}$$

Gauss' Law for the Magnetic Field. This law states the non-existence of magnetic charge.

$$\nabla \cdot B = 0 \tag{2.4}$$

The related material equations, also called constitutive relations, for linear, isotropic and non-dispersive media are defined as follows:

$$D = \epsilon E \tag{2.5}$$

$$B = \mu H \tag{2.6}$$

$$J = J_{source} + \sigma E \tag{2.7}$$

with $\epsilon = \epsilon_0 \epsilon_r(x)$ and $\mu = \mu_0 \mu_r(x)$, where ϵ_0, μ_0 are constants for the permittivity and permeability in vacuum, respectively, and $\epsilon_r = \epsilon_r(x)$, $\mu_r = \mu_r(x)$ describe the relative permittivity and permeability respectively. These parameters may vary over the domain, e.g., depending on the obstacle or the surrounding area. The same holds for $\sigma = \sigma(x)$, which represents the conductivity. The incoming pulse, also called force, is denoted by J_{source}. One example of an incoming excitation in the context of electromagnetic measurements is a modulated Gaussian pulse and can be modelled as

$$\text{Gaussian}(x) = \frac{1}{\tilde{\sigma}\sqrt{2\pi}} \cdot e^{-0.5\left(\frac{x-S_p}{\nu}\right)^2}, \tag{2.8}$$

where $\tilde{\sigma}$ is the square root of the variance $\tilde{\sigma}^2$, S_p is a shift of the pulse and ν is a scaling factor.

General Problem Formulation Our general optimization problem is constrained by Maxwell's equations, which are listed above. Consequently, the electric field E and the magnetic field H have to fulfill Eqs. (2.1) - (2.4). Coupling these equations with the material equations (2.5) and (2.6), leads to

$$\mu \frac{\partial H}{\partial t} = -\nabla \times E \qquad \text{in } \Omega \tag{2.9}$$

$$\epsilon \frac{\partial E}{\partial t} = \nabla \times H - J \qquad \text{in } \Omega. \tag{2.10}$$

Additionally, we define the following boundary conditions for our problem

$$H \times n = 0 \qquad \text{on } \Gamma_{incl} \tag{2.11}$$
$$E(0) = E_0 \qquad \text{on } \Gamma_{in/out} \tag{2.12}$$
$$H(0) = H_0 \qquad \text{on } \Gamma_{in/out}, \tag{2.13}$$

where Eq. (2.11) denotes the reflective boundary condition on the facet of the inclusion and Eq. (2.12) and Eq. (2.13) determine the initial conditions at time $t = 0$.

Boundary Conditions of the Interface in Detail In this thesis, we consider the case that a surface $\tilde{S}$ separates two homogenous materials leading to discontinuous jumps in ϵ, μ and σ. Therefore, certain boundary conditions must be defined. For different materials at the interface, the boundary conditions on $\tilde{S}$ can be expressed as follows [55]

$$\hat{n} \times (E_1 - E_2) = 0 \quad \Longleftrightarrow \quad (E_{1t} - E_{2t}) = 0 \tag{2.14}$$
$$\hat{n} \cdot (B_1 - B_2) = 0 \quad \Longleftrightarrow \quad (B_{1n} - B_{2n}) = 0 \tag{2.15}$$
$$\hat{n} \cdot (D_1 - D_2) = \rho_s \quad \Longleftrightarrow \quad (D_{1n} - D_{2n}) = \rho_s \tag{2.16}$$
$$\hat{n} \times (H_1 - H_2) = J_s \times \hat{n} \quad \Longleftrightarrow \quad (H_{1t} - H_{2t}) = J_s \times \hat{n}, \tag{2.17}$$

where $\hat{n}$ denotes the unit normal vector to the interface $\tilde{S}$ and ρ_s and J_s represent the densities of surface current and charges. The vector-valued field parameter E_1, H_1, D_1, B_1 occur on the other side of the interface than E_2, H_2, D_2, B_2 (cf. Fig. 2.1). Subscript t resp. n indicates the tangential resp. normal part of the field variable. Boundary conditions (2.14) and (2.15) reflect the continuity of the normal components of B and the tangential components of E.

A comprehensive derivation of the boundary conditions for the considered problem can be found in Sec. 4.2.

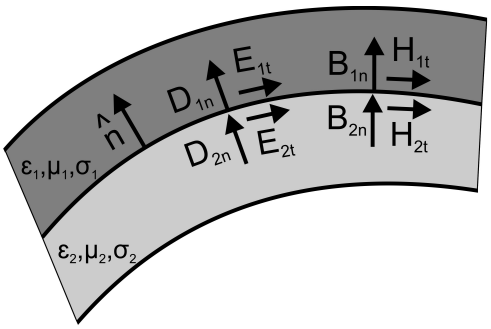

Figure 2.1: Interfacial boundary conditions for two different material distributions.

2.1.1 Useful Theorems

The following theorems (taken from [43]) are used for the theory developed in this thesis. Therefore they are stated here for completeness.

Let u, v be scalar functions, w a vector-valued function of $x \in \mathbb{R}^d$ and n represents the normal vector.

Theorem 2.1 (Divergence theorem)**.**

$$\int_\Omega \nabla \cdot w \; d\Omega = \int_{\partial\Omega} w \cdot n \; dS$$

where Ω denotes the domain, $\partial\Omega$ the boundary and S a closed surface.

Applying the divergence theorem to the product $v \cdot w$, one gets

Theorem 2.2 (Green's formula)**.**

$$\int_\Omega w \cdot \nabla v \; dx = \int_{\partial\Omega} w \cdot nv \; ds - \int_\Omega \nabla w \cdot v \; dx$$

Another useful formula is given by

Theorem 2.3 (Integration by parts for the *curl* operator)**.**

$$\int_\Omega \nabla \times u \cdot v \; dx = \int_\Omega u \cdot \nabla \times v \; dx - \int_{\partial\Omega} (u \times n) \cdot v \; dS$$

2.1.2 Notations and Functional Analysis

Notations One remark regarding the whole thesis has to be made: Sometimes the same names for different variables are used. This can be explained by the fact that some variables have characteristic names in the context they are usually used and this naming convention is maintained in this thesis for consistency with the literature.

In this subsection, we denote a domain by $\Omega \subset \mathbb{R}^d$ with boundary $\Gamma = \partial\Omega$.

The notation $\mathcal{L}(V, W)$ is used for the set of all continuous linear operators from a normed space V to another normed space W. Then, $\ell \in \mathcal{L}(V, W)$ denotes one linear functional. The dual space of V is denoted by V^{-1}.

Another space we introduce is $\mathcal{D}(\Omega)$ representing the space of infinitely continuously differentiable functions with compact support endowed with Schwartz's topology and its dual space $\mathcal{D}(\Omega)^{-1}$, which describes the space of distributions.

Functional Analysis For the following sections of this thesis, the introduction of fundamental definitions of functional analysis is useful. These basics are mainly taken from [5, 43].

Definition 2.4 (Inner product space)**.** *Let V be a linear space over $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$. An inner product $(\cdot, \cdot)$ is a function from $V \times V$ to $\mathbb{K}$, where the following properties hold:*

- *For any $u \in V, (u, u) \geq 0$ and $(u, u) = 0$ if and only if $u = 0$.*
- *For any $u, v \in V, (u, v) = \overline{(v, u)}$.*
- *For any $u, v, w \in V$, any $\alpha, \beta \in \mathbb{K}, (\alpha u + \beta v, w) = \alpha(u, w) + \beta(v, w)$.*

The space V with the inner product $(\cdot, \cdot)$ is called inner product space.

Definition 2.5 (Hilbert space)**.** *A complete inner product space is called Hilbert space.*

Theorem 2.6 (Riesz representation theorem)**.** *Let V be a Hilbert space and $\ell \in V^{-1}$. Then there exists a unique $u \in V$ for which the following equation holds*

$$\ell(v) = (v, u) \qquad \forall v \in V.$$

Additionally, one has

$$\|\ell\| = \|u\|\,.$$

We introduce a d-dimensional vector $\alpha = (\alpha_1, \ldots, \alpha_d)$, called multi-index, where $\alpha_i \in \mathbb{N}_0$. The length of α is given by $|\alpha| = \sum_{i=1}^{d} \alpha_i$. For a given function $v : \mathbb{R}^d \mapsto \mathbb{R}$, the partial derivatives of order $|\alpha|$ can be written as

$$D^\alpha v = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}.$$

Definition 2.7 (Spaces of continuously differentiable functions)**.** *For any $m \in \mathbb{N}$, $C^m(\Omega)$ is a linear space of functions with the property that the functions as well as their derivatives of order less or equal to m are continuous on Ω, i.e.,*

$$C^m(\Omega) = \{v \in C(\Omega) | D^\alpha v \in C(\Omega) \text{ for } |\alpha| \leq m\}.$$

In this context, we can also define C_0^∞ by

$$C_0^\infty(\Omega) = \{v \in C^\infty(\Omega) |\ \text{support } v \subset \Omega\} \text{ with support } v = \overline{\{x \in \Omega | v(x) \neq 0\}}.$$

Definition 2.8 (L^p spaces)**.** *For $p \in [1, \infty)$, $L^p(\Omega)$ is the linear space of measurable functions $v : \Omega \mapsto \mathbb{R}$ such that*

$$\|v\|_{L^p(\Omega)} = \left\{ \int_\Omega |v(x)|^p \, dx \right\}^{\frac{1}{p}} < \infty.$$

The space $L^\infty(\Omega)$ consists of all essential bounded measurable functions $v : \Omega \mapsto \mathbb{R}$ such that

$$\|v\|_{L^\infty(\Omega)} = \inf_{meas(\Omega')=0} \sup_{x \in \Omega \setminus \Omega'} |v(x)| \ < \infty.$$

$meas(\Omega') = 0$ means that Ω' is a measurable set with measure zero.

Definition 2.9 (Locally integrable functions)**.** *Let $1 \leq p < \infty$. A function $v : \Omega \subseteq \mathbb{R}^d \mapsto \mathbb{R}$ is said to be locally p-integrable, $v \in L^p_{loc}(\Omega)$, if for every $x \in \Omega$, there is an open neighborhood Ω' of x such that $\overline{\Omega}' \subseteq \Omega$ and $v \in L^p(\Omega')$.*

Definition 2.10 (Sobolev spaces)**.** *Let k be a non-negative integer and $p \in [1, \infty]$. The Sobolev space $W^{k,p}(\Omega)$ is the set of all functions $v \in L^1_{loc}(\Omega)$ such that for each multi-index α with $|\alpha| \leq k$, the α^{th} weak derivative D^α exists and $D^\alpha v \in L^p(\Omega)$. The norm in the space $W^{k,p}(\Omega)$ is defined as*

$$\|v\|_{W^{k,p}(\Omega)} = \begin{cases} \left(\sum_{|\alpha|\leq k} \|D^\alpha v\|^p_{L^p(\Omega)}\right)^{\frac{1}{p}}, & 1 \leq p < \infty \\ \max_{|\alpha|\leq k} \|D^\alpha v\|_{L^\infty(\Omega)}, & p = \infty. \end{cases}$$

For p=2, we write $\mathcal{H}^k(\Omega) \stackrel{def}{=} W^{k,2}(\Omega)$.

Furthermore we can define $W^{k,p}_0(\Omega)$ as the space of all functions v in $W^{k,p}(\Omega)$ such that

$$D^\alpha v(x) = 0 \quad \text{on } \partial\Omega, \quad \forall \alpha \text{ with } |\alpha| \leq k-1.$$

holds.

Definition 2.11 (Trace operator)**.** *The trace operator $\gamma : C^\infty(\Omega) \mapsto C^\infty(\Gamma)$ is defined by*

$$\gamma v := v|_\Gamma.$$

In the following subsection, solution methods for the Maxwell's equations are introduced. Galerkin methods, described in Sec. 2.2.2 require the weak formulation of the Maxwell's equations. The *strong form* of the variational formulation is given by

$$\int_0^T \int_\Omega v \cdot \left(\mu \frac{\partial H}{\partial t} + \operatorname{curl} E\right) + w \cdot \left(\epsilon \frac{\partial E}{\partial t} - \operatorname{curl} H + \sigma E\right) \, dx \, dt = 0, \tag{2.18}$$

where v, w denote appropriate test functions living in a Sobolev space. The *weak form* of the variational formulation is determined via integration by parts and given

by

$$0 = \int_0^T \int_\Omega \langle v, \frac{\partial H}{\partial t} \rangle + \langle E, \frac{1}{\mu} \operatorname{curl} v \rangle + \langle w, \frac{\partial E}{\partial t} \rangle - \langle H, \frac{1}{\varepsilon} \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E \rangle \, dx \, dt$$
$$- \int_0^T \int_{\partial\Omega \cup \tilde{S}} \frac{1}{\mu} \langle v, E \times n \rangle - \frac{1}{\varepsilon} \langle w, H \times n \rangle \, dS \, dt.$$

The set $\tilde{S}$ comprises all inner interfaces, where no continuity for test- and ansatz-functions is required due to possible jumps at the interface.

2.2 Solution Methods

The analytical solution of Maxwell's equations can only be obtained under simplifying assumptions [20, 42]. Especially for applications, which arise from real world problems, an analytical solution is usually not possible. Therefore, numerical solution methods have to be applied.

Since the main focus of this thesis lies on the optimizational part, we only briefly summarize the main aspects of the solution methods in the remainder of this chapter.

Very often $\mathcal{H}(\text{curl})$ spaces are used for the solution of the PDE. This ansatz can be derived from a transformation of Maxwell's equations as described in [4] such that one of the two unknown variables can be eliminated. Works like [49, 79], also analyze these problems theoretically and numerically. Since this procedure is quite often used, it is presented here in a short form.
To eliminate one of the two unknown variables, we consider the coupled Maxwell's equations Eq. (2.9) and Eq. (2.10). First, Eq. (2.10) is differentiated w.r.t. t, leading to

$$\epsilon \frac{\partial^2 E}{\partial t^2} = \nabla \times \frac{\partial H}{\partial t} - \frac{\partial J}{\partial t}. \tag{2.19}$$

From Eq. (2.9) we have the expression for $\frac{\partial H}{\partial t}$ given by

$$\frac{\partial H}{\partial t} = -\frac{1}{\mu}\nabla \times E$$

and we can obtain

$$\epsilon\frac{\partial^2 E}{\partial t^2} = -\frac{1}{\mu}\nabla \times (\nabla \times E) - \frac{\partial J}{\partial t}. \tag{2.20}$$

Conducting the last steps for Eq. (2.9) analogously, leads to

$$\mu\frac{\partial H^2}{\partial^2 t} + \frac{1}{\epsilon}\nabla \times (\nabla \times H) + \frac{1}{\epsilon}\nabla \times J = 0. \tag{2.21}$$

To obtain a variational formulation of Eq. (2.21), we multiply Eq. (2.21) with an appropriate test function v and have

$$\int_\Omega v \cdot \left(\mu\frac{\partial H^2}{\partial^2 t} + \frac{1}{\epsilon}\nabla \times (\nabla \times H) + \frac{1}{\epsilon}\nabla \times J\right)\, dx = 0. \tag{2.22}$$

Applying the formula for the integration by parts for the *curl* operator yields

$$\begin{aligned}\int_\Omega v \cdot \mu\frac{\partial H^2}{\partial^2 t} + \frac{1}{\epsilon}\nabla \times (\nabla \times H)(\nabla \times v) + \frac{1}{\epsilon}J \cdot \nabla \times v\, dx \\ - \frac{1}{\epsilon}\int_\Gamma (((\nabla \times H) \times n) + (J \times n)) \cdot v\, d\Gamma = 0.\end{aligned}$$

Here, it is notable that the test functions are in the vector-valued space

$$\mathcal{H}(\mathrm{curl}, \Omega) = \{v \in L^2(\Omega)^3, \nabla \times v \in L^2(\Omega)^3\},$$

which only requires tangential continuity over material interfaces. This property corresponds to Eq. (2.14). Therefore, $\mathcal{H}(\mathrm{curl})$ is less smooth than $\mathcal{H}^1$. Using the above problem formulation, $\mathcal{H}(\mathrm{curl}, \Omega)$ conforming elements such as Nédélec elements ([50, 51]) are needed to solve the problem numerically.

In our considered applications, we do not want to solve Maxwell's equations in the first place, but an inverse problem. To model our problem, we need leaving and

incoming information of the domain. Therefore, the characteristic variables are necessary. The solver works on the hyperbolic form of Maxwell's equations. Thus, a DG-Solver is applied instead of a solver using Nédélec elements.

2.2.1 Classical Simulation Method: FDTD

A very classic way to solve (2.9) - (2.10) is the Finite-Differences Time-Domain (FDTD) approach since it is easy to implement and known to be robust [74]. To guarantee robustness, Maxwell's equations are not only solved for the electric (resp. the magnetic) field, but also for the magnetic (resp. the electric) field. For this purpose the coupled Maxwell's equations are solved. The FDTD algorithm was first introduced by Yee in 1966 [77]. The so-called Yee-cell is a rectangular cell in a uniform Cartesian grid where the electric and magnetic field components are sampled. The temporal discretization is performed by the *leapfrog scheme*, which is also a fully explicit scheme. First, one characteristic field is completely computed and stored in memory. Second, the other characteristic field is computed a half timestep later by using the field computations of the first step. More details concerning the theoretical discussion of the algorithm and the numerical realization can be found in [74].
The drawback of this method is the strict requirement of a sufficient small discretization with respect to time and space due to the Courant-Friedrichs-Lewy (CFL) condition [18], which is usually applied in the context of explicit time integration schemes. This condition has to be fulfilled to guarantee convergence while solving certain PDEs. It relates the length of the time step with the length of the spatial step. For the 3D case, the condition is given by

$$C = \frac{u_x \Delta t}{\Delta x} + \frac{u_y \Delta t}{\Delta y} + \frac{u_z \Delta t}{\Delta z} \leq C_{max},$$

where Δt and Δx, Δy, Δz denote the length of the time and spatial step and u_* the magnitude of the velocity in the respective direction. C denotes the Courant number and C_{max} determines the upper bound, which is usually equal to 1 for explicit schemes. In accordance to this condition, several problems in handling arbitrary complex geometries in terms of a sufficient high resolution occur [35]. With this naive finite difference approach, prior knowledge of the solution, which may lead to adaptive discretizations of the domain, cannot be exploited. Additionally, the treat-

ment of discontinuous layers, e.g., in our case discontinuous material coefficients, turned out to be very complicated and error-prone [35].

Finite Volume Methods promise more geometric flexibility since they are able to operate on unstructured and locally refined grids. One possibility to extend this flexibility to higher-order accuracy is the usage of weighted essentially non-oscillatory (WENO) schemes since they are able to provide high order accuracy in smooth complex regions and essentially non-oscillatory transitions in regions with discontinuities like shocks [44]. In this thesis, we concentrate on the usage of high-order techniques, in particular on Galerkin methods.

2.2.2 Galerkin Methods

Galerkin methods consist of an intelligent combination of features from Finite Volume and Finite Element schemes. They operate on the weak formulation. The solution to the PDEs is approximated by the solution of the corresponding variational equation.

In case of discontinuous jumps in the material coefficients of neighboring cells in a given mesh, an appropriate handling is guaranteed by Discontinuous Galerkin (DG) methods. These methods use discontinuous test and trial functions and numerical flux functions.

For a better understanding, we illustrate one variant of the DG-method, the Discontinuous Galerkin Finite Element method, by considering the nonlinear, scalar, conservation law, as stated in [35]

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \qquad x \in [lb, ub] = \Omega \tag{2.23}$$

$$u(x,0) = u_0 \qquad t = 0 \tag{2.24}$$

$$u(lb,t) = g_1(t) \qquad \text{when } f_u(u(lb,t)) \geq 0 \tag{2.25}$$

$$u(ub,t) = g_2(t) \qquad \text{when } f_u(u(ub,t)) \leq 0 \tag{2.26}$$

with the initial condition (2.24) and the boundary conditions (2.25), (2.26) are valid when the boundary is an inflow boundary. The domain is constrained by a lower bound $lb \in \mathbb{R}$ and an upper bound $ub \in \mathbb{R}$. The next step is the approximation of

the domain Ω by K nonoverlapping elements, such that $\Omega \simeq \Omega_h = \bigcup_{k=1}^{K} D^k$ with $D^k = x \in [x_{lb}^k, x_{ub}^k]$ holds. Then the global solution $u(x,t)$ can be represented as

$$u(x,t) \simeq u_h(x,t) = \bigoplus u_h^k(x,t).$$

Thus, a direct sum over piecewise local polynomials $u_h^k(x,t)$ is computed. In accordance to the nodal representation, one defines N_p nodal values as $N_p = N+1$ where N describes the order of the local approximation. The local solutions of order N can be expressed as

$$x \in D^k : u_h^k(x,t) = \sum_{n=1}^{N_p} \hat{u}_n^k(t)\psi_n(x) = \sum_{i=1}^{N_p} u_h^k(x_i^k,t) l_i^k(x),$$

where the first sum computes a product of $\hat{u}_n^k(t)$ as the expansion coefficients and ψ_n as local polynomial basis. This is known as the modal formulation. The second sum represents the nodal form of the DG scheme with a nodal basis containing interpolating Lagrange polynomials $l_i^k(x)$. Then, we can state the local residual

$$x \in D^k : r_h(x,t) = \frac{\partial u_h^k}{\partial t} + \frac{\partial f_h^k}{\partial x}$$

and postulate the residual to vanish in a Galerkin sense

$$\int_{D^k} r_h(x,t) l_i^k(x)\, dx = 0, \quad n \in [1, N_p], \tag{2.27}$$

on all K elements, i.e., that the residual is orthogonal to all test functions $l_i^k(x)$. This leads to N_p equations for N_p local unknowns. Since these considerations are made in a local sense, one has to extend them to a global solution. Therefore, one conducts the spatial integration by parts of Eq. (2.27) leading to

$$\int_{D_k} \left(\frac{\partial u_h^k}{\partial t} l_h^k - f_h^k(u_h^k) \frac{dl_h^k}{dx} \right) dx = - \left[f_h^k, l_i^k \right]_{x^k}^{x^{k+1}} \tag{2.28}$$

$$= - \left[f^*, l_i^k \right]_{x^k}^{x^{k+1}} . \tag{2.29}$$

Eq. (2.28) requires a value f_h^k at the interface, where possibly a discontinuous jump in the material coefficients leads to different local solutions. To decide which local solution or how these local solutions are combined to one global solution at the interface, a problem depending numerical flux $f^* = f^*(u_h^-, u_h^+)$ (Eq. (2.29)) is necessary. Linear numerical fluxes are used in order to preserve the monotonicity of the solution. Higher-order schemes can generate additional extrema. This is captured in the Godunov theorem [27, 28]. One choice of an interface flux $f^* = (au)^*$ is given by

$$(au)^* = \frac{au^- + au^+}{2} + |a|\frac{1-\alpha}{2}(\hat{n}^- u^- + \hat{n}^+ u^+) \qquad \alpha \in [0,1], \tag{2.30}$$

where $a = a(x)$ denotes a piecewise constant and $u(x,t) = u$ the local solution. The computation of the numerical flux au^- is based on the elements of the left hand side and au^+ analogously depends on the elements of the right hand side. The same interpretation holds for $\hat{n}^\pm$ and $u^\pm$. Furthermore the equation $n^- = -n^+$ holds. The decision parameter α determines the characteristic of the interface flux:

- $\alpha = 1$: average of left and right flux (*central/ "natural" flux*)
- $\alpha = 0$: interface flux carries information from where it is coming (*upwind flux*)

Both fluxes are consistent and energy conserving. But the central flux leads to an unstable discontinuous Galerkin discretization whereas the usage of an upwind flux ensures stability. The addition of artificial diffusion in upwind direction is sufficient to stabilize the central flux. For more details see [33].
In this thesis, an appropriate upwind flux for the Maxwell's equations is derived in Section 4.1.

