# Data-driven enhanced methods in terahertz tomography including a partially learned Landweber iteration

**Dissertation**

zur Erlangung des Grades des Doktors der Naturwissenschaften der Fakultät Mathematik und Informatik der Universität des Saarlandes

von

**Clemens Meiser**

Saarbrücken, 2024

UNIVERSITÄT DES SAARLANDES

| | |
|---|---|
| **Tag des Kolloquiums** | 29.11.2024 |
| Dekan: | Prof. Dr. Roland Speicher |

**Mitglieder des Prüfungsausschusses**:

| | |
|---|---|
| Vorsitzender: | Prof. Dr. Michael Bildhauer |
| 1. Berichterstatter: | Prof. Dr. Thomas Schuster |
| 2. Berichterstatter: | Prof. Dr. Anne Wald |
| Protokollführer | Dr. Dimitri Rothermel |

# Acknowledgements

I extend my sincerest gratitude to all those who have supported and motivated me throughout this challenging journey of completing my doctoral thesis.

First of all, I would like to express my deepest thanks to my supervisor Professor Thomas Schuster. I am grateful for his guidance, patience and belief in my abilities. I appreciate his knowledge, expertise and commitment to academic excellence. His lectures at Saarland University motivated me to continue studying mathematics after my first state exam in teaching. It is a pleasure to finish my studies in his working group. Also, I would like to mention that the quote on the first page of the introduction is taken from a presentation by Professor Schuster that awakened my interest in inverse problems already as a student.

Secondly, I thank Professor Anne Wald for taking over the co-referee of this thesis. Prof. Wald acted as my contact person for scientific questions when I started in the working group of Prof. Schuster.

I also want to express my gratitude to my esteemed colleagues Dr. Dimitri and Dr. Rebecca Rothermel, Lukas Vierus, Alice Oberacker, Petra Schuster-Gentes and Claudia Stoffer. The collaborations, discussions and constructive feedback have greatly enriched my research.

A special appreciation goes to my beloved wife Catrin. Her support, patience and understanding have given me strength during this challenging journey. Her belief in me and constant encouragement have been a constant source of inspiration. I am truly grateful to have her unwavering love and support. Our son Samuel has given strength to me to finish my thesis. I dedicate this work to both of you.

Lastly, I would like to thank my parents Christoph and Bettina. Their support and guidance have shaped the person I am today. I owe them an immeasurable debt of gratitude for instilling in me the values of perseverance, determination and lifelong learning.

To all those who have been a part of my journey, whether mentioned or not, I am truly grateful for your contributions. Each of you has played a significant role in shaping my academic and personal growth.

# Abstract

We consider the inverse problem of terahertz tomography which is an imaging technique for monitoring plastics and ceramics. The underlying mathematical problem is associated with Maxwell's equations and simplifications thereof. Starting with the non-linear eikonal equation as a physical model, we address this problem by the Landweber iteration respecting the theory of inverse problems. The eikonal equation results as a high-frequency approximation of the Helmholtz equation and more generally, of the wave equation. The primary task is to deduce the refractive index, denoted as $n(\mathbf{x})$, from time-of-flight measurements.

In a second step, we introduce neural networks in the Landweber iteration for the reconstruction of the refractive index $n(\mathbf{x})$. Applying Landweber's method, we have to solve a non-linear partial differential equation for the forward operator for each initial condition in every step but also need to compute the adjoint operator of the linearization. To reduce the computing time in the reconstruction process, we substitute the forward operator $F$ with a convolutional neural network. Using synthetic training data which consists of the refractive index as the input and simulated measure data as the output, we accelerate significantly the evaluation of the forward operator, i.e. the solution of the eikonal equation, compared to standard techniques such as marching schemes. Furthermore, we save energy in the learning process of the network by generating a sparse forward operator. We add an $\ell_1$-regularization term to the cost functional of the convolutional neural network. Then, we compare the standard Landweber method with the partially learned and the sparse alternative presenting numerical results.

Additionally, we develop and implement a data-driven anomaly detection algorithm for inline monitoring with a particular focus on plastics. We propose a density-based approach to detect anomalies in the radiation's measured data automatically. We illustrate numerical results to support our findings. The real measured values are provided by the German Plastics Center in Würzburg.