In case that an arbitrary mesh is given without any prior knowledge of the distribution of the material coefficients, this method is the method of choice. However, knowing or having predefined boundary conditions within the mesh, a combination of DG methods and other methods, which assume a continuous distribution of the material coefficients and test- and trial functions makes sense due to memory and efficiency reasons. E.g., Streamline Upwind Petrov Galerkin methods [3, 8, 23] may serve as possible alternative using artificial numerical diffusivity with continuous test- and trial functions.

3

Shape Calculus

Motivated by several real-life applications, e.g., in electromagnetics and aeronautics, the focus of this thesis lies on the differentiation of functions with respect to a geometric domain. The gradients of these functions are called *shape gradients*. While in shape optimization the unknown variables to determine are the structure and/or the form of a geometric object, in conventional optimization the unknowns are a set of functions and/or parameters. Thus, in shape optimization the integration domain is dependent on a variable whereas the integration domain stays fixed in the other case. In the theory of PDEs the term *shape sensitivity analysis* describes the analysis of the sensitivity of the solution of a boundary value problem with respect to the geometry of the domain.

In this chapter, the underlying theory and useful definitions are introduced, which are needed in Chapter 4 for the derivation of the shape gradient of the least-squares objective functional with Maxwell's equations as equality constraints. The fundamental basics are taken from [21, 22, 34, 70, 76].

Motivation to Use Shape Calculus One possible approach to detect geometric objects in a domain is the usage of an appropriate indicator function. An indicator function determines the material distribution within the domain. The proposed approach requires the discretization of the domain into a structured grid and computes the characteristic field variables via a FDTD-algorithm as described in Subsec. 2.2.1.

In this case, the usage of an indicator function can also be interpreted as a level-set-method [56, 57]. One characteristic of level-set-methods is that the contour/boundary of the geometric object is regarded as a root of a multidimensional function. The movement of the boundary can be dependent on different things such as the curvature of the boundary, the gradient or on a velocity field. In our case the movement of the boundary is dependent on the gradient. The geometric object, which we want to track as an example to illustrate the approach, is an ellipsoid in a given computational domain. The zero level set Φ is then given by

$$\Phi = \left\{ (x, y, z) \in \mathbb{R}^3 \mid \frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} + \frac{z^2}{r_z^2} - 1 = 0 \right\}$$

in a three dimensional Euclidean space with r_x, r_y, and r_z as the semi-principal axes. Based on Φ we can model an appropriate if-condition to check whether an arbitrary point is inside or outside the ellipsoid. However, this if-condition naturally leads to difficulties for the computation of the derivatives which are required for a derivative-based optimization algorithm. One remedy is the usage of a smoothed indicator function. A possible definition of an indicator function for an ellipsoid is given by a smoothed Heaviside-function

$$I(x, y, z) = \frac{1}{2} \left(\frac{r + r_{trans}}{\sqrt{(r + r_{trans})^2 + h^2}} - \frac{r - r_{trans}}{\sqrt{(r - r_{trans})^2 + h^2}} \right)$$

with

$$r_{trans} = \sqrt{x_{trans}^2 + y_{trans}^2 + z_{trans}^2}, \qquad r = \sqrt{x^2 + y^2 + z^2}$$

and h small.

Here, the transformation into a spherical coordinate system can be applied

$$\begin{pmatrix} x_{trans} \\ y_{trans} \\ z_{trans} \end{pmatrix} = \begin{pmatrix} r_x \cos\theta \cos\phi \\ r_y \cos\theta \sin\phi \\ r_z \sin\theta \end{pmatrix}, \qquad \theta = \sin^{-1}(\frac{z}{r}), \qquad \phi = \tan^{-1}(\frac{y}{z}).$$

Consequently, the design parameters for the algorithm are the spherical coordinates.

This approach seems to be very simple to solve the problem formulation. However, several restrictions lead to a negative evaluation of the approach:

- lack of accuracy due to smoothed indicator function
  (dependent of choice of h)
- difficult to detect arbitrary geometric objects
  (always adaption of zero level set resp. indicator function necessary
  + prior knowledge of form of geometric object necessary)
- staircase effect due to cartesian cells
- strategy to enforce connectivity of domain required
- FDTD-approach does not allow adaptive discretizations near the object, which leads to high computational effort.

Due to these disadvantages, the usage of an indicator function is no suitable choice to solve our problem. Hence, a more sophisticated approach is necessary. Here, the concept of Shape Calculus comes into play. Despite the fact, that the knowledge of Shape Calculus has to be deepened and extended for our Maxwell problem, Shape Calculus turned out to be the method of choice. For a comprehensive understanding of the developed theory, basic concepts of Shape Calculus will be detailed in the remainder of this chapter.

3.1 Shape Optimization

A very general formulation of a shape optimization problem can be given by

Problem 3.1 (Generalized shape optimization problem)**.**

$$\min_{u(\tilde{\Omega}),\tilde{\Omega}\subset\Omega} J(u(\tilde{\Omega}))$$
$$\text{such that} \quad \begin{aligned} f(u(\tilde{\Omega})) &= 0 \qquad \text{in } \Omega \\ g(u(\tilde{\Omega})) &= 0 \qquad \text{on } \Gamma_i. \end{aligned}$$

The shape optimization problem is constrained by $\Gamma_i, i \in \mathcal{I}$, where $\mathcal{I}$ denotes an index-set of different boundary conditions. Here, $\cup_i \Gamma_i = \partial\Omega$ and $\tilde{\Omega} \subset \Omega$, where Ω is the so-called "hold-all", is part of the geometric domain, where the target function $J(u(\tilde{\Omega}))$ is defined. $J(u(\tilde{\Omega}))$ can be constrained by several state constraints f and several boundary conditions g.

In this thesis, we primary focus on the following class of problems

Problem 3.2 (Generalized problem (divergence form))**.**

$$\min_{\Gamma_{incl}} J(H,E,\Omega) = \frac{1}{2}\int_{t_0}^{t_f}\int_{\Gamma_{in/out}} \phi^\top B(n)\phi \; dS \; dt + \delta \int_{\Gamma_{incl}} 1 \; dS \tag{3.1}$$

$$\text{such that} \quad \begin{aligned} \frac{\partial \phi}{\partial t} + \operatorname{div} F(\phi) &= 0 \qquad \text{in } \Omega \\ F_\Gamma &= 0 \qquad \text{on } \partial\Omega \end{aligned}, \tag{3.2}$$

where $\phi = (H, E)$ denotes the state variable, $B(n)$ a symmetric positive definite matrix, n the outward pointing normal. The parameter δ serves as regularization/penalty parameter for the perimeter of the inclusion. The operator $F(\phi) \in \mathcal{H}^1$ does not contain any differentiation formulas and F_Γ are problem dependent boundary conditions.

Maxwell's equations in divergence form Maxwell's equations as well as the considered target function can be transformed into the generalized form of problem (3.2). The divergence formulation of Maxwell's equations can be derived as

$$(\mu, \varepsilon)\frac{\partial Q}{\partial t} + \nabla \cdot F(Q) = J \tag{3.3}$$

$$Q = \left(\; H_x, H_y, H_z, E_x, E_y, E_z \; \right), \qquad J = \sigma E, \tag{3.4}$$

where Q contains the time-dependent variables, F the flux terms and J the source terms. $F(Q)$ is defined as

$$F(Q) := \begin{pmatrix} R_1 E & R_1^\top H \\ R_2 E & R_2^\top H \\ R_3 E & R_3^\top H \end{pmatrix}, \tag{3.5}$$

where R_i denotes the Levi-Civita permutation tensor:

$$(R_i)_{jk} = \epsilon_{jik}, \quad 1 \leq i \leq \mathrm{d},$$

i.e., $\epsilon_{jik} = 0$ if any index is repeated, $\epsilon_{jik} = 1$ if it is a cyclic permutation and $\epsilon_{jik} = -1$ if it is an anticyclic permutation, e.g.,

$$R_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

F can be decomposed into $F = \hat{\imath} f + \hat{\jmath} g + \hat{k} h$, where

$$f = \left(\; R_1 E, R_1^\top H \; \right) = \begin{bmatrix} 0 & R_1 \\ R_1^\top & 0 \end{bmatrix} \begin{pmatrix} H \\ E \end{pmatrix}, \hat{\imath} = (\mathbb{1}_{3\times 3}, 0_{3\times 3}, 0_{3\times 3}),$$

and analogously

$$g = \left(\; R_2 E, R_2^\top H \; \right), \hat{\jmath} = (0_{3\times 3}, \mathbb{1}_{3\times 3}, 0_{3\times 3}),$$
$$h = \left(\; R_3 E, R_3^\top H \; \right), \hat{k} = (0_{3\times 3}, 0_{3\times 3}, \mathbb{1}_{3\times 3}).$$

Hence, the divergence of F can be obtained as

$$\begin{aligned}\operatorname{div}(F(Q)) &= \frac{\partial}{\partial x}\begin{bmatrix} 0 & R_1 \\ R_1^\top & 0 \end{bmatrix}\begin{pmatrix} H \\ E \end{pmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} 0 & R_2 \\ R_2^\top & 0 \end{bmatrix}\begin{pmatrix} H \\ E \end{pmatrix} \\ &\quad + \frac{\partial}{\partial z}\begin{bmatrix} 0 & R_3 \\ R_3^\top & 0 \end{bmatrix}\begin{pmatrix} H \\ E \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial}{\partial x}R_1E + \frac{\partial}{\partial y}R_2E + \frac{\partial}{\partial z}R_3E \\ -\frac{\partial}{\partial x}R_1H - \frac{\partial}{\partial y}R_2H - \frac{\partial}{\partial z}R_3H \end{pmatrix} \\ &= \begin{pmatrix} \operatorname{curl} E \\ -\operatorname{curl} H \end{pmatrix}.\end{aligned}$$

An appropriate formulation of the target function is given in Sec. 4.5.

3.2 Shape Optimization Design Cycle

In Fig. 3.1 a very general scheme of the shape optimization design cycle is shown. First of all, an initial geometric object has to be guessed and placed into the computational domain. Then, the domain and possibly inner geometric objects are discretized in a suitable manner, which usually depends on the considered application. A forward simulation is conducted and especially in our formulation the primal data is compressed to the boundary.
After that, a problem-dependent target function is evaluated. Next, the adjoints are computed and take part in the evaluation of the shape gradient. To avoid discontinuities of the mesh and to speed up the optimization, a smoothed descent direction is computed in the space of continuous functions. The smoothing can be interpreted as a Newton-step for a circumference penalization. After these steps, one or several convergence criteria, e.g., with respect to the reduction of the target functional or with respect to the norm of the gradient, are checked. If the defined convergence criteria are satisfied, the iteration stops with a final shape Γ^*. Otherwise, the domain discretization and the discretization of the geometric object, in particular the discretization of the boundaries, will be updated, the mesh will be moved, and the steps from the forward simulation onwards will be repeated until a convergence criterion is reached.

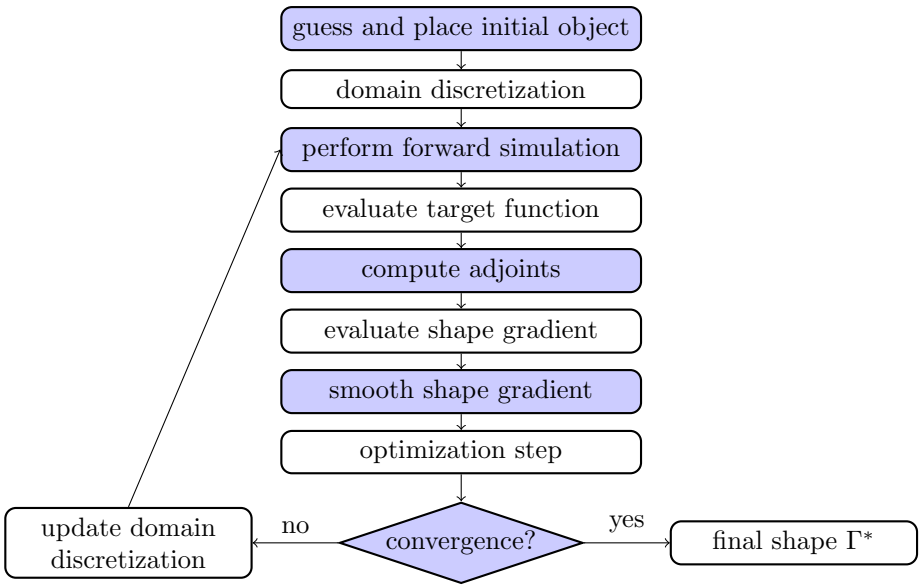

Figure 3.1: Scheme of shape optimization design cycle.

In the next section, the Brachistochrone problem is presented. It is not only one of the first problems in the context *calculus of variations* but also an illustrative example in the field of shape optimization. More similar introductory problems are described, e.g., in [10, 11, 75]. The solution of the Brachistochrone problem should not be in the focus of this section but the proposed task should illustrate that shape optimization problems have been known for a long time although they have been solved via other methods. Furthermore, the example should demonstrate that even simple problems can be formulated as a shape optimization problem.

Brachistochrone problem Johann Bernoulli posed the Brachistochrone problem in 1696. The task was to find the optimal shape of a curve $y(x)$, which provides the least time to travel (without friction) from a point (a, y_a) to a point (b, y_b) in a constant gravity field g. First, one might think of a straight line as a minimizer. However, in fact a cycloid turns out to be the optimal curve. A cycloid is the curve traced by a point attached to a circular wheel travelling along a straight line. A sketch of the problem is shown in Fig. 3.2. The problem was solved by Johann Bernoulli, Jacob Bernoulli, Isaac Newton, Gottfried Leibniz, and Guillaume de L'Hôpital applying

different approaches. Nevertheless, all of them came to similar solutions.

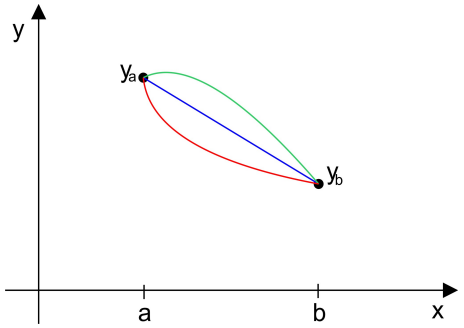

Figure 3.2: Brachistochrone problem. Possible paths from (a, y_a) to (b, y_b). Red curve (cycloid) needs the least time for the transit.

Thereof, one can formulate a shape optimization problem with the target to find a minimal curve $y(x) \in \mathbb{R}^2$ with fix start and end-point.

Problem 3.3 (Shape optimization problem: Brachistochrone)**.**

$$\begin{aligned} &\min_{\Gamma} y(\Gamma) \\ &\textit{such that} \quad y(a) = y_a, \quad y(b) = y_b. \end{aligned}$$

In order to solve the problem, one employs basic equations and laws from mechanics. The time to travel between two points is given by

$$T_{ab} = \int_a^b \frac{ds}{v},$$

where s denotes the arc length and v the speed. For a rectifiable curve and infinitesimal ds, the following equation holds

$$ds^2 = dx^2 + dy^2 = (1 + y'(x)^2)dx^2. \tag{3.6}$$

Assuming the total energy of the point mass to be zero and exploiting the energy

law, which involves the kinetic energy E_{kin} and the potential energy E_{pot},

$$E_{kin} = E_{pot} \Leftrightarrow \frac{1}{2}mv^2 = m \cdot g \cdot y(x),$$

one arrives at an expression for v of the form

$$v = \sqrt{2gy(x)}.$$

Now, we can transform the shape Brachistochrone problem into a variational problem with fixed limits:

Problem 3.4 (Variational problem: Brachistochrone)**.**

$$\min_{y(x)} \int_a^b \frac{\sqrt{1+y'(x)^2}}{\sqrt{2gy(x)}}\, dx$$
$$\textit{such that} \quad y(a) = y_a, \quad y(b) \quad = y_b.$$

This problem might be solved, e.g., by the Euler-Lagrange differential equation. In the end, one arrives at a solution for $y(x)$, which describes a cycloid. A possible parametric representation is given by

$$x(t) = \frac{k}{2}(t - \sin t), \qquad y(t) = \frac{k}{2}(1 - \cos t),$$

where k describes the diameter of a circular wheel. The trigonometric functions describe the circular motion. This cycloid is called Brachistochrone.

3.3 Basics of Shape Calculus

In this section, the most common concepts to obtain shape derivatives are explained. One term, which is indispensable in this context, is the *vector/velocity field*. The vector field is a continuous differentiable map, which operates on the whole computational domain. Hence, it contains a definition for the movement for *every* point. Therefore, it can also be interpreted as a domain perturbation. Illustrations of the vector field for different perturbations methods are given by Fig. 3.3 and Fig. 3.4.

Lagrange Method One approach to obtain the shape gradient of a PDE-constrained problem is based on the Lagrangian formulation. This technique was first introduced by Cea in 1986 [14]. It can be considered as a comparatively simple method to obtain the shape derivative since one has to define the Lagrangian formulation. The Lagrangian function is denoted by

$$\mathcal{L}(\Omega, u, \lambda_1, \lambda_2) = J(u(\tilde{\Omega})) + \lambda_1 f(u(\tilde{\Omega})) + \lambda_2 g(u(\tilde{\Omega})),$$

where u denotes the state and λ_1, λ_2 the adjoint variables. The state constraints are denoted by $f(u(\tilde{\Omega}))$ and the boundary conditions by $g(u(\tilde{\Omega}))$. Then, the shape derivative in the direction of the velocity field V is given by

$$dJ(\Omega; V) = \frac{d}{d\varepsilon}(\Omega_\varepsilon, u_\varepsilon, \lambda_{1_\varepsilon}, \lambda_{2_\varepsilon}; V).$$

In order to arrive at an explicit expression for the gradient, the Lagrangian is analyzed for saddle points and an adjoint system is solved. Applying this method, it is assumed that u and λ_1, λ_2 are differentiable with respect to the domain. Therefore, it is also called a *formal* method. The required differentiability might cause problems for certain applications. For the considered Maxwell problem, the mentioned problem occurs as well since, e.g., the state variable H is defined for the space where $curl(H)$ lives, but not necessarily on the whole domain.

For the following two fundamental methods we can define the family of perturbed domains Ω_ε

$$\Omega_\varepsilon = T_\varepsilon(\Omega) = \{T_\varepsilon(x) : x \in \Omega\}.$$

Analogously, we have for the perturbed boundary $\Gamma_\varepsilon = \tilde{T}_\varepsilon = \{\tilde{T}_\varepsilon(x) | x \in \Gamma\}$.
T_ε and $\tilde{T}_\varepsilon$ are bijective transformations of $\mathbb{R}^n$ with $T_0 = I$. Here, we assume that the initial domain $\Omega = \Omega_0$ and all perturbed domains Ω_ε have the same topological properties and the same regularity.
Another relevant definition is the definition of a shape functional. It is given by

Definition 3.1 (Shape functional)**.** *A shape functional J is a map*

$$\begin{aligned} J(\cdot) : D &\to \mathbb{R} \\ \Omega &\mapsto J(\Omega), \end{aligned}$$

where D denotes the "hold-all" of admissible domains with $\Omega \in D \subset \mathcal{P}(\mathbb{R}^d)$, which is a domain of class $C^k, k \geq 1$. $J(\cdot)$ is associated to the unperturbed domain.

Perturbation of Identity For a shape functional $J(\Omega)$ in a given domain $\Omega \subset \mathbb{R}^n$, one can construct a family of transformations

$$T_\varepsilon[V](x) \stackrel{\text{def}}{=} id_\Omega(x) + \varepsilon V(x) = x + \varepsilon V(x), \qquad x \in \Omega \qquad \varepsilon \in [0, \tilde{\varepsilon})$$

for small perturbations $0 \leq \varepsilon < \tilde{\varepsilon}$ of Ω.

In Fig. 3.3, a sketch of the perturbation of the initial geometric domain Ω is given. The vector v is an element of the velocity field V and points in the direction of the normal of the boundary. The description of the concrete deformation of the boundary that we are using for our application, is denoted in the update scheme of the boundary (3.8). The velocity method, in the literature also called *speed method*, is based on this perturbation and can be regarded as a generalization of the described method.

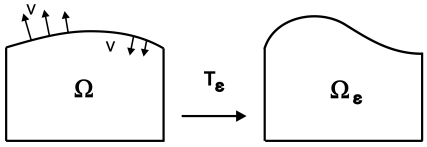

Figure 3.3: 2D representation of perturbation of the domain.

Velocity Method The velocity method covers additional time dependence in the vector fields. Therefore these fields are also called *nonautonomous* vector fields. The *flow* (solution) of the differential equation $x(\varepsilon, X) = x(\varepsilon)$ is defined as

$$\frac{dx}{d\varepsilon}(\varepsilon) = V(\varepsilon, x(\varepsilon)), \qquad \varepsilon \in [0, \tilde{\varepsilon}), \qquad x(0, X) = X,$$

where $V : [0, \tilde{\varepsilon}] \times \Omega \mapsto \mathbb{R}^n$ denotes the vector field.
Then the perturbed domain is given by

$$T_\varepsilon[V](X) \stackrel{\text{def}}{=} x(\varepsilon, X) : \mathbb{R}^n \mapsto \mathbb{R}^n \qquad \varepsilon \in [0, \tilde{\varepsilon}), \qquad X \in \Omega.$$

A sketch of this method can be found in Fig. 3.4.

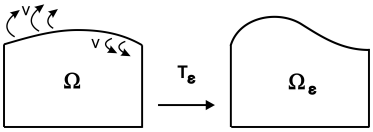

Figure 3.4: 2D representation of velocity method.

Remark 3.2. *Under suitable assumptions, the method of perturbation of identity and the velocity method yield the same first order shape derivative. An additional acceleration term appearing in the second order shape derivatives using the method of perturbation of identity, leads to different results for higher order shape derivatives [22].*

A formula to transform the perturbed domain Ω_ε back to the original form Ω is given by

$$J_\varepsilon = \int_{\Omega_\varepsilon} f \, dx = \int_{\Omega} f(T_\varepsilon) |\det (DT_\varepsilon)| \, dx, \tag{3.7}$$

where D denotes the Jacobi operator. This transformation is the general analogon to Eq. (3.6) of the Brachistochrone problem.

Motion of the Geometry Recalling basic concepts of nonlinear optimization [54], one can find several parallels concerning the determination of a new iterate, e.g., considering the steepest descent method, where a new iterate is computed through $x_{new} = x_{old} - \alpha \nabla F(x_{old})$. A new point x_{new} is computed as a composition of the old point x_{old} minus a product of an appropriately chosen step size α with $\nabla F(x_{old})$, which is the gradient of the function F at the former point x_{old}.

In the field of shape optimization, Zolésio also proposed an explicit formula of the

update of the boundaries depending on the explicit formulation of the gradient, as will be derived in Sec. 4.5. Obviously, this is a gradient method applied to a non-convex problem, which often leads to local minima and therefore the choice of the initial geometry Ω_0 is of high relevance.

Definition 3.3 (Update of boundary)**.** *For Γ being compact, there exists $\tilde{\varepsilon} > 0$ such that for any $\varepsilon, |\varepsilon| < \tilde{\varepsilon}$, the following equation for the update of the boundary holds*

$$\Gamma_\varepsilon = \Gamma + \varepsilon g(x)n(x) = \{y | y = x + \varepsilon g(x)n(x) \text{ for } x \in \Gamma\}. \tag{3.8}$$

Thus, an update of a point x on the boundary Γ is determined by a shift of x in the direction of the outward pointing normal $n(x)$ multiplied with the gradient $g(x)$ and a step size ε.

An illustration of the update of the boundary is given in Fig. 3.5.

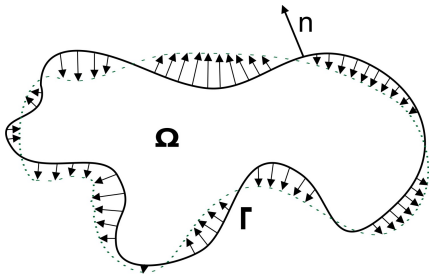

Figure 3.5: Update of the boundary Γ directly depending on the gradient g in the direction of the normal n.

Important to notice is that the updated shape of the boundary Γ_{update} depends on the former shape Γ_{current} and not directly on the initial shape Γ_0, thus
$\Gamma_{\text{update}} = \Gamma_{\text{current}} + \varepsilon g n$.
The step size ε can be determined via various step size control methods adapted from [54].
Applying various methods, one can modify the update formula of the geometric variable. Various approaches can be followed, e.g., the definition of a Newton-type speed function, which includes calculating second Eulerian derivatives for shape functionals. This approach is discussed in [37] in the context of image segmentation. Another very successful technique is a gradient smoothing technique of Newton-type.

This gradient smoothing method promises to accelerate the optimization process similar to approximative Newton schemes. A possible approximation is using the Hessian of the perimeter penalization in problem (3.2), which leads to a classical Laplacian-type smoothing also found in so-called Sobolev methods [59]. More details on smoothing can be found in Sec. 3.6.

3.4 Shape Gradient

Some preliminary definitions regarding the definition of the shape gradient are necessary. Central concepts are defined in this section and are also taken from [21]. For the derivation of the shape gradient of the least-squares objective functional with Maxwell's equations as equality constraints, the *local shape derivative* $u'(\Omega; V)$, the *material derivative* $\dot{u}(\Omega; V)$, the *boundary shape derivative* $u'(\Gamma; V)$, and the *adjoint system* are required. The definition of the *shape derivative* $dJ(\Omega; V)$ is also necessary for the definition of the shape gradient in Hadamard form. With a deformation via *perturbation of identity* the shape derivative of a domain functional $J(\Omega)$ is defined as

Definition 3.4 (Shape derivative)**.** *The shape functional $J : \mathcal{P}(\Omega) \to \mathbb{R}$ is called shape differentiable along a vector field V, also called velocity field, where V indicates the direction of the derivative, if the limit*

$$dJ(\Omega; V) := \lim_{\varepsilon \to 0^+} \frac{J(\Omega_\varepsilon) - J(\Omega)}{\varepsilon} \tag{3.9}$$

exists and the mapping $V \mapsto dJ(\Omega; V)$ is continuous and linear. Here, $\Omega_\varepsilon = T_\varepsilon(\Omega)[V]$ denotes the perturbed domain as described above.

The shape derivative $dJ(\Omega; V)$ describes the change in the functional J depending on the perturbed domain $\Omega_\varepsilon[V]$.
The shape derivative is also known as *Eulerian semi-derivative*. Usual rules of differential operators such as product rule or chain rule also hold for this operator.

Remark 3.5 (Normal component)**.** *Under the assumptions that the boundary* Γ *is Lipschitz continuous and* $\operatorname{div}(f \cdot V(0))$ *exists, the shape derivative only depends on the normal component of the deformation on the boundary of the reference domain [70].*

For the derivation of the shape gradient the material derivative is needed. The material derivative describes the change of the quantity of one specific property depending on a velocity field V at a certain time at a certain place. The material derivative is pointwise defined and if it exists at $x \in \Omega$, it is defined as

Definition 3.6 (Material derivative)**.** *For a given family of functions* $u_\varepsilon : D \mapsto \mathbb{R}$*,* D *= "hold-all", on a perturbed domain the material derivative is defined as*

$$\dot{u}(x;V) := \frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} u_\varepsilon(x_\varepsilon) \tag{3.10}$$

$$:= \lim_{\varepsilon \to 0^+} \frac{u_\varepsilon(T_\varepsilon(x)) - u(x)}{\varepsilon}. \tag{3.11}$$

The material derivative is also known as *Lagrangian derivative* or *total derivative.*

Definition 3.7 (Local shape derivative)**.** *For a given family of functions* $u_\varepsilon : D \mapsto \mathbb{R}$*,* D *= "hold-all", the local shape derivative can be read as partial derivative and is expressed by*

$$u'(x;V) := \frac{\partial}{\partial \varepsilon} u_\varepsilon(x). \tag{3.12}$$

The local shape derivative of $J(\Omega)$ in the direction of V and the material derivative are related as follows

$$\dot{u}(x;V) = u'(x;V) + \nabla u(x) \cdot V(0). \tag{3.13}$$

Here, the physical meaning of the material derivative can be explained. The first part, the shape derivative, describes the local part, in particular the sensitivity of a specific property at a fixed place. The second part describes the convected part, which contains additional sensitivity due to a spatial movement.

In this thesis, the Hadamard representation of the shape gradient plays a central role. The following definition states the shape derivative as a scalar distribution on the boundary, which only depends on the normal part.

Definition 3.8 (Hadamard formula)**.** *Let $J : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ be a shape functional which is shape differentiable at every domain Ω of class $C^k, \Omega \subset D$. Additionally, let $\Omega \subset D$ be a domain with boundary of class C^{k-1}.*
There exists the scalar distribution

$$g(\Gamma) \text{ in } \mathcal{D}^{-k}(\Gamma) \tag{3.14}$$

with the vector distribution $G(\Omega) \in \mathcal{D}^{-k}(\Omega; \mathbb{R}^n) = \mathcal{D}^k(\Omega; \mathbb{R}^n)'$ which is the gradient of the shape functional J at Ω. It is expressed by

$$G(\Omega) = \gamma_\Gamma^*(g \cdot n), \tag{3.15}$$

where γ_Γ denotes the trace operator of $g \cdot n$ on Γ and γ_Γ^* the adjoint of γ_Γ.

Directly connected to this definition, the following formula is often used for convenience.

Remark 3.9. *If g is integrable on Γ, the shape gradient g can also be expressed as [70]*

$$dJ(\Omega; V) = \int_\Gamma g(x) \langle V(0, x), n(x) \rangle_{\mathbb{R}^n} d\Gamma. \tag{3.16}$$

This definition will be used throughout the whole thesis. The explicit form of g is required for implementation purposes, leads to significant data reduction and prevents checkpointing.

3.4.1 Tangential Differential Calculus

In this section, we briefly summarize one of the basic ideas of shape optimization, in particular the main concepts of tangential differential calculus. Classically, these notions are introduced via a smooth extension R of a scalar function $r : \Gamma \mapsto \mathbb{R}$ in

a tubular neighborhood of Γ. This tubular neighborhood can be defined as

$$U_k(\Gamma) = \{x \in \mathbb{R}^d : b(x) < k\}, \qquad k > 0, \quad k \in \mathbb{R}$$

with the oriented distance function $b = b_\Omega$. Assuming Ω to be an open domain of class C^2 in $\mathbb{R}^d$ with compact boundary Γ, then there exists $h > 0$ such that $b = b_\Omega \in C^2(U_{2h}(\Gamma))$ [22]. The idea of extending boundary values can be applied in this context since the expression of the tangential gradient on Γ is independent of the choice of the extension R. A projection of a point x onto Γ is given by a projection function $p = p_\Gamma$ and is defined as

$$p(x) = x - b(x)\nabla b(x).$$

Furthermore $\nabla(r \circ p)|_\Gamma$ denotes the tangential gradient $\nabla_\Gamma r$ (cf. Def. 3.10). The extension $r \circ p$ is called *canonical extension*. The gradient of $r \circ p$ is tangent to the level sets of b, e.g., for $d = 2$ the gradient is tangent to the isolines of b. Proofs for the last two statements are given in [22]. The orthogonal projection operator of a vector onto a tangent plane $T_{p(x)}\Gamma$ is defined as

$$P(x) = I - \nabla b(x)\nabla b(x)^\top.$$

With a smooth extension, the Euclidean differential calculus in the ambient neighborhood of Γ can be used. E.g., for a normal extension of $r \in C^1(\Gamma)$, one basically determines normals on the boundary Γ and extends values of Γ of a certain area in a tubular neighborhood. This tubular extension is also illustrated in Fig. 3.6.

The tangential gradient, also referred to as the surface gradient, is calculated along a surface. It is similar to the conventionally known gradient without the normal component of the surface. Therefore, it is tangential to the surface. The definition of the tangential gradient uses a smooth extension R of $r \in C^1(\Gamma)$ in the neighborhood of Γ and is formulated as

Definition 3.10 (Tangential gradient)**.** *For Γ being compact, the tangential gradient of $r \in C^1(\Gamma)$ in a point of Γ reads as*

$$\nabla_\Gamma r = \nabla R|_\Gamma - \frac{\partial R}{\partial n} n. \tag{3.17}$$

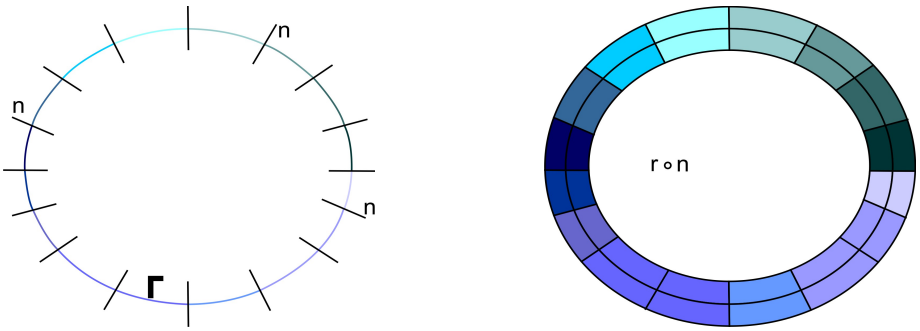

Figure 3.6: Illustration of tubular extension. Left: boundary Γ of ellipse with varying values $r(x)$, $x \in \Gamma$; not-oriented normals given in black. Right: related normal extension in a tubular neighborhood of Γ.

The tangential gradient is the orthogonal projection $P(x)\nabla R(x)$ of $\nabla R(x)$ onto the tangent plane to Γ at x, that is $T_x\Gamma$. A sketch of a tangent plane of an ellipsoid is shown in Fig. 3.7.

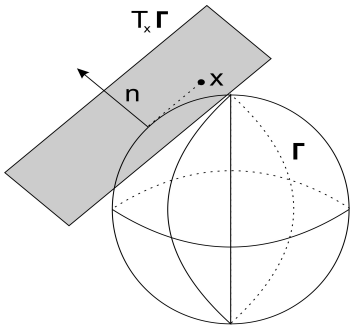

Figure 3.7: Tangent plane of an ellipsoid.

Definition 3.11 (Tangential Jacobian matrix)**.** *The tangential Jacobian matrix of a vector function $r \in C^1(\Gamma)^M, M \geq 1$, is defined as follows*

$$(D_\Gamma r)_{ij} = (\nabla_\Gamma r_i)_j \text{ or } D_\Gamma r = D(r \circ p)|_\Gamma,$$

where D denotes the Jacobian and $r \circ p$ the canonical extension of r.

Definition 3.12 (Tangential divergence)**.** *The tangential divergence for a vector function $r \in C^1(\Gamma)$ is defined as*

$$\operatorname{div}_\Gamma r = \operatorname{div}(r \circ p)|_\Gamma,$$

where $r \circ p$ denotes the canonical extension of r.

In general, the following notation is used for the tangential divergence [70]

$$\operatorname{div}_\Gamma r = \operatorname{div} R|_\Gamma - DRn \cdot n. \tag{3.18}$$

In $\mathbb{R}^d$, the *Tangential Stokes formula* and the *Tangential Green's formula* reads as follows

Theorem 3.13 (Tangential Stokes formula)**.** *For a vector $r \in C^1(\Gamma)^d$, the tangential Stokes formula is given by*

$$\int_\Gamma \operatorname{div}_\Gamma r \; d\Gamma = \int_\Gamma \kappa r \cdot n \; d\Gamma, \tag{3.19}$$

where $\kappa = \operatorname{div}_\Gamma n$ denotes the curvature.

And subsequently, we have

Theorem 3.14 (Tangential Green's formula)**.** *For a function $f \in C^1(\Gamma)$ and a vector $r \in C^1(\Gamma)^d$ and a closed boundary Γ, the tangential Green's formula equals*

$$\int_\Gamma f \operatorname{div}_\Gamma r + \nabla_\Gamma f \cdot r \; d\Gamma = \int_\Gamma \kappa f r \cdot n \; d\Gamma, \tag{3.20}$$

where κ denotes the curvature.

Corresponding proofs for Thm. 3.13 and Thm. 3.14 can be found in [22].

3.5 Adjoint Equations

For the derivation of the shape gradient, the determination of the adjoint system is necessary. The adjoint operator was originally defined by Lagrange [47].

For the introduction of the adjoint operator we follow [5]. The adjoint operator can be interpreted as a generalization of the matrix transpose to infinite-dimensional spaces. Therefore we introduce some properties of the matrix transpose. For a given linear continuous operator $A : \mathbb{R}^n \mapsto \mathbb{R}^m$, we can use the Euclidean inner products for $\mathbb{R}^n$ and $\mathbb{R}^m$

$$y^\top Ax = (Ax, y)_{\mathbb{R}^m}, \quad x^\top A^\top y = (x, A^\top y)_{\mathbb{R}^n} \qquad \forall x \in \mathbb{R}^n, y \in \mathbb{R}^m.$$

This yields the definition for the transpose (*adjoint*) operator

$$(Ax, y)_{\mathbb{R}^m} = (x, A^\top y)_{\mathbb{R}^n} \qquad \forall x \in \mathbb{R}^n, y \in \mathbb{R}^m.$$

Thus, the adjoint operator is equivalent to the transpose operator considering real spaces.

For the general case, we can state the following. We assume the spaces V and W are Hilbert spaces and $L \in \mathcal{L}(V, W)$. Additionally, we assume for the sake of simplicity $\mathbb{K} = \mathbb{R}$ for the corresponding set of scalars. We define an operator $L^* : W \mapsto V$ as the *adjoint* of L as follows.

First, we define a linear functional $\ell_w \in V'$ for a given $w \in W$ via the Riesz representation by

$$\ell_w(v) = (Lv, w)_W \quad \forall v \in V.$$

The boundedness of l_w can be derived by

$$|\ell_w(v)| \leq \|Lv\| \, \|w\| \leq \|L\| \, \|v\| \, \|w\| \, .$$

Using then the Riesz representation theorem (Thm. 2.6), we obtain a uniquely determined element $L^*(w) \in V$ such that

$$\ell_w(v) = (v, L^*(w)) \quad \forall v \in V$$

or equivalently

$$(Lv, w)_W = (v, L^*(w))_V \quad v \in V, w \in W.$$

This is also known as the *Lagrange identity*. The proofs showing the linearity and boundedness of the operator L^* can be found in [5].

Remark 3.15 (Self-adjoint operator)**.** *If for the linear operator L and the corresponding adjoint operator L^* the equalities $V = W$ and $L^* = L$ hold, then L is called a self-adjoint operator. A linear operator $A \in \mathbb{R}^{n\times n}$ is self-adjoint, if one has for A^* that $A^* = A^\top = A$.*

3.6 Smoothing

All of the above definitions and theoretic concepts are needed to derive the shape gradient. For the applicability of gradient-based optimization techniques, an explicit form of the gradient is required. An explicit formulation of the shape gradient on the boundary also leads to a significant reduction of the memory requirement since a huge data reduction can be achieved. Therefore, we express the gradient in the Hadamard sense. Despite having an explicit form of the gradient, one cannot directly conduct the update of the mesh as proposed in Eq. (3.8) due to possible discontinuities in the gradient, e.g., distinct sharp features, arising from the geometry modification. This fact leads to problems in the numerical realization such as undesirable artifacts. Consequently, we have to deal with an ill-conditioned problem. Usually a regularization of the inverse problem is conducted. This can be achieved by a penalization of the circumference of the geometric object. This additional term is described in [70]. The penalization can also be interpreted as a smoothing constraint, since it prevents the optimization algorithm from suffering from objects with sharp features. So, just considering the penalization term of problem 3.2, we can obtain with [37]

$$\begin{aligned} J(H, E, \Omega) &= \int_{\Gamma_{\text{incl}}} 1 \; dS \\ \rightarrow d^2J(H, E, \Omega; V, W) &= \int_{\Gamma_{\text{incl}}} \langle \nabla_\Gamma \langle V, n\rangle, \nabla_\Gamma \langle W, n\rangle\rangle + \kappa \langle V, n\rangle \langle W, n\rangle \; dS. \end{aligned}$$

This second derivative would represent a Newton-type update within the optimization process, but this can also be achieved by a Laplacian gradient smoothing as described in the following. Examples for gradient smoothing approaches are geometric strategies, which adjust mesh points close to the boundaries or filter techniques. The latter one is often applied in topology optimization in order to avoid checkerboards [7, 68].
Here, we focus on implicit gradient smoothing methods. In this research field, Jameson was one of the first who developed and implemented suitable algorithms [39, 40, 41]. These smoothing methods can also be interpreted as preconditioners or Newton approximation. Instead of using the shape gradient directly, a smoothed gradient leads to a better handling of the ill-conditioned problem. The smoothing is usually performed by a change of the inner product from the L_2 space to an appropriate Sobolev space. The use of an alternate inner product indicates why these methods are also called *Sobolev gradient* smoothing methods. They are present in many application areas such as aerodynamic, geometric modeling problems and signal processing. Details of the mentioned applications and further developments can be found in [48, 52].

In this thesis, we focus on *Laplacian* smoothing. Stemming from the Newton-step for circumference regularization, more useful information concerning the direction of the minimum is available. Thanks to this additional information, larger step sizes in the update (Eq. (3.8)) are possible and consequently less optimization iterations are needed. The originally proposed steepest descent method then resembles rather a Newton-type method. Therefore, a better convergence rate can be expected.

In order to determine the smoothed gradient we solve the Laplacian equation

$$-\Delta u + u = u_{old} \qquad \text{in } \Omega \tag{3.21}$$

$$u = u_0 \qquad \text{on } \partial\Omega \tag{3.22}$$

in the smooth space $H_0^1(\Omega)$. The function u_{old} represents the L^2 gradient. Therefore, this way of smoothing is also called implicit smoothing. From Eq. (3.21), we form

the weak formulation $a(u, v) = L(v)$ with

$$a(u, v) = \int_\Gamma (\epsilon_{smooth} \langle \nabla_\Gamma v, \nabla_\Gamma u \rangle + v \cdot u) \, dS$$
$$L(v) = \int_\Gamma u_{old} \cdot v \, dS,$$

where v denotes the test function, u the trial function and ϵ_{smooth} the smoothing parameter, which is related to δ in problem (3.2). A well-chosen ϵ_{smooth} is very important since the update of the boundary and consequently a successful optimization directly depends on the smoothed gradient. Solving this problem stated in the weak formulation in the finite dimensional space $V_h \subset V$, we receive the smoothed gradient. The smoothed gradient replaces the previously calculated gradient $g(x)$ in the update scheme of Eq. (3.8), whereas the rest of the optimization process remains unchanged.

3.7 General Formulation for the Shape Gradient for Linear State Constraints

In this section, a general formulation for the shape gradient for linear state constraints is derived. This result serves as basis for the derivation of the shape gradient in Sec. 4.5.

First of all, we start with the shape derivative in the Hadamard form of a domain integral and a boundary integral as introduced by Zolésio in [70].

Shape Derivative for Domain Integral A general target function $J(\Omega)$, defined on the domain Ω and depending on the state u, is given by

$$J(\Omega) = \int_\Omega h(\Omega; u) \, dx,$$

then the Eulerian derivative is defined as

$$dJ(\Omega; V) = \int_\Omega \dot{h}(\Omega; u)\ dx + \int_\Omega \operatorname{div}(h(\Omega; u)V)\ dx.$$

Assuming Γ is of class $C^k, k \geq 1$, the Eulerian derivative can be formulated as

$$dJ(\Omega; V) = \int_\Omega h'(\Omega; u)\ dx + \int_\Gamma h(\Omega; u)\langle V, n\rangle\ dS. \tag{3.23}$$

Corresponding proofs can be found in [70].