# Zusammenfassung

Wir betrachten das inverse Problem der Terahertz-Tomographie, einer bildgebenden Technik zur Überwachung von Kunststoffen und Keramiken. Das zugrundeliegende mathematische Problem ist mit den Maxwell Gleichungen und deren Vereinfachungen verbunden. Ausgehend von der nichtlinearen Eikonalgleichung als physikalisches Modell gehen wir diese Herausforderung mit Hilfe der Landweber-Iteration unter Berücksichtigung der Theorie der inversen Probleme an. Die Eikonalgleichung ist eine hochfrequente Annäherung an die Helmholtz Gleichung und, allgemeiner, an die Wellengleichung. Die primäre Aufgabe besteht darin, den Brechungsindex, bezeichnet als $n(\mathbf{x})$, aus Laufzeitmessungen abzuleiten.

In einem zweiten Schritt führen wir neuronale Netze in der Landweber-Iteration zur Rekonstruktion des Brechungsindex $n(\mathbf{x})$ ein. Bei der Anwendung der Landweber-Methode müssen wir die nichtlineare partielle Differentialgleichung für den Vorwärtsoperator in jedem Schritt lösen, aber auch den adjungierten Operator der Linearisierung berechnen. Um die Rechenzeit im Rekonstruktionsprozess zu reduzieren, ersetzen wir den Vorwärtsoperator $F$ durch ein Convolutional Neural Network. Unter Verwendung synthetischer Trainingsdaten, die aus dem Brechungsindex als Input und simulierten Messdaten als Output bestehen, beschleunigen wir die Auswertung des Vorwärtsoperators, d.h. die Lösung der Eikonalgleichung, im Vergleich zu Standardtechniken wie Marschschemata erheblich. Außerdem sparen wir im Lernprozess unseres Netzes Energie, indem wir einen sparsen Vorwärtsoperator erzeugen. Wir fügen dem Kostenfunktional des Convolutional Neural Network einen $\ell_1$-Regularisierungsterm hinzu. Anschließend vergleichen wir die Standard-Landweber-Methode mit der teilweise gelernten und der sparsen Alternative und stellen numerische Ergebnisse vor.

Desweiteren entwickeln und implementieren wir einen datengesteuerten Algorithmus zur Erkennung von Anomalien für die Inline-Überwachung mit besonderem Schwerpunkt auf Kunststoffen. Wir schlagen einen dichtebasierten Ansatz vor, um automatisch Anomalien in den Messdaten der Strahlung zu erkennen. Wir liefern numerische Ergebnisse, um unsere Erkenntnisse zu untermauern. Die realen Messwerte werden vom Deutschen Kunststoffzentrum in Würzburg zur Verfügung gestellt.

# Contents

**Abstract** **III**

**Zusammenfassung** **V**

**List of Figures** **IX**

**List of Tables** **X**

**Introduction** **1**

**1 Motivation** **7**

**2 Physical background** **13**
2.1 Maxwell's equations . . . . . 13
2.2 Wave equation . . . . . 15
2.3 Helmholtz equation . . . . . 17
2.4 Eikonal equation . . . . . 17

**3 Inverse problem of terahertz tomography** **19**
3.1 The forward operator based on the eikonal equation . . . . . 19
3.2 Linearization of the forward operator . . . . . 24
3.3 Adjoint linearized problem . . . . . 26
3.4 Solving the inverse problem using Landweber iteration . . . . . 29
3.4.1 Numerical solution of the forward operator using the Fast Marching Method . . . . . 31
3.4.2 Numerical solution of the adjoint linearized problem . . . . . 35
3.4.3 Numerical reconstruction with the Landweber iteration . . . . 38

**4 Learned Landweber iteration for terahertz tomography** **46**
4.1 Introduction to convolutional neural networks and sparse networks . . 47
4.2 Learned solution of the eikonal equation . . . . . 56
4.3 Numerical evaluation of the learned Landweber iteration . . . . . 64
4.4 Sparsity-based learning of the eikonal equation . . . . . 68
4.5 Numerical evaluation of the sparsity-based learned Landweber iteration 72

**5 Learned anomaly detection with terahertz radiation in inline monitoring** **75**
5.1 Classification algorithm . . . . . 76
5.2 Terahertz measuring system and data set . . . . . 79

5.3 Numerical results for measured data sets . . . . . 82
5.4 Numerical results based on partly simulated data . . . . . 84

**6 Conclusion and outlook** **89**

**A Notations** **93**

**B Some supplementary mathematical theory** **95**

**C Source code** **98**

# List of Figures


1 Structure of an inverse problem . . . . . 1

1.1 Electromagnetic spectrum . . . . . 7
1.2 Schematic two-dimensional representation of a THz tomograph . . . . 9
1.3 Time and phase shift of a THz signal . . . . . 10