Shape Derivative for Boundary Integral Now we consider a general target functional $J(\Gamma)$, which is only defined on a boundary Γ and depends on the state u, i.e.,

$$J(\Gamma) = \int_\Gamma h(\Gamma; u)\ dS.$$

Applying the Hadamard theorem, the Eulerian derivative can be expressed as

$$dJ(\Gamma; V) = \int_\Gamma \dot{h}(\Gamma; u)\ dS + \int_\Gamma h(\Gamma; u) \operatorname{div}_\Gamma(V)\ dS. \tag{3.24}$$

Using Eq. (3.13), which describes the relation between the material derivative and the shape derivative, yields

$$dJ(\Gamma; V) = \int_\Gamma h'(\Gamma; u)\ dS + \int_\Gamma \left[\nabla_\Gamma h(\Gamma; u) \cdot V + h(\Gamma; u) \operatorname{div}_\Gamma(V)\right]\ dS. \tag{3.25}$$

Exploiting the tangential Green's formula (Thm. 3.14), one gets

$$dJ(\Gamma; V) = \int_\Gamma h'(\Gamma; u) + \kappa h(\Gamma; u)\langle V, n\rangle\ dS, \tag{3.26}$$

where κ denotes the curvature. Additionally, the equation

$$h'(\Gamma; u) = h'(\Omega; u)|_\Gamma + \frac{\partial h}{\partial n}\langle V, n\rangle$$

holds.

Finally, the shape derivative of the boundary integral can be expressed as

$$dJ(\Gamma; V) = \int_\Gamma h'(\Omega; u)|_\Gamma + \langle V, n\rangle \left[\frac{\partial h}{\partial n} + \kappa h\right] \, dS. \tag{3.27}$$

The expressions for the shape derivative for the domain integral (3.23) and for the boundary integral (3.27) will be reused in the remainder of the section to derive the shape derivative for the general problem (3.2). In the next chapter, this result will be transferred to the Maxwell problem (4.1).
Furthermore, we refer to results obtained in [71]. In particular, they consider the variational formulation of the shape derivatives for the compressible Navier-Stokes equations. For obtaining a gradient representation in Hadamard form, results of [60] are recalled and extended.
In order to obtain an actual expression for the shape derivative for linear state constraints, we revisit the generalized problem (3.2), which is formulated in divergence form and derive the weak formulation of the (domain) constraint.

Then, the strong variational formulation of the (domain) constraint is given by

$$\begin{aligned} 0 &= \int_\Omega \langle \xi, \frac{\partial \phi(x)}{\partial t}\rangle + \langle \xi, \operatorname{div} F(\phi(x))\rangle \, dx \\ &= \int_\Omega \langle \xi, \frac{\partial \phi(x)}{\partial t}\rangle + \operatorname{div} F(\xi\phi(x)) - \langle \nabla, \xi \cdot F(\phi(x))\rangle \, dx \qquad \forall \xi \in H^1(\Omega). \end{aligned}$$

Using the divergence theorem (Thm. 2.1) yields

$$0 = \int_\Omega \langle \xi, \frac{\partial \phi(x)}{\partial t}\rangle - \langle F(\phi(x)), \nabla \xi\rangle \, dx \qquad \forall \xi \in H^1(\Omega).$$

Consequently, we get the following expression for the weak formulation

$$0 = \int_{\Omega} \langle \xi, \frac{\partial \phi(x)}{\partial t} \rangle - \langle F(\phi(x)), \nabla \xi \rangle \; dx + \int_{\partial\Omega} \xi \cdot F_\Gamma(\phi(x), n) \; dS.$$

The expression $F_\Gamma(\phi(x), n)$ indicates that the boundary conditions might also depend on the outer pointing normal n as it is the case for the Maxwell's equations.

First, we concentrate on the derivation of the local shape derivative of the (domain) constraint. Therefore, we assume $\Omega_\varepsilon \subset D \; \forall \varepsilon$. Then, we can formulate the last equation as

$$0 = \int_{\Omega_\epsilon} \langle \xi, \frac{\partial \phi(x_\epsilon)}{\partial t} \rangle - \langle F(\phi(x_\epsilon)), \nabla \xi \rangle \; dx_\epsilon \qquad \forall \xi \in H^1(D).$$

Next, we can determine the local shape derivative of the (domain) constraint

$$\begin{aligned} 0 &= \frac{d}{d\epsilon} \int_{\Omega_\epsilon} \langle \xi, \frac{\partial \phi(x_\epsilon)}{\partial t} \rangle - \langle F(\phi(x_\epsilon)), \nabla \xi \rangle \; dx_\epsilon \big|_{\epsilon=0} \\ &= \int_{\Omega} \langle \xi, \frac{\partial \phi'(x)}{\partial t} \rangle - \langle F'(\phi(x))\phi'[V], \nabla \xi \rangle \; dx - \int_{\partial\Omega} \langle V, n \rangle \langle F(\phi(x)), \nabla \xi \rangle \; dS. \end{aligned}$$

Now, we continue with the derivation of the local shape derivative for the boundary constraints F_Γ. Here, we have to take the additional dependence on the normal n into account. Hence, analogously to Eq. (3.24), also cf. [71], we can determine the Eulerian derivative

$$\begin{aligned} 0 = \int_{\partial\Omega} \langle V, n \rangle [&\langle \nabla(\xi \cdot F_\Gamma(\phi(x), n)), n \rangle \\ &+ \kappa \left(\xi \cdot F_\Gamma(\phi(x), n) - D_n(\xi \cdot F_\Gamma(\phi(x), n)) \cdot n \right) \\ &+ \operatorname{div}_\Gamma(D_n^\top(\xi \cdot F_\Gamma(\phi(x), n)))] \; dS. \end{aligned} \tag{3.28}$$

If the boundary conditions F_Γ are linear, the curvature κ vanishes since $D_n(\xi \cdot F_\Gamma(\phi, n)) \cdot n = \xi \cdot F_\Gamma(\phi, n)$.

Using the definition of the tangential divergence (Def. 3.12), one gets

$$\begin{aligned} dJ(\Omega;V) &= \int_{\partial\Omega} \langle V,n\rangle[\langle\nabla(\xi\cdot F_\Gamma(\phi(x),n)),n\rangle \\ &\quad + \operatorname{div}(D_n^\top(\xi\cdot F_\Gamma(\phi(x),n))) - \langle\nabla(D_n^\top(\xi\cdot F_\Gamma(\phi(x),n))n),n\rangle \; dS \\ &= \int_{\partial\Omega} \langle V,n\rangle \; \operatorname{div}(D_n^\top(\xi\cdot F_\Gamma(\phi(x),n))) \; dS. \end{aligned} \tag{3.29}$$

This important result is recalled in Sec. 4.5 to derive the shape derivative for the Maxwell problem (4.1).

4

Shape Optimization for Electromagnetic Simulations

In this chapter, a detailed analysis of a 3D time-dependent Maxwell problem is conducted. The focus lies on the general Maxwell problem subject to boundary conditions motivated by real-life applications. Therefore, an exact formulation of the underlying problem has to be defined. This comprises the derivation of an appropriate upwind flux for possible jumps in the material coefficients and general stability as well as the definition of suitable boundary conditions. Defining a problem-dependent target functional completes the formulation of the optimization problem. Adjoint equations, which take part in the derivation of the Maxwell shape gradient, are introduced as well. This thorough analysis contains new approaches and ideas and is of high interest especially concerning real-life applications.

4.1 Derivation of the Steger-Warming Upwind Flux for Maxwell's Equations

Let us recall the motivation of Galerkin methods from Subsec. 2.2.2: In general, one needs an appropriate strategy to relate local solutions at the interface to one global solution. In our application, we have discontinuous jumps in the material coefficients since we consider a domain with different electromagnetic properties such as electric permittivity ε, magnetic permeability μ or conductivity σ.
In the following, we derive an appropriate flux contained in the class of upwind fluxes to ensure stability and to preserve monotonicity of the solution. A more detailed discussion of fluxes can be found in Subsec. 2.2.2.

Using local solution methods such as Galerkin methods, accuracy in the solutions is ensured. To guarantee stability, a carefully chosen upwind flux is indispensable. Here, we derive the Steger–Warming upwind flux for the 3D Maxwell's equations. Later, the derived expression enters the problem formulation. The Steger–Warming upwind scheme is based on a flux–projection onto the space of eigenvectors [67, 72]. Therefore, the flux vectors are linearized with respect to the primitive variables H and E resulting into matrices with real eigenvalues and a complete set of eigenvectors. With an appropriate similarity transformation, we can split the system matrix into two parts: one describes the behavior of the characteristic variables of the left-hand side of the interface and the other one the behavior of the right-hand side.

First of all, we start with the variational formulation for the Maxwell's equations for our local approach, cf., Chapter 2

$$\int_0^T \int_\Omega v \cdot (\frac{\partial H}{\partial t} + \frac{1}{\mu} \operatorname{curl} E) + w \cdot (\frac{\partial E}{\partial t} - \frac{1}{\varepsilon} \operatorname{curl} H + \frac{\sigma}{\varepsilon} E)\ dx\ dt = 0,$$

where v and w are the test functions.

Integration by parts for the curl operator leads to

$$0 = \int_0^T \int_\Omega \langle v, \frac{\partial H}{\partial t} \rangle + \langle E, \frac{1}{\mu} \operatorname{curl} v \rangle + \langle w, \frac{\partial E}{\partial t} \rangle - \langle H, \frac{1}{\varepsilon} \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E \rangle \, dx \, dt$$
$$- \int_0^T \int_{\partial\Omega \cup \tilde{S}} \frac{1}{\mu} \langle v, E \times n \rangle - \frac{1}{\varepsilon} \langle w, H \times n \rangle \, dS \, dt.$$

Here, $\tilde{S}$ collects inner interfaces, where no continuity for test- and ansatzfunctions is required due to possible jumps at the interface.

To obtain the general form as stated in Sec. 3.7, Eq. (3.29) can also be expressed as

$$\int_{t_0}^{t_f} \int_\Omega \langle v, \frac{\partial H}{\partial t} \rangle + \langle E, \frac{1}{\mu} \operatorname{curl} v \rangle + \langle w, \frac{\partial E}{\partial t} \rangle - \langle H, \frac{1}{\varepsilon} \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E \rangle \, dx \, dt$$
$$- \int_{t_0}^{t_f} \int_{\Gamma \cup \tilde{S}} h(s, n) \, dS \, dt = 0$$

with

$$h(s,n) = (v, w) \begin{bmatrix} 0 & \frac{1}{\mu} A_n \\ -\frac{1}{\varepsilon} A_n & 0 \end{bmatrix} \begin{pmatrix} H \\ E \end{pmatrix}, \quad A_n = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}.$$

The definition of A_n is closely related to the flux definition of the Maxwell's equations (3.5) through $A_n = R_1 n_x + R_2 n_y + R_3 n_z$. For the projection onto the space of eigenvectors, the computation of the eigenvalues and eigenvectors of

$$\hat{A} := \begin{bmatrix} 0 & \frac{1}{\mu} A_n \\ -\frac{1}{\varepsilon} A_n & 0 \end{bmatrix}$$

is needed. For the system considered here, the eigenvalues are

$$\lambda_1 = 0, \ \lambda_2 = 0, \ \lambda_3 = -\frac{1}{\sqrt{\mu\varepsilon}}, \ \lambda_4 = -\frac{1}{\sqrt{\mu\varepsilon}}, \ \lambda_5 = \frac{1}{\sqrt{\mu\varepsilon}}, \ \lambda_6 = \frac{1}{\sqrt{\mu\varepsilon}}.$$

Corresponding eigenvectors can then be computed as

$$
\begin{aligned}
v_1 &= \left(\; 0,0,0,n_x,n_y,n_z \; \right)^\top, \qquad v_2 = \left(\; n_x,n_y,n_z,0,0,0 \; \right)^\top, \\
v_3 &= \left(\; -\sqrt{\tfrac{\varepsilon}{\mu}}n_x n_y, \sqrt{\tfrac{\varepsilon}{\mu}}(n_x^2+n_z^2), -\sqrt{\tfrac{\varepsilon}{\mu}}n_y n_z, -n_z, 0, n_x \; \right)^\top, \\
v_4 &= \left(\; \sqrt{\tfrac{\varepsilon}{\mu}}n_x n_z, \sqrt{\tfrac{\varepsilon}{\mu}}n_y n_z, -\sqrt{\tfrac{\varepsilon}{\mu}}(n_x^2+n_y^2), -n_y, n_x, 0 \; \right)^\top, \\
v_5 &= \left(\; \sqrt{\tfrac{\varepsilon}{\mu}}n_x n_y, -\sqrt{\tfrac{\varepsilon}{\mu}}(n_x^2+n_z^2), \sqrt{\tfrac{\varepsilon}{\mu}}n_y n_z, -n_z, 0, n_x \; \right)^\top, \\
v_6 &= \left(\; -\sqrt{\tfrac{\varepsilon}{\mu}}n_x n_z, -\sqrt{\tfrac{\varepsilon}{\mu}}n_y n_z, \sqrt{\tfrac{\varepsilon}{\mu}}(n_x^2+n_y^2), -n_y, n_x, 0 \; \right)^\top.
\end{aligned}
$$

In the following, we use the abbreviations

$$
Y = \sqrt{\frac{\varepsilon}{\mu}}, \quad Z = \sqrt{\frac{\mu}{\varepsilon}}, \quad c = \frac{1}{\sqrt{\mu\varepsilon}}.
$$

To compute the Jacobian $\hat{A}$, we exploit the similarity transformation

$$
\Lambda = P^{-1}\hat{A}P,
$$

where

$$
\begin{aligned}
P &:= [v_5, v_3, v_1, v_6, v_4, v_2] \\
&= \begin{bmatrix}
n_x n_y Y & -n_x n_y Y & 0 & -n_x n_z Y & n_x n_z Y & n_x \\
-(n_x^2+n_z^2)Y & (n_x^2+n_z^2)Y & 0 & -n_y n_z Y & n_y n_z Y & n_y \\
n_y n_z Y & -n_y n_z Y & 0 & (n_x^2+n_y^2)Y & -(n_x^2+n_y^2)Y & n_z \\
-n_z & -n_z & n_x & -n_y & -n_y & 0 \\
0 & 0 & n_y & n_x & n_x & 0 \\
n_x & n_x & n_z & 0 & 0 & 0
\end{bmatrix}
\end{aligned}
$$

and the diagonal matrix $\Lambda = \mathrm{diag}(c, -c, 0, c, -c, 0)$, i.e., the diagonal elements of Λ are given by the corresponding eigenvalues. Defining $A^{\pm} = P\Lambda^{\pm}P^{-1}$, where $\Lambda^{+} = \mathrm{diag}(\max(\lambda_i, 0))$ and $\Lambda^{-} = \mathrm{diag}(\min(\lambda_i, 0))$, it is easy to see that $\hat{A}$ can be

split into a sum of

$$A^+ = \frac{1}{2}\begin{bmatrix} n_y^2+n_z^2 & -n_xn_y & -n_xn_z & 0 & -n_zY & n_yY \\ -n_xn_y & n_x^2+n_z^2 & -n_yn_z & n_zY & 0 & -n_xY \\ -n_xn_z & -n_yn_z & n_x^2+n_y^2 & -n_yY & n_xY & 0 \\ 0 & n_zZ & -n_yZ & n_y^2+n_z^2 & -n_xn_y & -n_xn_z \\ -n_zZ & 0 & n_xZ & -n_xn_y & n_x^2+n_z^2 & -n_yn_z \\ n_yZ & -n_xZ & 0 & -n_xn_z & -n_yn_z & n_x^2+n_y^2 \end{bmatrix}$$

and

$$A^- = \frac{1}{2}\begin{bmatrix} -(n_y^2+n_z^2) & n_xn_y & n_xn_z & 0 & -n_zY & n_yY \\ n_xn_y & -(n_x^2+n_z^2) & n_yn_z & n_zY & 0 & -n_xY \\ n_xn_z & n_yn_z & -(n_x^2+n_y^2) & -n_yY & n_xY & 0 \\ 0 & n_zZ & -n_yZ & -(n_y^2+n_z^2) & n_xn_y & n_xn_z \\ -n_zZ & 0 & n_xZ & n_xn_y & -(n_x^2+n_z^2) & n_yn_z \\ n_yZ & -n_xZ & 0 & n_xn_z & n_yn_z & -(n_x^2+n_y^2) \end{bmatrix}.$$

By setting

$$\tilde{A}_n = \begin{bmatrix} n_y^2+n_z^2 & -n_xn_y & -n_xn_z \\ -n_xn_y & n_x^2+n_z^2 & -n_yn_z \\ -n_xn_z & -n_yn_z & n_x^2+n_y^2 \end{bmatrix},$$

one can compute $A_nE = n \times E = -E \times n$ and $\tilde{A}_nE = n \times (E \times n) = -A_n^2E$. The decomposed parts of $\hat{A}$ can then be expressed as

$$A^+ = \frac{1}{2}\begin{bmatrix} \tilde{A}_nc & A_nYc \\ -A_nZc & \tilde{A}_nc \end{bmatrix}, \quad A^- = \frac{1}{2}\begin{bmatrix} -\tilde{A}_nc & A_nYc \\ -A_nZc & -\tilde{A}_nc \end{bmatrix}. \tag{4.1}$$

Summarizing the above considerations, the upwind flux is given by

$$(v,w)\cdot\hat{A}\cdot\begin{pmatrix}H\\E\end{pmatrix}=(v,w)\cdot\begin{bmatrix}0 & \frac{1}{\mu}A_n\\ -\frac{1}{\varepsilon}A_n & 0\end{bmatrix}\begin{pmatrix}H\\E\end{pmatrix} \tag{4.2}$$

$$\begin{aligned}&=(v,w)(P\Lambda P^{-1})\cdot\begin{pmatrix}H\\E\end{pmatrix}\\ &=(v,w)\left[(P\Lambda^{+}P^{-1})\cdot\begin{pmatrix}H^{-}\\E^{-}\end{pmatrix}+(P\Lambda^{-}P^{-1})\cdot\begin{pmatrix}H^{+}\\E^{+}\end{pmatrix}\right]\\ &=(v,w)\frac{1}{2}\begin{bmatrix}\tilde{A}_nc & A_nYc\\ -A_nZc & \tilde{A}_nc\end{bmatrix}\cdot\begin{pmatrix}H^{-}\\E^{-}\end{pmatrix}\\ &\quad+(v,w)\frac{1}{2}\begin{bmatrix}-\tilde{A}_nc & A_nYc\\ -A_nZc & -\tilde{A}_nc\end{bmatrix}\cdot\begin{pmatrix}H^{+}\\E^{+}\end{pmatrix}:=h(s,n)\end{aligned} \tag{4.3}$$

In the derived formulation, the primitive variables H and E are split into H^-, H^+ resp. E^-, E^+ for a proper representation of the non continuous values at inner interfaces $\tilde{S}$. The matrix block in front of $(H^+, E^+)^\top$ denotes the incoming information of the domain and consequently, the matrix block in front of $(H^-, E^-)^\top$ denotes the outgoing information of the domain and this matrix block is positive semidefinite.

This general expression $h(s,n)$ for the upwind scheme has to be adjusted to the specific boundary conditions that hold on the different surfaces. Appropriate formulations are derived in Sec. 4.2.

4.2 Boundary Formulation

Two different kinds of boundary conditions govern the surfaces of the bounding box of the computational domain. One boundary condition defines the pulse p entering the domain. This surface is denoted by $\Gamma_{in/out}$. All other surfaces $\Gamma_{non-reflective}$ of the bounding box should have the property to be non-reflective. The surface of the obstacle is supposed to be reflective and is denoted by $\Gamma_{reflective} = \Gamma_{incl}$. On $\Gamma_{in/out}$, the values of the reflected E and H field are recorded and enter the evaluation of the target function as explained in detail in Sec. 4.3. Boundary conditions for the remaining surfaces are specified in the remainder of this section.

First, we consider the reflection boundary of the hidden obstacle. If the material of the obstacle has zero electric permeability, i.e., a perfect conductor, then we want the electric field on the boundary and inside the obstacle to vanish for the observation time interval. An example for this is a metallic object, where the resulting currents automatically eliminate the electric field. The system matrix belonging to the local magnetic field H^- and electric field E^- remains always the same, while the system matrix belonging to the responding remote fields H^+, E^+ varies due to the desired behavior.

Reflective Boundary Condition For the reflective boundary, we determine

$$\begin{aligned} h_{reflective}(H,E,n) &= \frac{1}{2}(v,w)\begin{bmatrix} \tilde{A}_n c & A_n Y c \\ -A_n Z c & \tilde{A}_n c \end{bmatrix} \cdot \begin{pmatrix} H^- \\ E^- \end{pmatrix} \\ &\quad + \frac{1}{2}(v,w)\begin{bmatrix} -\tilde{A}_n c & -A_n Y c \\ -A_n Z c & -\tilde{A}_n c \end{bmatrix} \cdot \begin{pmatrix} H^+ \\ E^+ \end{pmatrix} \\ &= (v,w)\begin{bmatrix} 0 & 0 \\ -A_n Z c & 0 \end{bmatrix} \cdot \begin{pmatrix} H \\ E \end{pmatrix} = w \cdot (-A_n) Z c H. \end{aligned}$$

Inlet Boundary Condition For the inlet boundary $\Gamma_{in/out}$, one can define the following boundary conditions

$$h_{in/out}(H,E,n) = \frac{1}{2}(v,w)\begin{bmatrix} \tilde{A}_n c & A_n Y c \\ -A_n Z c & \tilde{A}_n c \end{bmatrix} \cdot \begin{pmatrix} H^- \\ E^- \end{pmatrix} + \frac{1}{2}(v,w)\begin{pmatrix} p_H \\ p_E \end{pmatrix}. \tag{4.4}$$

Here, the remote fields H^+ and E^+ are replaced with the incoming pulse p_H and p_E, which varies depending on the considered application. For the incoming flow at the inlet boundary, one can show redundancy of

$$\frac{1}{2}\begin{bmatrix} -\tilde{A}_n c & A_n Y c \\ -A_n Z c & -\tilde{A}_n c \end{bmatrix} \cdot \begin{pmatrix} H \\ E \end{pmatrix} = \begin{pmatrix} p_H \\ p_E \end{pmatrix}.$$

It follows that

$$\frac{1}{2}(n \times (H \times n)c + (n \times E)Yc) = p_H, \tag{4.5}$$

$$\frac{1}{2}((H \times n)Zc + (n \times (E \times n)c)) = p_E. \tag{4.6}$$

Equation (4.6) can be transformed to

$$(H \times n)Zc + (n \times (E \times n)c) = 2p_E \iff H \times n = \frac{2}{c}Yp_E - (n \times (E \times n)Y).$$

Inserting the expression for $H \times n$ in Eq. (4.5) yields

$$\begin{aligned}
&\frac{1}{2}\left(n \times \left[\frac{2}{c}Yp_E - n \times (E \times n)Y\right]c + (n \times E)Yc\right) = p_H \\
\iff &n \times p_E Y - \frac{1}{2}n \times [n \times (E \times n)Y]\,c + \frac{1}{2}(n \times E)Yc = p_H \\
\iff &Yn \times p_E = p_H \\
\iff &n \times p_E = p_H Z.
\end{aligned}$$

All together we have

$$\frac{1}{2}\left[(H \times n)Zc + (n \times (E \times n)c)\right] = p_E \iff \frac{1}{2}\left[-A_n H + \tilde{A}_n EY\right] = \frac{Y}{c}p_E \qquad \text{and} \qquad n \times p_E = p_H Z. \tag{4.7}$$

Due to the shown redundancy, one can choose one of the two lines of the matrix block of Eq. (4.4) to represent the boundary condition on $\Gamma_{in/out}$.

Non-reflecting Boundary Condition For all other surfaces of the bounding box, we define a non-reflecting behaviour, which means that only the leaving information is considered and no information is sent back into the domain. Consequently, the boundary condition is formulated as

$$h_{non-reflective}(H, E, n) = \frac{1}{2}(v, w)\begin{bmatrix} \tilde{A}_n c & A_n Yc \\ -A_n Zc & \tilde{A}_n c \end{bmatrix} \cdot \begin{pmatrix} H^- \\ E^- \end{pmatrix}.$$

Thus, we arrive at a detailed expression for the general Maxwell problem in a weak sense constrained by problem-dependent boundary conditions in a general form as explained in Sec. 3.7

$$\begin{aligned}
&\int_{t_0}^{t_f}\int_{\Omega}\langle v, \frac{\partial H}{\partial t}\rangle + \langle E, \frac{1}{\mu}\operatorname{curl} v\rangle + \langle w, \frac{\partial E}{\partial t}\rangle - \langle H, \frac{1}{\varepsilon}\operatorname{curl} w\rangle + \frac{\sigma}{\varepsilon}\langle w, E\rangle \; dx \; dt \\
&- \int_{t_0}^{t_f}\int_{\Gamma_{reflective}} h_{reflective}(H, E, n) \; dS \; dt - \int_{t_0}^{t_f}\int_{\Gamma_{in/out}} h_{in/out}(H, E, n) \; dS \; dt \\
&- \int_{t_0}^{t_f}\int_{\Gamma_{non-reflective}} h_{non-reflective}(H, E, n) \; dS \; dt. \qquad (4.8)
\end{aligned}$$

With this expression, we have a complete description of the boundary conditions that might occur in our application. In the following section a proper expression for the target functional is formulated and hence, an entire formulation of the underlying problem is determined.

4.3 Problem Formulation

Our goal is to determine the actual structure of a hidden object inside a given domain. Based on an initial geometry, the forward simulation is performed and the reflected electric and magnetic fields on the surface $\Gamma_{in/out}$ are simulated. To obtain an analytical expression for the simulated field values, we only consider the part that is leaving the computational domain, namely the reflected values at the inlet boundary $\Gamma_{in/out}$, which is the first term of Eq. (4.4). Analogously, redundancy can also be obtained in this case with $p_H = p_E = 0$.

This redundancy can also be exploited for the boundary $\Gamma_{non-reflective}$.
With the expression of the simulated fields (4.7) and $\Gamma_{incl} = \Gamma_{reflective}$, we can state a proper formulation of our optimization problem, namely

Problem 4.1 (Maxwell problem)**.**

$$\min_{\Gamma_{incl}} \; J(H,E,\Omega) = \frac{1}{2}\int_{t_0}^{t_f}\int_{\Gamma_{in/out}} \left\|-\alpha A_n H Z c + \tilde{A}_n E c - F_{meas}\right\|^2 \, dS \, dt + \delta \int_{\Gamma_{incl}} 1 \, dS \tag{4.9}$$

$$\begin{aligned}
\mu\frac{\partial H}{\partial t} &= -\operatorname{curl} E && \text{in } \Omega \\
\varepsilon\frac{\partial E}{\partial t} &= \operatorname{curl} H - \sigma E && \text{in } \Omega \\
\text{such that} \quad \frac{1}{2}\left(-A_n H Z c + \tilde{A}_n E c\right) &= p_E && \text{on } \Gamma_{in/out} \\
\frac{1}{2}\left(-A_n H Z c + \tilde{A}_n E c\right) &= 0 && \text{on } \Gamma_{non\text{-}reflective} \\
-A_n H Z c &= 0 && \text{on } \Gamma_{incl},
\end{aligned} \tag{4.10}$$

where F_{meas} denotes the actual measurement values. Depending on the factor $\alpha \in \{0,1\}$ either both reflected fields E and H are taken into account or just the reflected E field. The regularization parameter δ scales the perimeter penalization.

Using *curl-* instead of matrix notation, one obtains

$$\min_{\Gamma_{incl}} \; J(H,E,\Omega) = \frac{1}{2}\int_{t_0}^{t_f}\int_{\Gamma_{in/out}} \|-\alpha(n\times H) Z c + n\times(E\times n)c - F_{meas}\|^2 \, dS \, dt + \delta \int_{\Gamma_{incl}} 1 \, dS$$

$$\begin{aligned}
\mu\frac{\partial H}{\partial t} &= -\operatorname{curl} E && \text{in } \Omega \\
\varepsilon\frac{\partial E}{\partial t} &= \operatorname{curl} H - \sigma E && \text{in } \Omega \\
\text{such that} \quad \frac{1}{2}\left((H\times n) Z c + n\times(E\times n)c\right) &= p_E && \text{on } \Gamma_{in/out} \\
\frac{1}{2}\left((H\times n) Z c + n\times(E\times n)c\right) &= 0 && \text{on } \Gamma_{non-reflective} \\
(H\times n) Z c &= 0 && \text{on } \Gamma_{incl}.
\end{aligned}$$

Existence and Uniqueness of Local Shape Derivatives of Maxwell's Equations First analytical results concerning the existence and uniqueness of the local shape derivatives of the Maxwell's equations can be found in [13]. In that paper the authors formulate the required derivatives in the Lebesgue space. They make use of the concept of material derivatives. All their considerations and proofs correspond to $H(\mathrm{curl}, \Omega)$. Their main result is the proof of the proposition that the solution to the Maxwell's equations is weakly material differentiable in $H(\mathrm{curl}, \Omega)$ as well as weakly shape differentiable in $H(\mathrm{curl}, \Omega)$. Furthermore, the material derivative resp. the shape derivative are supposed to be the solution of corresponding derived systems. These propositions are proved by showing the regularity of the solution using shape difference quotients.

A further remark on the existence of the local shape derivatives can be extracted from a paper by Ito, Kunisch and Peichl [38]. One result of that paper is that the shape derivative of the objective function does not necessarily require existence of the variation of the PDE state. The idea is to formulate a variational problem whose solution may replace the variation of the PDE state.

4.4 Formulation of the Corresponding Adjoint Equations

Although it is not obvious in the general expression of the shape gradient for linear constraints formulated in Eq. (3.29), the adjoint system takes implicitly part in the final expression of the shape gradient as presented at the end of this chapter. Considering Eq. (4.2), one can obtain the adjoint system by exploiting $A_n^\top = -A_n$ and rescaling of Y and Z. For completeness, we derive the adjoint system via the Lagrange ansatz.

Exploiting the Optimality Condition We follow the approach presented in [31] to derive the adjoint system. For this purpose, we consider the Lagrange function defined by

$$\mathcal{L}(H, E, \Omega, \lambda_H(t), \lambda_E(t)) := J(H, E, \Omega) + \begin{pmatrix} \lambda_H(t) \\ \lambda_E(t) \end{pmatrix} c(H, E) \tag{4.11}$$

where $c(H,E)$ describes the state equations given by the problem constraints in Eq. (4.10). We only focus on the parts depending on the states H and E. Therefore, the boundary conditions for the *non-reflective* part and the *in/out* part are identical. The functions $\lambda_H(t)$ and $\lambda_E(t)$ denote the Lagrange-multiplier for the H- and E-field, respectively. Taking the different boundary conditions into account, one obtains

$$\begin{aligned}
&\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix} c(H,E) = \\
&\int_{t_0}^{t_f}\int_{\Omega} \lambda_H(t)\left[\frac{\partial H}{\partial t} + \frac{1}{\mu}\operatorname{curl} E\right]\, dx\, dt + \int_{t_0}^{t_f}\int_{\Omega} \lambda_E(t)\left[\frac{\partial E}{\partial t} - \frac{1}{\varepsilon}\operatorname{curl} H + \frac{\sigma}{\varepsilon}E\right]\, dx\, dt \\
&+ \int_{t_0}^{t_f}\int_{\Gamma_{non-reflective}\cup\Gamma_{in/out}} (H,E)\begin{bmatrix}\tilde{A}_n c & A_n Z c\\ -A_n Y c & \tilde{A}_n c\end{bmatrix}\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}\, dS\, dt \\
&+ \int_{t_0}^{t_f}\int_{\Gamma_{incl}} (H,E)\begin{bmatrix}0 & -A_n Z c\\ 0 & 0\end{bmatrix}\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}\, dS\, dt \begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}\, dS\, dt.
\end{aligned}$$

At an optimal point, the optimality condition

$$\mathcal{L}_{(H,E)}(H,E,\Omega,\lambda_H(t),\lambda_E(t)) = 0$$

must hold. This condition is exploited to determine the adjoint system.
We omit the integration over time and the regularization term for brevity and therefore use

$$\tilde{J}(H,E,\Omega) = \frac{1}{2}\int_{\Gamma_{\text{in/out}}} \left\| -\alpha A_n H + \tilde{A}_n E Y - F_{meas}\right\|^2\, dS.$$

in (4.11). First, we derive the adjoint equation for the target functional. Therefore,

we transform the functional in a more general form

$$\begin{aligned}
&\frac{1}{2}(-H^\top A_n^\top + E^\top \tilde{A}_n^\top Y - F_{meas}^\top)(-A_n H + \tilde{A}_n E Y - F_{meas}) \\
&= \frac{1}{2}(H^\top A_n^\top A_n H - H^\top A_n^\top \tilde{A}_n E Y + H^\top A_n^\top F_{meas} - E^\top \tilde{A}_n^\top Y A_n H + E^\top \tilde{A}_n^\top Y \tilde{A}_n E \\
&- E^\top \tilde{A}_n^\top Y F_{meas} + F_{meas}^\top A_n H - F_{meas}^\top A_n E Y + F_{meas}^\top F_{meas})
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2}\left((H^\top, E^\top)\begin{bmatrix} A_n^\top A_n & -A_n^\top \tilde{A}_n Y \\ -\tilde{A}_n^\top A_n Y & Y\tilde{A}_n^\top \tilde{A}_n Y \end{bmatrix}\begin{pmatrix} H \\ E \end{pmatrix}\right) \\
&\quad + \frac{1}{2}\left((2 \cdot F_{meas}^\top A_n, -2 \cdot F_{meas}\tilde{A}_n Y)\begin{pmatrix} H \\ E \end{pmatrix} + F_{meas}^\top F_{meas}\right)
\end{aligned}$$

In the following, we use the abbreviations

$$Q_2 = \begin{bmatrix} A_n^\top A_n & -A_n^\top \tilde{A}_n Y \\ -\tilde{A}_n^\top A_n Y & Y\tilde{A}_n^\top \tilde{A}_n Y \end{bmatrix}, \quad Q_1 = (2 \cdot F_{meas}^\top A_n, -2 \cdot F_{meas}\tilde{A}_n Y)$$

leading to

$$= \frac{1}{2}\left[(H^\top, E^\top) \cdot Q_2 \begin{pmatrix} H \\ E \end{pmatrix} + Q_1 \begin{pmatrix} H \\ E \end{pmatrix} + F_{meas}^\top F_{meas}\right]. \tag{4.12}$$

Next, we differentiate with respect to the state variables H and E.

$$\begin{aligned}
&\frac{d}{d\varepsilon}\frac{1}{2}\left((H_\varepsilon^\top, E_\varepsilon^\top) \cdot Q_2 \begin{pmatrix} H_\varepsilon \\ E_\varepsilon \end{pmatrix} + Q_1 \begin{pmatrix} H_\varepsilon \\ E_\varepsilon \end{pmatrix} + F_{meas}^\top F_{meas}\right)\Big|_{\varepsilon=0} \\
&= \frac{1}{2}\left((H'^\top, E'^\top) Q_2 \begin{pmatrix} H \\ E \end{pmatrix} + (H^\top, E^\top) Q_2 \begin{pmatrix} H' \\ E' \end{pmatrix} + Q_1 \begin{pmatrix} H' \\ E' \end{pmatrix}\right).
\end{aligned}$$

Setting $Q_2 = B^\top B$, one arrives at

$$
\begin{aligned}
&= \frac{1}{2}\left[\left(B\begin{pmatrix} H' \\ E' \end{pmatrix}\right)^\top B\begin{pmatrix} H \\ E \end{pmatrix} + \left(B\begin{pmatrix} H \\ E \end{pmatrix}\right)^\top B\begin{pmatrix} H' \\ E' \end{pmatrix} + Q_1\begin{pmatrix} H' \\ E' \end{pmatrix}\right] \\
&= \frac{1}{2}\left[2\cdot(H^\top, E^\top)B^\top B\begin{pmatrix} H' \\ E' \end{pmatrix} + Q_1\begin{pmatrix} H' \\ E' \end{pmatrix}\right] \\
&= (H^\top, E^\top)Q_2\begin{pmatrix} H' \\ E' \end{pmatrix} + \frac{1}{2}Q_1\begin{pmatrix} H' \\ E' \end{pmatrix}
\end{aligned}
$$

This expression determines the right-hand side of the adjoint boundary condition on $\Gamma_{in/out}$.

In the following, the adjoint equations for the partial differential system are derived. For brevity reasons, we also omit the integration over time and differentiate the corresponding defined $\tilde{c}(H, E)$ with respect to the states H and E, and set $\Gamma_{non-reflective} \cup \Gamma_{in/out} = \Gamma_i \backslash \Gamma_{incl}$, yielding

$$
\begin{aligned}
&-\tilde{c}_{(H,E)}(H, E, \Omega)\begin{pmatrix} \lambda_H(t) \\ \lambda_E(t) \end{pmatrix} \\
&\quad = \int_\Omega \lambda_H(t)\left[\frac{\partial H'}{\partial t} + \frac{1}{\mu}\operatorname{curl} E'\right]\, dx + \int_\Omega \lambda_E(t)\left[\frac{\partial E'}{\partial t} - \frac{1}{\varepsilon}\operatorname{curl} H' + \frac{\sigma}{\varepsilon}E'\right]\, dx \\
&\quad + \int_{\Gamma_i\backslash\Gamma_{incl}} (H', E')\begin{bmatrix} \tilde{A}_n c & A_n Z c \\ -A_n Y c & \tilde{A}_n c \end{bmatrix}\begin{pmatrix} \lambda_H(t) \\ \lambda_E(t) \end{pmatrix}\, dS \\
&\quad + \int_{\Gamma_{incl}} (H', E')\begin{bmatrix} 0 & -A_n Z c \\ 0 & 0 \end{bmatrix}\begin{pmatrix} \lambda_H(t) \\ \lambda_E(t) \end{pmatrix} dS.
\end{aligned}
$$

Integration by parts for the domain integrals leads to

$$\begin{aligned}
&\int_\Omega \lambda_H(t)\left[\frac{\partial H'}{\partial t}+\frac{1}{\mu}\operatorname{curl}E'\right]\,dx+\int_\Omega \lambda_E(t)\left[\frac{\partial E'}{\partial t}-\frac{1}{\varepsilon}\operatorname{curl}H'+\frac{\sigma}{\varepsilon}E'\right]\,dx\\
&\quad=\int_\Omega -H'\dot{\lambda}_H(t)+\frac{1}{\mu}(E',\operatorname{curl}\lambda_H(t))\,dx\\
&\quad+\int_\Omega -E'\dot{\lambda}_E(t)-\frac{1}{\varepsilon}(H',\operatorname{curl}\lambda_E(t))+\frac{\sigma}{\varepsilon}(E',\lambda_E(t))\,dx\\
&\quad+\int_{\partial\Omega}\frac{1}{\mu}E'(\lambda_H(t)\times n)-\frac{1}{\varepsilon}H'(\lambda_E(t)\times n)\,dS.
\end{aligned}$$

One has

$$\begin{aligned}
&\int_{\partial\Omega}\frac{1}{\mu}E'(\lambda_H(t)\times n)-\frac{1}{\varepsilon}H'(\lambda_E(t)\times n)\,dS\\
&=-\int_{\partial\Omega}(H',E')\begin{bmatrix}0 & -\frac{1}{\varepsilon}A_n\\ \frac{1}{\mu}A_n & 0\end{bmatrix}\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}\,dS\\
&=\int_{\partial\Omega}(H',E')\left(\frac{1}{2}\begin{bmatrix}\tilde{A}_nc & A_nZc\\ -A_nYc & \tilde{A}_nc\end{bmatrix}+\frac{1}{2}\begin{bmatrix}-\tilde{A}_nc & A_nZc\\ -A_nYc & -\tilde{A}_nc\end{bmatrix}\right)\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}\,dS.
\end{aligned}$$

Combining this equation with the expression for $\Gamma_{non-reflective}$ and the outgoing part of $\Gamma_{in/out}$, one obtains

$$\begin{aligned}
&\int_{\Gamma_i\setminus\Gamma_{incl}}(H',E')\left(\frac{1}{2}\begin{bmatrix}\tilde{A}_nc & A_nZc\\ -A_nYc & \tilde{A}_nc\end{bmatrix}+\frac{1}{2}\begin{bmatrix}-\tilde{A}_nc & A_nZc\\ -A_nYc & -\tilde{A}_nc\end{bmatrix}\right)\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}\\
&\qquad-(H',E')\frac{1}{2}\begin{bmatrix}\tilde{A}_nc & A_nZc\\ -A_nYc & \tilde{A}_nc\end{bmatrix}\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}dS\\
&=-\int_{\Gamma_i\setminus\Gamma_{incl}}(H',E')\frac{1}{2}\begin{bmatrix}\tilde{A}_nc & -A_nZc\\ A_nYc & \tilde{A}_nc\end{bmatrix}\begin{pmatrix}\lambda_H(t)\\ \lambda_E(t)\end{pmatrix}dS
\end{aligned}$$

yielding a positive semidefinite operator. Finally, for Γ_{incl} one gets

$$\int_{\Gamma_{incl}} (H', E') \begin{bmatrix} 0 & -A_n Zc \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \lambda_H(t) \\ \lambda_E(t) \end{pmatrix} dS = H' A_n Zc \lambda_E(t).$$

Then we sum up the expressions for the adjoints of the boundary and we formulate the *adjoint equations* for the constraints for H

$$-\frac{\partial \lambda_H(t)}{\partial t}\mu - \frac{1}{\varepsilon}\operatorname{curl}\lambda_E(t) = 0 \qquad \text{in } \Omega \tag{4.13}$$

$$\frac{1}{\varepsilon}(n \times \lambda_E(t)) - A_n Zc\lambda_E(t) = 0 \qquad \text{on } \Gamma_{incl} \tag{4.14}$$

$$\frac{1}{\varepsilon}(n \times \lambda_E(t)) - \frac{1}{2}\lambda_E(t) A_n Zc = 0 \qquad \text{on } \Gamma_{non-reflective} \tag{4.15}$$

$$\frac{1}{\varepsilon}(n \times \lambda_E(t)) - \frac{1}{2}\lambda_E(t) A_n Zc = A_n^\top A_n H^\top - \tilde{A}_n^\top A_n Y E^\top + F_{meas} A_n \qquad \text{on } \Gamma_{in/out} \tag{4.16}$$

and analogously for E

$$-\frac{\partial \lambda_E(t)}{\partial t} + \frac{1}{\mu}\operatorname{curl}\lambda_H(t) + \frac{\sigma}{\varepsilon}\lambda_E(t) = 0 \qquad \text{in } \Omega \tag{4.17}$$

$$\frac{1}{\mu}\lambda_H(t) \times n + \frac{1}{2}\lambda_E(t)\tilde{A}_n c = 0 \qquad \text{on } \Gamma_{non-reflective} \tag{4.18}$$

$$\frac{1}{\mu}\lambda_H(t) \times n + \frac{1}{2}\lambda_E(t)\tilde{A}_n c = \tag{4.19}$$

$$-A_n^\top \tilde{A}_n H^\top + \tilde{A}_n^\top \tilde{A}_n Y E^\top - F_{meas}\tilde{A}_n \qquad \text{on } \Gamma_{in/out}. \tag{4.20}$$

4.5 Derivation of the Shape Derivatives for Maxwell's Equations

In order to determine an expression for the shape gradient for the 3D time-dependent Maxwell's equations, a thorough analysis of the problem stated in Sec. 4.3 is performed. To achieve this, we go step by step through the given problem beginning with the target functional (4.9). First, we start with the regularization term of the target function and then, we continue with the actual tracking-type function. There-

fore, we determine the local derivative at the boundary where the target functional is evaluated. At the end of this chapter, we obtain an explicit expression for the shape gradient of the Maxwell problem. As mentioned in Sec. 3.4, this expression can be utilized for the optimization process to obtain numerical results.

The shape derivative of the perimeter penalization can be directly determined using Eq. (3.27)

$$J = \delta \int_{\Gamma_{incl}} 1 \; dS \;\Rightarrow\; dJ = \delta \int_{\Gamma_{incl}} \langle V, n\rangle \kappa \; dS, \quad \delta > 0.$$

Now, we reuse the general function of the actual tracking function (4.12) derived in the last section. With this reformulation of the target function and the variational formulation of the constraints, we can state the Maxwell problem (4.1) also as

$$\begin{aligned} &\min_{\Gamma_{\text{incl}}} \; J(H, E, \Omega) \\ &= \frac{1}{2} \int_{t_0}^{t_f} \int_{\Gamma_{\text{in/out}}} \left[(H^\top, E^\top) \cdot Q_2 \begin{pmatrix} H \\ E \end{pmatrix} + Q_1 \begin{pmatrix} H \\ E \end{pmatrix} + F_{meas}^\top F_{meas} \right] \; dS \; dt \end{aligned}$$

such that

$$\begin{aligned} 0 = &\int_{t_0}^{t_f} \int_{\Omega} \left[\langle v, \frac{\partial H}{\partial t} \rangle + \frac{1}{\mu} \langle E, \operatorname{curl} v \rangle + \langle w, \frac{\partial E}{\partial t} \rangle - \frac{1}{\varepsilon} \langle H, \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E \rangle \right] \; dx \; dt \\ &+ \int_{t_0}^{t_f} \int_{\partial\Omega} (v, w) h(H, E, n) \; dS \; dt \qquad \forall (v, w) \in \mathcal{H}^1(\Omega). \end{aligned}$$

Following the derivation of a general formulation for the shape gradient for linear state constraints in Sec. 3.7, in particular considering Eq. (3.29), we state the following proposition for an expression for the shape derivative resp. the shape gradient of the Maxwell's equations.

Proposition 4.1 (Shape derivative / shape gradient of Maxwell's equations)**.** *For a given domain Ω the shape derivative of the Maxwell's equations $dJ(H, E, \Omega; V)$ in the direction of the vector field V reads as*

$$dJ(H, E, \Omega; V) = \int_{t_0}^{t_f} \int_{\Gamma_{incl}} \langle V, n \rangle \cdot \operatorname{div} \left(Zc(\lambda_E(t) \times H) \right) \, dS \, dt. \tag{4.21}$$

The adjoint operator λ_E corresponds to the state variable E.
The explicit representation of the shape gradient is $g = \operatorname{div} \left(Zc(\lambda_E(t) \times H) \right)$.

Proof. First, we continue with the local derivative of the target function and determine

$$\begin{aligned}
&\frac{d}{d\varepsilon} \frac{1}{2} \left((H_\varepsilon^\top, E_\varepsilon^\top) \cdot Q_2 \begin{pmatrix} H_\varepsilon \\ E_\varepsilon \end{pmatrix} + Q_1 \begin{pmatrix} H_\varepsilon \\ E_\varepsilon \end{pmatrix} + F_{meas}^\top F_{meas} \right) \Big|_{\varepsilon=0} \\
&= \frac{1}{2} \left((H'^\top, E'^\top) Q_2 \begin{pmatrix} H \\ E \end{pmatrix} + (H^\top, E^\top) Q_2 \begin{pmatrix} H' \\ E' \end{pmatrix} + Q_1 \begin{pmatrix} H' \\ E' \end{pmatrix} \right).
\end{aligned}$$

Replacing Q_2 with $B^\top B$ yields

$$\begin{aligned}
&= \frac{1}{2} \left[\left(B \begin{pmatrix} H' \\ E' \end{pmatrix} \right)^\top B \begin{pmatrix} H \\ E \end{pmatrix} + \left(B \begin{pmatrix} H \\ E \end{pmatrix} \right)^\top B \begin{pmatrix} H' \\ E' \end{pmatrix} + Q_1 \begin{pmatrix} H' \\ E' \end{pmatrix} \right] \\
&= \frac{1}{2} \left[2 \cdot (H^\top, E^\top) B^\top B \begin{pmatrix} H' \\ E' \end{pmatrix} + Q_1 \begin{pmatrix} H' \\ E' \end{pmatrix} \right] \\
&= (H^\top, E^\top) Q_2 \begin{pmatrix} H' \\ E' \end{pmatrix} + \frac{1}{2} Q_1 \begin{pmatrix} H' \\ E' \end{pmatrix}.
\end{aligned}$$

Thus, the derivative of the target function can be expressed as

$$dJ(H, E, \Omega) = \int_{t_0}^{t_f} \int_{\Gamma_{in/out}} \left[(H^\top, E^\top) Q_2 + \frac{1}{2} Q_1 \right] \cdot \begin{pmatrix} H' \\ E' \end{pmatrix} \, dS \, dt \tag{4.22}$$

with the first order optimality system of the constraints

$$0 = \int_{t_0}^{t_f} \int_{\Omega} \left[\langle v, \frac{\partial H'}{\partial t} \rangle + \frac{1}{\mu} \langle E', \operatorname{curl} v \rangle + \langle w, \frac{\partial E'}{\partial t} \rangle - \frac{1}{\varepsilon} \langle H', \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E' \rangle \right] \, dx \, dt$$
$$+ \int_{t_0}^{t_f} \int_{\partial\Omega} D_{(H,E)} h(H, E, n) \begin{pmatrix} H' \\ E' \end{pmatrix} \, dS \, dt. \tag{4.23}$$

It is noticeable that the boundary, where the values for the target function are recorded, does not depend on the shape of the geometric object.
Having the first order optimality system of the constraints (4.23) and applying the derived formula for optimization problems constrained by linear PDEs Eq. (3.28), we obtain an expression for the local shape derivatives for the boundary conditions

$$\int_{t_0}^{t_f} \int_{\partial\Omega} \langle V, n \rangle \left[\langle v, \frac{\partial H}{\partial t} \rangle + \frac{1}{\mu} \langle E, \operatorname{curl} v \rangle + \langle w, \frac{\partial E}{\partial t} \rangle - \frac{1}{\varepsilon} \langle H, \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E \rangle \right] \, dS \, dt$$
$$+ \int_{t_0}^{t_f} \int_{\partial\Omega} \langle V, n \rangle [\langle \nabla \left(h(H, E, n) \right), n \rangle + \kappa \left(h(H, E, n) - D_n \left(h(H, E, n) \right) \cdot n \right)$$
$$+ \operatorname{div}_\Gamma \left(D_n^\top h(H, E, n) \right)] \, dS \, dt$$
$$+ \int_{t_0}^{t_f} \int_{\Omega} \left[\langle v, \frac{\partial H'}{\partial t} \rangle + \frac{1}{\mu} \langle E', \operatorname{curl} v \rangle + \langle w, \frac{\partial E'}{\partial t} \rangle - \frac{1}{\varepsilon} \langle H', \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E' \rangle \right] \, dx \, dt$$
$$+ \int_{t_0}^{t_f} \int_{\partial\Omega} D_{(H,E)} (h(H, E, n)) \cdot \begin{pmatrix} H' \\ E' \end{pmatrix} \, dS \, dt = 0.$$

Due to first order optimality conditions, the boundary conditions on the facet, where the target function operates, have to vanish. In particular, this comprises the first order derivative of the general boundary conditions of the inlet boundary derived in Sec. 4.2 as well as the first order derivative of the target function (4.22). Hence,

choose a $(\lambda_H(t), \lambda_E(t)) \in \mathcal{H}^1(\Omega)$ such that

$$0 \stackrel{!}{=} \int_{t_0}^{t_f} \int_{\Gamma_{in/out}} \left[\frac{1}{2}(\lambda_H(t), \lambda_E(t)) \begin{pmatrix} \tilde{A}_n c & A_n Y c \\ -A_n Z c & \tilde{A}_n c \end{pmatrix} + (H^\top, E^\top) Q_2 + \frac{1}{2} Q_1 \right] \cdot \begin{pmatrix} H' \\ E' \end{pmatrix} \, dS \, dt.$$

Now, we can derive the shape gradient with respect to the obstacle for our Maxwell problem, which obviously only exists on the surface of the obstacle Γ_{incl} and is expressed by

$$\begin{aligned} &dJ(H, E, \Omega; V) = \\ &= \int_{t_0}^{t_f} \int_{\Gamma_{incl}} \langle V, n \rangle \left[\langle v, \frac{\partial H}{\partial t} \rangle + \frac{1}{\mu} \langle E, \operatorname{curl} v \rangle + \langle w, \frac{\partial E}{\partial t} \rangle - \frac{1}{\varepsilon} \langle H, \operatorname{curl} w \rangle + \frac{\sigma}{\varepsilon} \langle w, E \rangle \right] \, dS \, dt \\ &+ \int_{t_0}^{t_f} \int_{\Gamma_{incl}} \langle V, n \rangle [\langle \nabla \left(h(H, E, n) \right), n \rangle + \kappa \left(h(H, E, n) - D_n \left(h(H, E, n) \right) \cdot n \right) \\ &\qquad + \operatorname{div}_\Gamma (D_n^\top \left(h(H, E, n) \right)] \, dS \, dt. \end{aligned}$$

In particular, on $\Gamma_{incl} = \Gamma_{reflect}$, one has

$$\begin{aligned} h_{reflect}(s, n) &= (\lambda_H(t), \lambda_E(t)) \begin{bmatrix} 0 & 0 \\ -A_n Z c & 0 \end{bmatrix} \begin{pmatrix} H \\ E \end{pmatrix} = (-\lambda_E(t) A_n Z c, 0) \begin{pmatrix} H \\ E \end{pmatrix} \\ &= -\lambda_E(t) A_n H = \lambda_E(t) \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} \cdot H. \end{aligned}$$

This yields

$$D_n h(H, E, n) = \lambda_E(t) Z c \begin{bmatrix} 0 & H_z & -H_y \\ -H_z & 0 & H_x \\ H_y & -H_x & 0 \end{bmatrix}, \quad D_n h(H, E, n) \cdot n = h(H, E, n).$$

In this case the linearity of the operator h can be exploited, cf. (3.29). Thus, the expression of the gradient can be reduced to

$$
\begin{aligned}
&dJ(H,E,\Omega;V)\\
&= \int_{t_0}^{t_f}\int_{\Gamma_{incl}} \langle V,n\rangle \left[\langle v,\frac{\partial H}{\partial t}\rangle + \frac{1}{\mu}\langle E,\operatorname{curl} v\rangle + \langle w,\frac{\partial E}{\partial t}\rangle - \frac{1}{\varepsilon}\langle H,\operatorname{curl} w\rangle + \frac{\sigma}{\varepsilon}\langle w,E\rangle\right] \, dS\; dt\\
&+ \int_{t_0}^{t_f}\int_{\Gamma_{incl}} \langle V,n\rangle \cdot \operatorname{div}\left(Zc \begin{bmatrix} 0 & H_z & -H_y \\ -H_z & 0 & H_x \\ H_y & -H_x & 0 \end{bmatrix}^{\top} \lambda_E(t)\right) \, dS\; dt
\end{aligned}
$$

Under the assumption that the trace of the PDE vanishes on the boundary if the problem is smooth enough, we finally obtain the shape gradient

$$
dJ(H,E,\Omega;V) = \int_{t_0}^{t_f}\int_{\Gamma_{incl}} \langle V,n\rangle \cdot \operatorname{div}\left(Zc(\lambda_E(t)\times H)\right) \, dS\; dt. \tag{4.24}
$$

□

With this important result, we have an explicit representation of the shape gradient and we are able to perform a first numerical verification. In the next chapter, a numerical analysis with various test cases can be found.

5

Numerical Results and Discussion

In this chapter, first numerical results confirm the theoretical formulation of the shape gradient. Therefore several tests based on different test-setups are analyzed.

The following excitations are considered

- point source (e.g. dipole antenna),
- plane waves (e.g. horn antenna),
- sine or a modulated Gaussian pulse.

These excitations have to be modelled in an appropriate way. Moreover the solution of the Maxwell's equations as well as the solution of the adjoints have to be defined. The numerical realization of the various test cases builds on features provided by the software project FEniCS, which is described in the following.

Used Modelling Software The software package *FEniCS* [45, 46] is a collaborative project, which comprises several other projects such as the problem-solving environment dolfin, the form compiler FFC, the form language UFL, the finite element tabulator FIAT and many more. The software package dolfin can be used as standalone solver for PDEs. In particular, it focuses on the automated solution of differential equations by finite element methods including discontinuous Galerkin methods. It serves as a tool for development and implementation of new methods. Since the code is near mathematical notations, it is comparatively easy to read. This project forms a fundamental basis regarding our problem-dependent implementation for the forward integration of the PDEs. Although there exists an extension for dolfin to compute the corresponding adjoints (*dolfin-adjoint*, *libadjoint* [25]), this extension does not provide needed facilities to compute the adjoint solutions of the tracking type functional considered here. Since shape derivatives live on the boundary of the inclusion, a data compression to the boundary would be desirable for an efficient implementation. Another advantage of data compression to the boundary is that all forward and adjoint variables can be stored for every timestep. Consequently, there is no necessity to apply a checkpoint strategy. Due to these reasons we reuse the manually derived adjoint equations from Sec. 4.4, solve them in dolfin and use the results for calculating the shape gradient.

For the conservation of energy in the differential system, a symplectic procedure for the discretization in time is the method of choice. Here, we apply a simple Runge–Kutta method: the implicit midpoint rule, which is a symplectic integrator. For a general state u_t, the next state u_{t+1} for the timestep $t+1$, can be calculated through:

$$u_{t+1} = u_t + \Delta t[(1-\alpha)f(u_t) + \alpha f(u_{t+1})].$$

For our calculations, we use $\alpha = 0.5$. Since f is linear, the implicit midpoint method corresponds to the Crank–Nicolson method [19]. In the implemented code, we apply this method for the Maxwell's equations. Since the chosen method is implicit, a linear algebraic equation system arises and has to be solved. FEniCS also provides several preconditioners, we choose an iterative solver of Krylov type, namely the parallel preconditioner *Hypre* [24].

5.1 Identification of Geometric Objects

One of the main issues of the real-life problems presented in Chap. 1 is the detection of geometric objects in a predefined computational domain. In this section, we concentrate on the identification of ellipsoids resp. spheres and a cuboid. First, some results using an indicator function are presented and second, a similar test case is solved via a shape optimization method.

5.1.1 First Results: Approach Using Indicator Function

A similar test scenario has been analyzed using the approach based on an appropriate indicator function. This indicator function determines the material distribution within the discretized domain and is described in Sec. 3.

The test-setup is the following

- 3D case: discretized Cartesian grid into 50^3 grid cells
- FDTD-approach for solving the PDEs
- modulated Gaussian pulse as excitation
- 100 timesteps, smoothing parameter $h = 0.001$
- automatic differentiation for computing the derivatives (ADOL-C, [29])
- target functional of tracking type
- BFGS optimization algorithm [12]

The whole test scenario was implemented in the script language Python.

Testcase Ellipsoid In this test setup, we want to determine the significant parameters of an ellipsoid. These are the center c and the length of the semi-principal axes r in x, y, and z direction. A suitable formulation of an indicator function is given in Sec. 3. To generate the reference data, an ellipsoid of radius $r_x = r_y = r_z = 5$ is placed into the middle of the computational domain (see also Tab. 5.1). Then we perform a forward simulation using the FDTD-approach and measure and store the primitive field values at a predefined sink. For the optimization part, we have to define an initial guess. Here, the radius in x direction is modified to $r_x = 6$. Then we

	c_x	c_y	c_z	r_x	r_y	r_z
reference data	25	25	25	5	5	5
initial guess	25	25	25	6	5	5

Table 5.1: Testcase ellipsoid: reference data and initial guess.

repeat the forward simulation with the initial guess and compare the characteristic field values at the sink in an L_2-norm. Furthermore, we compute the gradient of the objective function. In advance, one has to define suitable termination criteria, e.g., the relative reduction of the objective function is smaller than a predefined parameter or the norm of the gradient is smaller than a predefined parameter. As soon as one termination criterion is satisfied, the optimization process stops.

Analyzing the optimization results, one can recognize a significant decrease in the magnitude of the objective function (c.f. Fig. 5.1). Also, the value of the initial guess $r_x = 6$ converges to the reference value 5. The decrease of the relative error of r_x is displayed in Fig. 5.2. Therefore, one can conclude that this approach is satisfying for very simple cases.

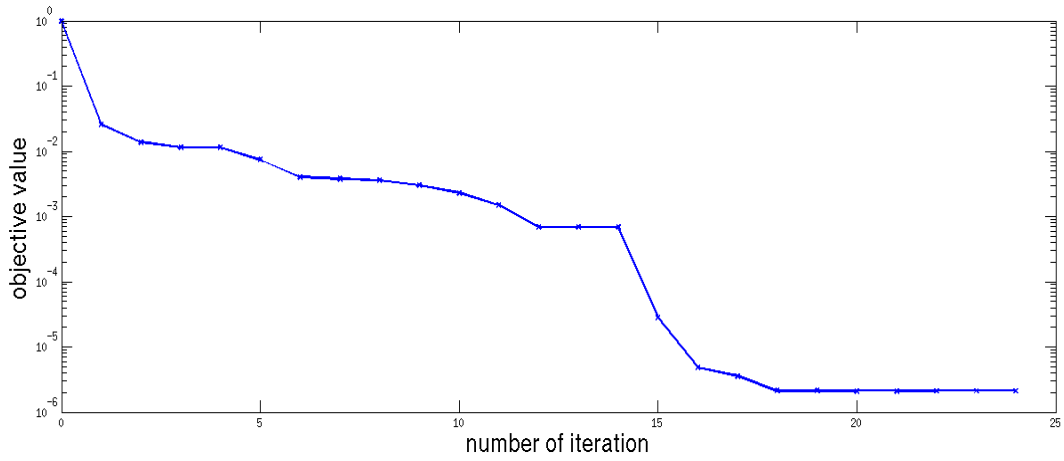

Figure 5.1: Testcase ellipsoid. Reduction of objective function.

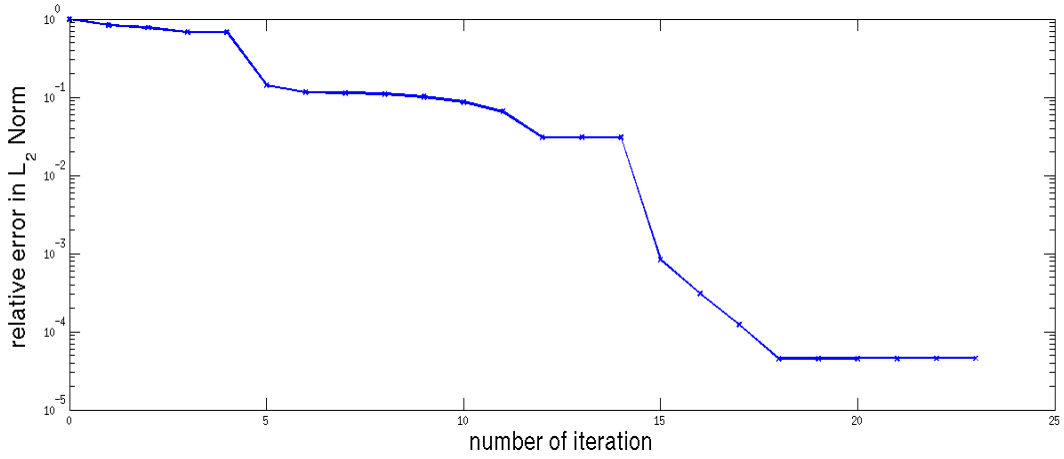

Figure 5.2: Testcase ellipsoid. Relative error of r_x.

Considering other geometric objects, these results cannot be transfered. One of those examples is described in the following.

Testcase Cuboid The subject of this task is to identify the location of a cuboid in a predefined computational domain (cf. Fig. 5.3). For the identification of the sig-

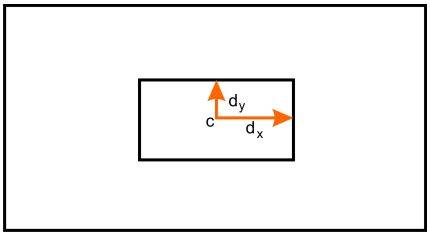

Figure 5.3: 2D representation of cuboid.

nificant parameters of a cuboid, an appropriate indicator function has to be defined. For a cuboid it is given by

$$I(x,y,z) = \prod_{i=\{x,y,z\}} \frac{1}{2}\left(\frac{i+d_i}{\sqrt{(i+d_i)^2+h^2}} - \frac{i-d_i}{\sqrt{(i-d_i)^2+h^2}}\right),$$

where d_i are the semi-axis in x, y, and z direction and h a small smoothing parameter. Therefore, we consider a testcase with 6 parameters, namely the center c_i and the

semi-axis d_i, to be determined.

As a very simple scenario, we again only modify the parameter for the semi-axis in x direction in the initial guess. All other parameters remain the same as in the reference data. The concrete parameters are the following:

	c_x	c_y	c_z	d_x	d_y	d_z
reference data	25	25	25	2	7	10
initial guess	25	25	25	3	7	10

Table 5.2: Testcase cuboid: reference data and initial guess.

As one can recognize in Fig. 5.4, the optimization using the indicator approach is not successful. Although the objective value decreases around one magnitude, the parameter d_x does not converge to the optimal value $d_x^* = 2$. The main error of r_x is hardly decreasing (cf. Fig. 5.5). This failure of convergence is due to several reasons. One is the choice of the smoothing parameter h and consequently, the not sufficiently accurate representation of the geometric object. Also, other variations

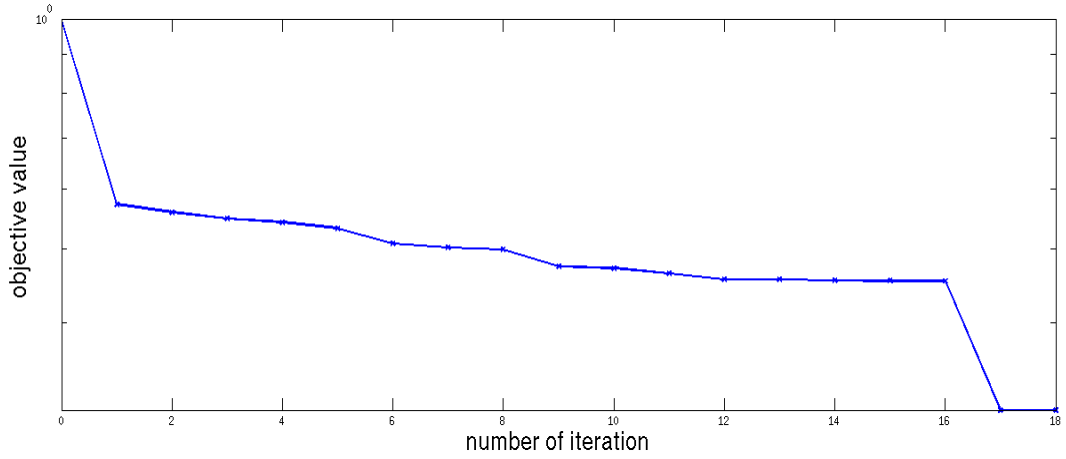

Figure 5.4: Testcase cuboid. Reduction of objective function.

of the initial guess do not lead to sufficient results due to the detection of local minima or due to the reason that the optimizer does not find a descent direction.

Another main problem for this identification task is the fact that for every different geometric object a new mapping between the indicator function and the design

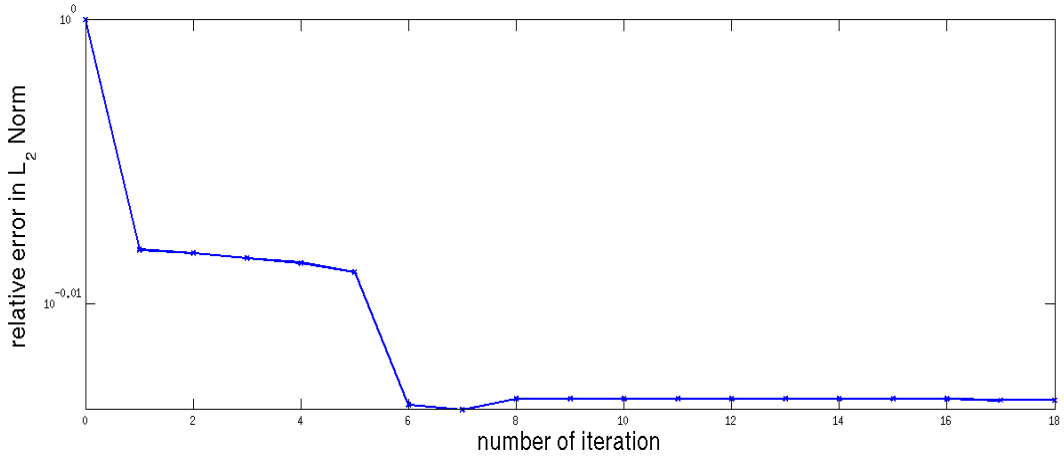

Figure 5.5: Testcase cuboid. Relative error of r_x.

parameters such as the semi-principal axes has to be defined. Especially for arbitrary geometric objects, this additional definition becomes a problem since the approach is not universally applicable to arbitrary objects and mistakes in the user-defined function might arise.

Of course, the presented test cases seem to be very simple problems but even for these cases, the limitations of the easy-to-implement approach become obvious.

5.1.2 Optimization via Shape Optimization Techniques

The proposed approach, based on the theoretical findings presented in the previous chapters, is implemented in FEniCS. To model the geometric domain properly, a suitable software is necessary since dolfin only provides simple built-in meshes such as unit cubes. It is recommended to use an external tool like *gmsh* [26] for the generation of more complicated meshes. With a dolfin command the generated meshes can easily be converted into the required .xml file format. For the following testcases, all initial meshes are generated with gmsh and furthermore, preprocessed with a very robust CVT-based mesh deformation algorithm [61]. All other meshes within the optimization algorithm as well as the reference solutions are processed with the mentioned mesh deformation algorithm such that the topological equivalence with the same connectivity can be preserved.

5.1.3 Example: Smooth Bump

The challenge of this testcase is to reconstruct a smooth bump located in a rectangular cuboid. The pulse, which we are measuring, is a near-field pulse. An overview of the most important parameters can be found in Tab. 5.3.

parameter	value
dimensions of cuboid	24 cm × 24 cm × 12 cm
wavelength Wv	$\frac{299792458\ \mathrm{ms}^{-1}}{4.1\ \mathrm{GHz}} \approx 7.31$ cm
center frequency f_c	8.2 GHz
spread f_s	4.1 GHz
scanable frequencies	4.1 GHz – 12.3 GHz
	≡ waves ranging between 2.4 cm and 7.3 cm
elevation caused by bump	≈ 3.65 cm
number of timesteps	548
duration of 1 timestep	6.1 ps (→ total time 3.336 ns)
perimeter penalization δ	$5 \cdot 10^{-5}$
unknowns for one primal trajectory	
for the whole domain	472 748 640 (≈ 3.52 GB storage)

Table 5.3: Test-setup smooth bump.

The bump, which we want to identify, is created on the rear side of the cuboid and relevant for computing the reference solution. I.e., based on this modified geometry a forward simulation is performed and later, the results serve as measured data F_{meas}. For our testcase the sigmoid function is given by

$$z_{new} = z + \frac{-Wv}{2 + 2\exp(100(\sqrt{x^2 + y^2} - Wv))},$$

where Wv describes the wavelength of a sinusoidal pulse. The expression under the square root defines the form of the bump, in our case the form of a sphere. The sigmoid function has the characteristic to operate smoothly on the facet. This is very important for the meshing process to end up with a suitable mesh for the simulation and optimization process.

Another important element of the modeling process is the choice of the entering pulse on $\Gamma_{in/out}$, which determines how the waves propagate through the domain. A common choice of the pulse is a sinusoidal pulse. For our problem, we decided to

take a truncated SINC-pulse of the following form

$$g(t) = \frac{\sin(2\pi f_s(t-t_c))}{2\pi f_s(t-t_c)} \sin(2\pi f_c(t-t_c))w(t,t_c).$$

The parameter f_c describes the desired center frequency, f_s the half bandwidth of the signal and t_c determines the time around which the pulse is centered. $w(t,t_c)$ denotes the Hamming window function, which is also of sine form, and given by

$$w(t,t_c) = \begin{cases} 0.54 + 0.46\cos\left(\frac{\pi(t-t_c)}{t_c}\right), & \text{for } t \in [0,2t_c], \\ 0, & \text{otherwise.} \end{cases}$$

As noted in Tab. 5.3 the total time of one complete forward run is 3.336 ns. This period of time corresponds to the travelled distance of 1m of the wave. Due to the domain size of 12 cm, it is realistic to be able to measure all the reflections of the object at $\Gamma_{in/out}$.

Due to the symmetry of the considered object, it is enough to compute only a quarter section leading to a significant reduction of computation time. The quarter of the geometry is shown in Fig. 5.6. One quarter section is discretized using 35 945 elements. Thus, 0.88 GB are required to store one primal trajectory. By exploiting the boundary representation of the gradient, additional computational cost can be saved.

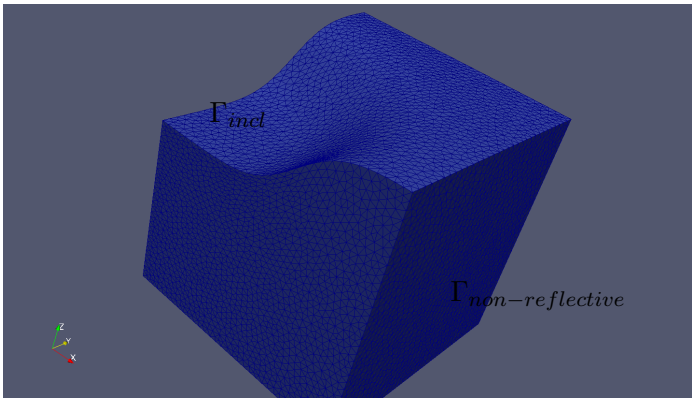

Figure 5.6: Cut through the near-field test-case. A quarter of the reference geometry shown with its discretization.

The optimization procedure is the following: Starting with a flat rear side of the cuboid, a forward simulation is performed. Afterwards, the gradient and the mismatch between the simulated data and data of the reference solution is computed. While no termination criterion is satisfied, the gradient step is performed leading to a new mesh. With this modified mesh, we reenter the optimization loop at the beginning with a forward simulation and repeat the above mentioned steps.

For this testcase, we obtain very convincing results for the reconstruction, which are illustrated in Figs. 5.7 and 5.8. On the right-hand side of these figures the target shape is displayed whereas on the left-hand side the reconstructed shape after 373 optimization steps is shown. It is remarkable that the form of the bumps nearly perfectly coincides.

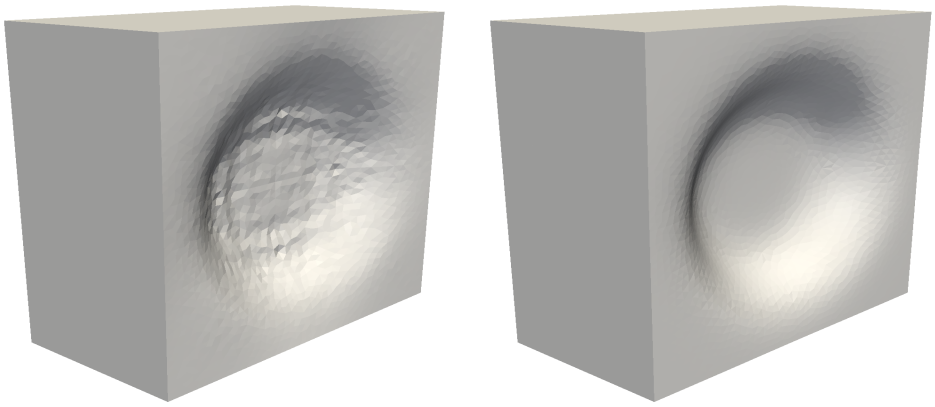

Figure 5.7: Left: reconstructed shape after 373 optimization steps. Right: target shape.

The graph for the reduction of the objective value and the reduction of the norm of the gradient (both in Fig. 5.9) support the observation of the quality of the reconstruction. In Fig. 5.9 the objective value and the gradient norm are normed for a clearer interpretation of the evolution of the specific graphs. As can be seen in the figure, the two graphs describe a similar behavior, as one could have expected before. The decrease in the gradient norm does not seem as significant as in the objective norm, but this can be explained by the fact that the value of the gradient norm is already very small in the first iteration, namely $\approx 2.33 \cdot 10^{-5}$ whereas the objective value is ≈ 1.32.

Figure 5.8: Cross-sectional comparison between reconstructed bump after 373 optimization steps and target bump.

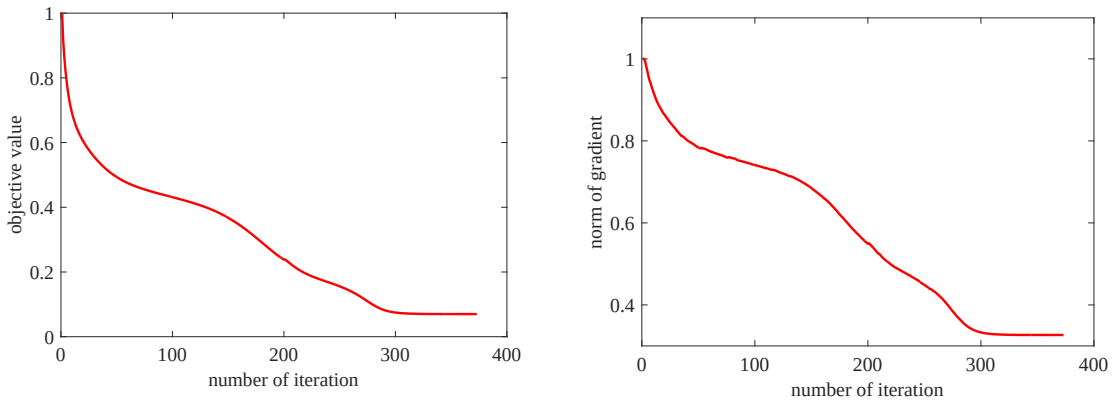

Figure 5.9: Left: relative decrease of objective value. Right: relative decrease of gradient norm.

5.1.4 Reconstruction of a Smooth Bump within Horn Antenna Test-Setup

The successful reconstructed bump of the testcase described in the last subsection motivates to extend the testcase with an antenna setup. The test setup was already presented in Chap. 1, Fig. 1.1. An overview of the most important parameters can be found in Tab. 5.4.

In contrast to the near-field situation in the last subsection, the horn antenna and

parameter	value
waveguide geometry	E-band WR229 waveguide
dimension of waveguide	5.8166 cm × 2.921 cm
attached horn	Pasternack PE9862-20
dimensions of horn	59.563 cm long, mouth: 31.496 cm × 22.225 cm
ideal frequencies	3.3 GHz – 4.9 GHz
elevation caused by bump	≈ 3.65 cm
number of timesteps	1505
duration of 1 timestep	12.2 ps (→ total time 18.36 ns)
unknowns for one primal trajectory for the whole domain	1 202 976 600 (≈ 8.96 GB storage)

Table 5.4: Test-setup smooth bump within horn antenna setup.

waveguide have the role of a polarization filter, meaning that only linearly polarized waves pass through. Due to the ideal frequency, denoted in Tab. 5.4, we have a center frequency of $f_c = 4.1$ GHz and a spread of $f_s = 0.41$ GHz.

As in the last testcase, we also perform our calculations on the quarter section to reduce computational cost. The quarter section is discretized using 33 305 elements. Thus, 2.24 GB are required to store one primal trajectory.

The principal procedure of the optimization process stays the same as in the last testcase, whereas the penalty term is set to zero here. Starting with a flat plate, the optimization process ends up with a final shape. In Fig. 5.10 the reflections of the initial geometry, meaning the flat plate, the reconstructed geometry and the reference measured by the antenna are shown. Here, one can notice that the signal of the reconstructed geometry and the reference solution coincides very fine, which may lead to the conclusion that the reconstructed shape also coincides with the target shape. Unfortunately, this is not the case as can be seen in Fig. 5.11.

One can only guess where the bump is approximatley located, but without knowing the target shape in advance, the findings of the reconstructed shape are limited. At a first glance, the results seem to contradict each other, but here, one can certainly denote that a local minimum is found. Perhaps the effect of polarization filtering by the antenna, is not an ideal situation as well.

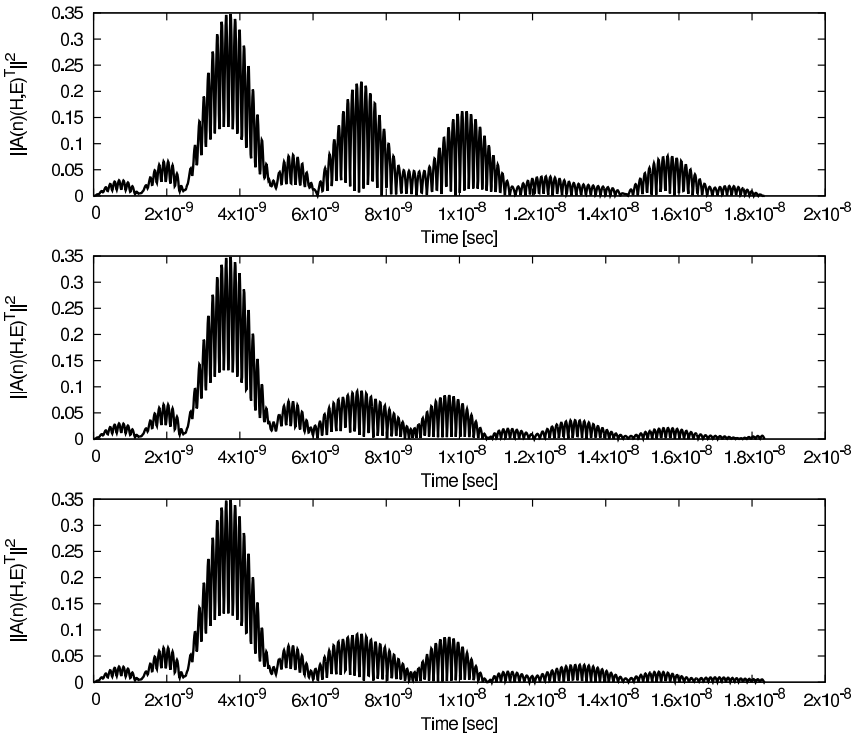

Figure 5.10: Reflections of the initial geometry, the reconstructed geometry and the reference geometry.

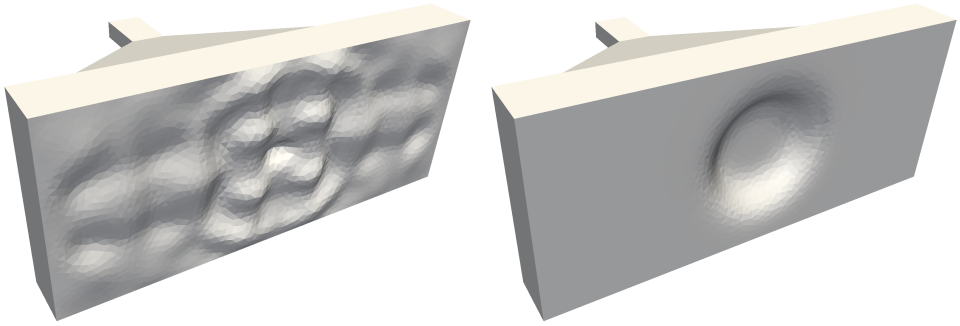

Figure 5.11: Reconstructed and target shape for the horn antenna testcase after 376 optimization steps.

5.2 Scaling Tests

In this section, we analyze the scalability of our approach. In particular, we examine the scalability in terms of strong scaling, i.e., the problem size stays fixed and the number of processing elements is increased. The weak scaling, where the workload assigned to each processing element stays the same and the problem size varies, is very difficult to analyze since the meshing process is very complex and difficult to influence.

In Tab. 5.5 significant parameters of each mesh are listed. The finest discretized mesh, mesh #3, has around 3 times more tetrahedrons than grid #2 and around 20 times more tetrahedrons than the coarsest mesh, mesh #1. Bigger grids are difficult to handle due to memory reasons. The enormous memory requirements are also the reason why there were no test workable for one process for the finest grid.

All scalability tests were performed with 30 timesteps, while being aware of the fact that a propagation through the whole domain is not possible with this unrealistic small number of timesteps. Nevertheless, scalability results can be obtained.

mesh id	refinement level	# of tetrahedrons	number of unknowns for one primal trajectory
#1	0	7418	1 335 240 (≈ 10.19 MB storage)
#2	1	48230	8 681 400 (≈ 66.23 MB storage)
#3	2	139168	25 050 240 (≈ 0.19 GB storage)

Table 5.5: Significant data of tested meshes.

Scalability of the Calculation of the Target Function For the analysis of the implemented approach, we measure the time, which is needed for the evaluation of the target function in one iteration. Considering Figs. 5.12 and 5.13, one recognizes that the computation for the finest grid scales best since the graph is nearly a line whereas the graphs corresponding to the coarser meshes rather resemble curves. The great scaling behavior, illustrated in Fig. 5.13 can be explained by a good ratio between number of processes and mesh size. The workload for each process up to 6 processes seems to be very efficient for a significant speedup. Using 8 or 12 processes also lead to a decrease in the runtime, but less significant.

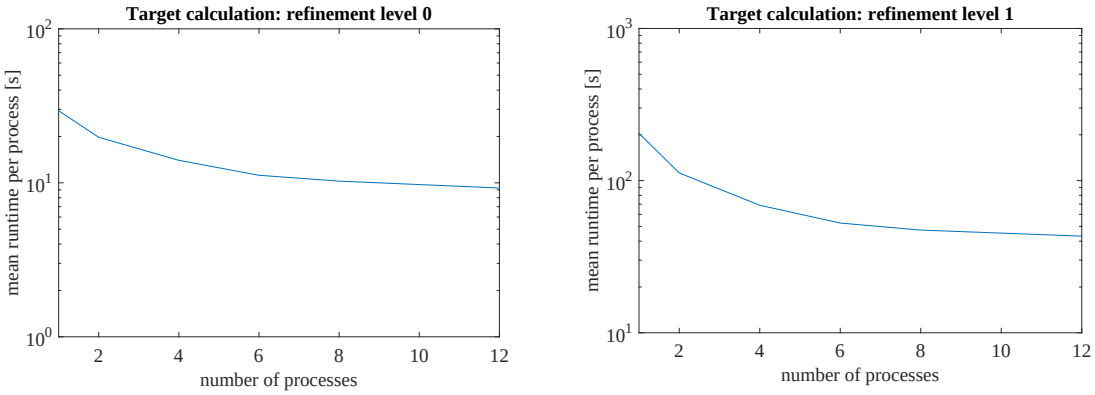

Figure 5.12: Calculation of target function for different refinement levels.

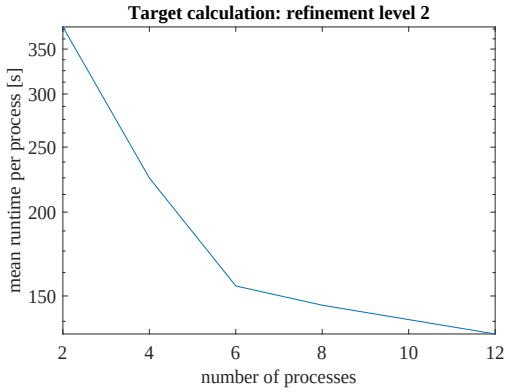

Figure 5.13: Calculation of target function for refinement level 2.

Scalability of Gradient Computation To determine the time of the gradient computation, we measure the time, which is needed to calculate the gradient expression for all timesteps of one optimization iteration. The obtained runtimes for the gradient computation are illustrated in Fig. 5.14, where the runtime is proportional to the mesh size. Another observation of Fig. 5.14 is that for the coarsest mesh, the needed time for the gradient computation is nearly the same independent of the number of processes. Additionally, the computation only takes a few seconds. Analyzing the runtime for mesh #2, one can recognize that there is a remarkable discrepancy in the runtimes obtained by 1, 2 and 4 processes. Using more than 4 processes does not lead to a significant reduction in runtime. The computation of the gradient of the finest mesh was not possible by 1 process due to memory requirements. However, for the usage of more processes, there can be a similar behavior as for mesh #2 observed. Here, the highest speed-up can be obtained for 2 and 4 processes.

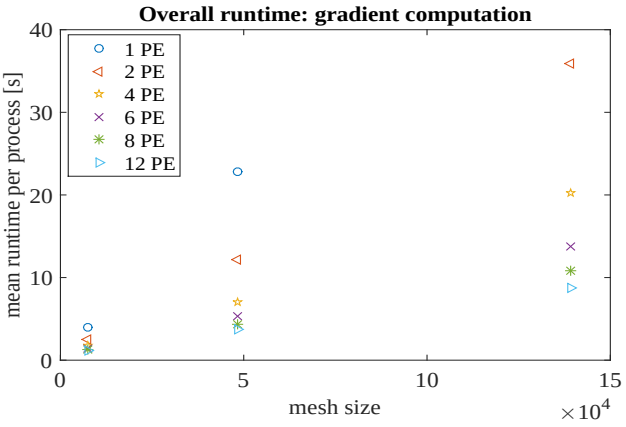

Figure 5.14: Overall runtime needed for gradient computation for increasing processing elements.

For further analysis of the runtime, we examine the data in terms of weak scaling. In Fig. 5.15, one can notice, that the scaling for refinement level 2 is slightly better. Furthermore, for mesh #1 and mesh # 2 the highest speed-up can be obtained for 1 and 2 processing elements. This result agrees with the above observation.
In summary, it can be stated that the gradient computation shows a weak scaling.

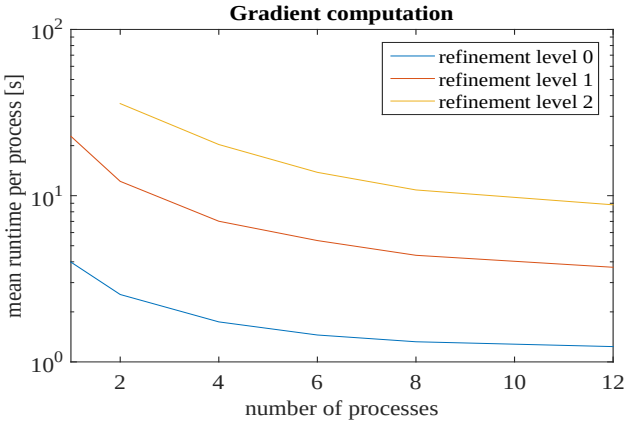

Figure 5.15: Time needed for one gradient computation for different refinement levels.

5.3 Preliminary Results: Shape Quasi-Newton Methods

In the field of shape optimization, steepest descent methods are very often used. In this thesis, we also use a steepest descent method (3.8) with a gradient smoothing technique of Newton-type. As alternative shape Newton optimization methods exploiting a Riemannian perspective can be applied. In [62] a Riemannian view on shape optimization is discussed. Basic operations in the tangent space of shape manifolds are transferred to classical shape optimization. This ansatz is revisited in [64, 65, 66], and a limited memory Broyden-Fletcher-Goldfarb-Shanno (l-BFGS) algorithm is reformulated for shape optimization. In contrast to the conventional usage of the BFGS method in nonlinear optimization, no linesearch strategy is applied here due to additional computational cost. In the cited papers, the usage of an l-BFGS algorithm leads to very good convergence results.

In this thesis, we reuse the proposed algorithm in an adapted form. Within this algorithm $a(\cdot,\cdot)$ denotes a scalar product, which is given by the symmetric positive definite stiffness matrix. U_j contains the domain representations of the gradient and Y_j is computed via $Y_j = U_{j+1} - U_j$. The vector S_j is a domain deformation vector. The stored gradients and deformation vectors of former iterations have to be interpolated to the current mesh in each iteration. The scalar m determines the number of stored gradients from previous iterations. If m equals zero, a steepest

descent method as described in Sec. 3.3 is applied.
The following algorithm only describes the l-BFGS method needed for the optimization step within the shape optimization cycle. The operations to transfer the vectors from the shape space to the finite dimensional space are omitted since the vectors U_j and S_j are already finite dimensional vectors in memory.

The l-BFGS algorithm directly generates an update vector q for the shape deformation. The update vector q is initially set to the current (smoothed) gradient and is modified via the rephrased BFGS-update, which consists of two for-loops and requires the evaluation of scalar products, where previous gradients and previous deformation fields take part.

Data: j denotes current outer iteration number, m denotes number of stored gradients from past iterations
Result: vector q, which serves as shape deformation
$\rho_j = \frac{1}{a(Y_j, S_j)}$
$q = U_j$
for $i = j-1, \cdots j-m$ **do**
  $\alpha_i = \rho_i \cdot a(S_i, q)$
  $q = q - \alpha_i Y_i$
end
$q = \frac{a(Y_{j-1}, S_{j-1})}{a(Y_{j-1}, Y_{j-1})} q$
for $i = j-m, \cdots j-1$ **do**
  $\beta_i = \rho_i \cdot a(Y_i, q)$
  $q = q + (\alpha_i - \beta_i) S_i$
end

Algorithm 1: Adapted l-BFGS algorithm

In the presented testcase, the l-BFGS method with 5 gradients in storage is applied. Due to the storage of previous gradient information, a better convergence is expected. As can be seen in Fig. 5.16, the l-BFGS method does not lead to better convergence results compared to the steepest descent method with gradient smoothing of Newton-type. Although the objective value obtained in the third iteration is

slightly smaller using l-BFGS, the algorithm performs worse. This can be due to several reasons. It turned out that the success of the algorithm is closely related to the stepsize of the steepest descent method since the first iteration determines the deformation field, which is reused in the l-BFGS algorithm later. In the presented testcase the stepsize of the steepest descent algorithm equals 0.5. Other tests with stepsize = 2.5 or stepsize = 1 failed in terms of a monotone decrease of the objective value. Another possibility is the usage of another scalar product as it is done, e.g., in [65]. Furthermore, the convergence of the BFGS algorithm can be accelerated by combining the BFGS update and the smoothing of the gradient in a more sophisticated way. However, Shape Quasi-Newton methods are not focus of this thesis and there has to be done further investigative work in order to determine appropriate parameters for this testcase.

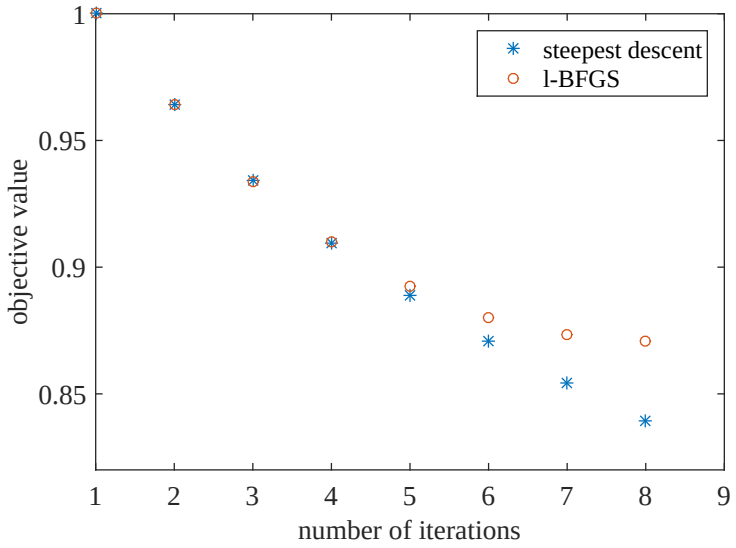

Figure 5.16: Comparison of objective value using steepest descent method and l-BFGS method.

6 Conclusion

This work captures a wide range of aspects from electromagnetic theory over shape sensitivity analysis to shape optimization. Challenges were solved in fields of modelling, theoretical analysis, and simulation/implementation issues. This thesis contributes new and beneficial results to all of the mentioned aspects. Therefore, it is important to catch the main ideas, visions and outcoming results altogether.

6.1 Summary

In many real-life applications, we are faced with problems detecting interior structures and material distributions of geometric objects without destroying the geometry. Formulating these problems as inverse electromagnetic scattering problems, and solving these afterwards, usually lead to good results.

First, an appropriate electromagnetic problem was defined. A Galerkin method turned out to fit best to our application. Hence, a suitable flux function covering

discontinuities at inner interfaces had to be derived. Since different boundary conditions rule on the different facets, the behavior of the characteristic variables had to be modelled as well. Directly related to the defined (forward) problem, the derivation of corresponding adjoint equations was indispensable to be able to perform shape optimization. The value of one adjoint variable takes directly part in the explicit expression of the shape gradient. Here, we only store the boundary representation of the gradient, which leads to a high reduction of the memory requirement. To the knowledge of the author, the theoretical derivation of the shape gradient for the fully time-dependent 3D Maxwell's equations and its implementation within a shape optimization procedure, was the first of its kind.

We defined a generalized optimization problem with a tracking type target function constrained by a linear hyperbolic partial differential system and boundary conditions. We have used basic results of shape analysis from Zolésio to develop a general formulation for the shape gradient for linear state constraints. These findings were transferred to the target problem constrained by Maxwell's equations.

The suggested approach, comprising simulation, gradient computation, and optimization, was implemented and tested in FEniCS.
Two different testcases were examined to validate the proposed approach. The first one was given a rectangular cuboid, where a smooth bump on the rear side had to be identified. Here, our method yielded excellent results, since not only the discrepancy between the simulated and measured data almost vanished, but also the reconstructed geometry was almost identical to the target geometry. To extend these convincing results, we constructed a testcase, where the smooth bump of the last testcase was embedded within a horn antenna setup. There, the waves entered the computational domain via a waveguide. It is noticeable, that this waveguide only let polarized waves into the domain. As in the last testcase, we could obtain perfect results in terms of matching the reflections. However, the reconstructed geometry did not coincide with the reference geometry. This might be caused by the fact, that the optimization method is a local method and probably found a local minimum, here. Additionally, the polarization filtering might have caused problems for the reconstruction process.

Moreover, the implemented approach was analyzed in terms of scaling. There, we had a closer look at the computation time of the target function and of the gradient

evaluation. We obtained good results showing that our approach scales well. Therefore, we can also expect a good scaling behavior for larger meshes on appropriate working machines.

6.2 Future Research Directions

The theory and software developed in this thesis provides a good basis for further research. The results presented in Chap. 5 demonstrate the practicability of the proposed approach. At present the reconstruction of the smooth bump within the horn antenna setup was not as good as expected and the origin of the problem might be the setup. Therefore, further investigations to find a suitable experimental setup are needed. Additionally, one can think of other possible setups for different applications.

A faster decrease of the reduction of the target function was desired by using a Shape Quasi-Newton method. These expectations could not be satisfied by preliminary results obtained in this thesis. However, the incorporation of Shape Quasi-Newton methods was not focus of this thesis and can be further developed in future research.

Another challenging task, which is certainly an interesting expansion of this thesis, is to analyze a substitution of the boundary condition at the interface Γ_{incl} by (at least) one PDE. This PDE describes the evolution of the phase field function, in particular how the structure of the interfaces changes with respect to time, e.g., for the transition between two stages. The phase field method is often applied in materials science for the simulation of microstructure evolution during solidification. An overview and further references are given in [73]. The incorporation of a phase field method opens our application for many other interesting applications. Since this incorporation will lead to additional PDEs, which have to be solved, the problem becomes even more complex.

Another aspect that can be analyzed in future work is the consideration of geometric uncertainties. This is quite often an important topic in real-life applications. Geometric variations along a boundary of a geometric object can model changes in the geometry due to naturally occuring wear or erosion but also due to manufacturing imperfections. These problems may directly affect the reliability and the efficiency

of the machines. Different approaches can be applied to incorporate geometric uncertainties and to solve the arising problem properly, e.g. [15, 63]. To counteract the additional computational effort caused by the more complicated task, a Karhunen-Loève expansion is used for dimension reduction in both ansatzes. In [63] one-shot methods are employed to solve the problem and [15] concentrates on a level-set based approach. The modelling of the geometric variation is realized by a random velocity field acting on the normal component of the velocity field. A noticeable challenge is that the shape sensitivity analysis is usually defined for deterministic boundary formulations, whereas for geometric uncertainties random boundaries have to be taken into account for the derivation of the shape derivative. Details can be found in [15].

Bibliography

[1] E. Abenius. Time-Domain Inverse Electromagnetic Scattering using FDTD and Gradient-based Minimization. Licentiate Thesis, 2004.

[2] R. Alcaraz and J.J. Rieta. A review on sample entropy applications for the non-invasive analysis of atrial fibrillation electrocardiograms. *Biomedical Signal Processing and Control*, 5(1):1 – 14, 2010.

[3] W. K. Anderson, L. Wang, S. Kapadia, C. Tanis, and B. Hilbert. Petrov-Galerkin and discontinuous-Galerkin methods for time-domain and frequency-domain electromagnetic simulations. *Journal of Computational Physics*, 230(23):8360 – 8385, 2011.

[4] F. Assous, P. Degond, E. Heintze, P.A. Raviart, and J. Segre. On a Finite-Element Method for Solving the Three-Dimensional Maxwell Equations. *Journal of Computational Physics*, 109(2):222 – 237, 1993.

[5] K.E. Atkinson and W. Han. *Theoretical Numerical Analysis: A Functional Analysis Framework*. Texts in applied mathematics. Springer, 2001.

[6] BiblioBazaar and J.C. Maxwell. *A Treatise on Electricity and Magnetism Volume 1*. A historical reproduction. BiblioBazaar, 2010.

[7] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[8] A. N. Brooks and T. J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199 – 259, 1982.

[9] K. Buckley and P. Matousek. Non-invasive analysis of turbid samples using

deep Raman spectroscopy. *Analyst*, 136:3039–3050, 2011.

[10] D. Bucur and G. Buttazzo. *Variational Methods in Shape Optimization Problems*. Progress in Nonlinear Differential Equations and their Applications. Birkhäuser Boston, 2006.

[11] J.A. Burns. *Introduction to the Calculus of Variations and Control with Modern Applications*. Chapman & Hall/CRC Applied Mathematics & Nonlinear Science. Taylor & Francis, 2013.

[12] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16:1190–1208, 1995.

[13] J. Cagnol, J.P. Marmorat, and J.P. Zolésio. Shape Sensitivity Analysis in the Maxwell's Equations. In J. Cagnol, M.P. Polis, and J.P. Zolésio, editors, *Shape Optimization and Optimal Design*, Lecture notes in pure and applied mathematics. Taylor & Francis, 2001.

[14] J. Cea. Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 20(3):371–402, 1986.

[15] S. Chen and W. Chen. A new level-set based approach to shape and topology optimization under geometric uncertainty. *Structural and Multidisciplinary Optimization*, 44(1):1–18, 2011.

[16] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93. Springer Science & Business Media, 2012.

[17] M. Costabel and F. Le Louër. Shape Derivatives of Boundary Integral Operators in Electromagnetic Scattering. Part I: Shape Differentiability of Pseudo-homogeneous Boundary Integral Operators. *Integral Equations and Operator Theory*, 72(4):509–535, 2012.

[18] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[19] J. Crank and P. Nicolson. A pratical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Proc. Cam-*

bridge Philos. Soc., 43:50–67, 1947.

[20] J.P. Dakin and R.G.W. Brown. *Handbook of Optoelectronics (Two-Volume Set)*. Handbook of Optoelectronics. Taylor & Francis, 2010.

[21] M. C. Delfour. *Introduction to Optimization and Semidifferential Calculus*. SIAM, 2012.

[22] M.C. Delfour and J.P. Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*. Advances in design and control. SIAM, 2011.

[23] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Finite Element Methods for Flow Problems. John Wiley & Sons, 2003.

[24] R. D. Falgout and U. Meier Yang. hypre: a Library of High Performance Preconditioners. In *Preconditioners, Lecture Notes in Computer Science*, pages 632–641, 2002.

[25] P. E. Farrell, D. A. Ham, S. W. Funke, and M. E. Rognes. Automated derivation of the adjoint of high-level transient finite element programs. *CoRR*, abs/1204.5577, 2012.

[26] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[27] S. K. Godunov. Different Methods for Shock Waves. Ph.D. Dissertation, 1954.

[28] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.

[29] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition*. SIAM e-books. SIAM, 2008.

[30] M. Gröschel, G. Leugering, and W. Peukert. *Model Reduction, Structure-property Relations and Optimization Techniques for the Production of Nanoscale Particles*, pages 541–559. Springer Basel, Basel, 2012.

[31] M. D. Gunzburger. *Perspectives in Flow Control and Optimization*. 2003.

[32] M. Haderlein, D. Segets, M. Gröschel, L. Pflug, G. Leugering, and W. Peukert. Fimor: An efficient simulation for zno quantum dot ripening applied to the optimization of nanoparticle synthesis. *Chemical Engineering Journal*, 260:706 – 715, 2015.

[33] R. Hartmann. Numerical analysis of higher order discontinuous Galerkin finite element methods. *CFD - ADIGMA course on very high order discretization methods, Oct. 13–17, 2008, Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium*, 2008.

[34] J. Haslinger and R. A. E. Makinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation*. SIAM, Philadelphia, PA, USA, 2003.

[35] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer, 2008.

[36] M. Hintermüller, A. Laurain, and I. Yousept. Shape sensitivities for an inverse problem in magnetic induction tomography based on the eddy current model. *Inverse Problems*, 31(6):065006, 2015.

[37] M. Hintermüller and W. Ring. A Second Order Shape Optimization Approach for Image Segmentation. *SIAM Journal on Applied Mathematics*, 64(2):442–467, 2004.

[38] K. Ito, K. Kunisch, and G. H. Peichl. Variational approach to shape derivatives for a class of Bernoulli problems. *Journal of Mathematical Analysis and Applications*, 314(1):126 – 149, 2006.

[39] A. Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3(3):233–260, 1988.

[40] A. Jameson. Automatic design of transonic airfoils to reduce the shock induced pressure drag. In *Proceedings of the 31st Israel annual conference on aviation and aeronautics, Tel Aviv*, pages 5–17. Citeseer, 1990.

[41] A. Jameson. *Optimum aerodynamic design via boundary control*, volume 94. NASA Ames Research Center, Research Institute for Advanced Computer Science, 1994.

[42] A.L. Karchevsky. A frequency-domain analytical solution of Maxwell's equations for layered anisotropic media. *Russian Geology and Geophysics*, 48(8):689

– 695, 2007.

[43] S. Larsson, V. Thomée, and M. Krieger-Hauwede. *Partielle Differentialgleichungen und numerische Methoden*. SpringerLink: Springer e-Books. Springer-Verlag Berlin Heidelberg, 2005.

[44] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.

[45] A. Logg, K.-A. Mardal, and G. N. Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer, 2012.

[46] A. Logg, G. N. Wells, and J. Hake. *DOLFIN: a C++/Python Finite Element Library*, chapter 10. Springer, 2012.

[47] G.I. Marchuk. *Adjoint Equations and Analysis of Complex Systems*. Boston Studies in the Philosophy of Science. Kluwer Academic Publishers, 1995.

[48] T. Martin, P. Joshi, M. Bergou, and N. Carr. Efficient Non-linear Optimization via Multi-scale Gradient Filtering. In *Computer Graphics Forum*, volume 32, pages 89–100. Wiley Online Library, 2013.

[49] P. Monk. *Finite Element Methods for Maxwell's Equations*. Numerical Mathematics and Scientific Computation. Clarendon Press, 2003.

[50] J. C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35(3):315–341, 1980.

[51] J. C. Nédélec. A new family of mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 50(1):57–81, 1986.

[52] J. Neuberger. *Sobolev Gradients and Differential Equations*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2009.

[53] S. Nicaise, S. Stingelin, and F. Tröltzsch. Optimal control of magnetic fields in flow measurement. *Discrete and Continuous Dynamical Systems - Series S*, 8(3):579–605, 2015.

[54] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[55] S. J. Orfanidis. *Electromagnetic Waves and Antennas*. published online:

www.ece.rutgers.edu/ orfanidi/ewa, last revision:August 2010.

[56] S. Osher and R. P. Fedkiw. Level Set Methods: An Overview and Some Recent Results. *Journal of Computational Physics*, 169(2):463 – 502, 2001.

[57] S. Osher and J. A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[58] R. Potthast. Domain Derivatives in Electromagnetic Scattering. *Mathematical Methods in the Applied Sciences*, 19(15):1157–1175, 1996.

[59] R. J. Renka. A simple explanation of the sobolev gradient method, 2006.

[60] S. Schmidt. *Efficient Large Scale Aerodynamic Design Based on Shape Calculus*. PhD thesis, University Trier, 2010.

[61] S. Schmidt. A two stage CVT / eikonal convection mesh deformation approach for large nodal deformations. eprint arXiv:1411.7663, 2014.

[62] V. H. Schulz. A riemannian view on shape optimization. *Foundations of Computational Mathematics*, 14(3):483–501, 2014.

[63] V. H. Schulz and C. Schillings. Optimal aerodynamic design under shape uncertainties. *Tech Report 09-5*, 2009.

[64] V. H. Schulz and M. Siebenborn. Computational comparison of surface metrics for PDE constrained shape optimization. *eprint arXiv:1509.08601v2*, 2016.

[65] V. H. Schulz, M. Siebenborn, and K. Welker. A novel Steklov-Poincaré type metric for efficient PDE constrained optimization in shape spaces. *eprint arXiv:1506.02244*, 2015.

[66] V. H. Schulz, M. Siebenborn, and K. Welker. Structured inverse modeling in parabolic diffusion problems. *SIAM Journal on Control and Optimization*, 53(6):3319–3338, 2015.

[67] J.S. Shang and R. M. Fithen. A Comparative Study of Characteristic-Based Algorithms for the Maxwell Equations. *Journal of Computational Physics*, 125(2):378 – 394, 1996.

[68] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and

local minima. *Structural optimization*, 16(1):68–75, 1998.

[69] M.K. Smail, T. Hacib, L. Pichon, and F. Loete. Detection and Location of Defects in Wiring Networks Using Time-Domain Reflectometry and Neural Networks. *IEEE Transactions on Magnetics*, 47(5):1502–1505, May 2011.

[70] J. Sokolowski and J.P. Zolésio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer, 1992.

[71] M. Sonntag, S. Schmidt, and N. R. Gauger. Shape Derivatives for the Compressible Navier-Stokes Equations in Variational Form. *eprint arXiv/1312.5861*, 2013.

[72] J. L Steger and R.F Warming. Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods. *Journal of Computational Physics*, 40(2):263 – 293, 1981.

[73] Ingo Steinbach. Phase-field models in materials science. *Modelling and Simulation in Materials Science and Engineering*, 17(7):073001, 2009.

[74] A. Taflove and S.C. Hagness. *Computational Electrodynamics: The Finite-difference Time-domain Method*. Artech House antennas and propagation library. Artech House, 2005.

[75] J.L. Troutman. *Variational Calculus and Optimal Control: Optimization with Elementary Convexity*. Undergraduate Texts in Mathematics. Springer New York, 2012.

[76] F.W. Warner. *Foundations of Differentiable Manifolds and Lie Groups*. Graduate Texts in Mathematics. Springer, 1971.

[77] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagation*, pages 302–307, 1966.

[78] I. Yousept. Optimal control of Maxwell's equations with regularized state constraints. *Computational Optimization and Applications*, 52(2):559–581, 2011.

[79] S. Zaglmayr. *High Order Finite Element Methods for Electromagnetic Field Computation*. PhD thesis, Johannes Kepler Universität, Linz, 2006.

[80] J.P. Zolésio. Hidden Boundary Shape Derivative for the Solution to Maxwell

Equations and Non Cylindrical Wave Equations. In K. Kunisch, J. Sprekels, G. Leugering, and F. Tröltzsch, editors, *Optimal Control of Coupled Systems of Partial Differential Equations*, volume 158 of *International Series of Numerical Mathematics*, pages 319–345. Birkhäuser Basel, 2009.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Paderborn, den Maria Schütte