3.1 Visualization of the FMM . . . . . 35
3.2 Outer normal vector of a square . . . . . 37
3.3 Reconstruction $v^{\delta}_{1,k^*}$ of the ground truth velocity $v_1$ with the Landweber iteration showing the error plot $v_1 - v^{\delta}_{1,k^*}$ . . . . . 41
3.4 Relative error and the norm of the residual during the Landweber iteration for the reconstruction of $v_1$ . . . . . 41
3.5 Reconstruction $v^{\delta}_{2,k^*}$ of the ground truth velocity $v_2$ with the Landweber iteration showing the error plot $v_2 - v^{\delta}_{2,k^*}$ . . . . . 43
3.6 Reconstruction $v^{\delta}_{3,k^*}$ of the ground truth velocity $v_3$ with the Landweber iteration showing the error plot $v_3 - v^{\delta}_{3,k^*}$ . . . . . 44
3.7 Reconstruction $v^{\delta}_{4,k^*}$ of the ground truth velocity $v_4$ with the Landweber iteration showing the error plot $v_4 - v^{\delta}_{4,k^*}$ . . . . . 45

4.1 Example of a multilayer perceptron including $L$ hidden layers. . . . . 47
4.2 Activation functions . . . . . 49
4.3 Visual example of Algorithm 4.10 . . . . . 52
4.4 Example of a CNN . . . . . 54
4.5 Idea of sparsity in the context of neural networks . . . . . 55
4.6 Surface plot of the refractive index of some training input data $X^{(i)}$ . 58
4.7 Contour plot of two angular positions of the output training data $Y^{(1)}$ 59
4.8 Structure of the implemented CNN . . . . . 60
4.9 Comparison of the performance of the FMM and the CNN . . . . . 63
4.10 Reconstruction $v^{\delta}_{1,k^*}$ of the material-dependent velocity $v_1$ with the learned Landweber iteration showing the error plot $v_1 - v^{\delta}_{1,k^*}$ . . . . . 65
4.11 Development of the relative error for the normal and the learned Landweber iteration for the reconstruction of phantom $v_1$ . . . . . 65
4.12 Reconstruction $v^{\delta}_{3,k^*}$ and $v^{\delta}_{5,k^*}$ of the material-dependent velocity $v_3$ and $v_5$ with the learned Landweber iteration . . . . . 66
4.13 Reconstruction $v^{\delta}_{5,k^*}$ of the velocity $v_5$ after $k^* = 25$ iterations of the learned Landweber method using an adapted training data set . . . . 67
4.14 Reconstruction $v^{\delta}_{5,k^*}$ of the velocity $v_5$ after $k^* = 25$ iterations of the normal Landweber method . . . . . 67

4.15 Reconstruction $v^{\delta}_{5,k^*}$ of the velocity $v_5$ after $k^* = 13000$ iterations of the normal Landweber method . . . . . 68
4.16 Visualization of the network performance . . . . . 70
4.17 Percentage of the weights equal to zero per layer . . . . . 73
4.18 Reconstruction $v^{\delta}_{1,k^*}$ and $v^{\delta}_{5,k^*}$ of the material-dependent velocity $v_1$ and $v_5$ with the sparsity-based learned Landweber method using $\tilde{\theta_3}^*$ . 74

5.1 THz tomography system at Plastic Center (SKZ), Würzburg . . . . . 80
5.2 Schematic THZ tomograph . . . . . 80
5.3 Data points for the path difference and the amplitude ratio of a solid pipe without defects in transmission . . . . . 81
5.4 Distribution of the data points registered at receiver $R_2$ . . . . . 82
5.5 Investigation of an unknown pipe using the classification algorithm . 84
5.6 Simulation of the travel time of the THz radiation by solving the eikonal equation and approaching the Gaussian profile . . . . . 85
5.7 Comparison of the simulated and the measured data for a solid pipe with a diameter of 10 cm and a refractive index of $n = 1.53$ . . . . . 86
5.8 One-dimensional anomaly detection for the unknown pipe using a hybrid data set . . . . . 87
5.9 One-dimensional anomaly detection set for the unknown pipe using only measured data . . . . . 88

# List of Tables


1.1 The $CO_2$ footprint for training a deep neural network compared to the energy consumption of a car during its lifetime and the average $CO_2$ emission produced by a human being within one year . . . . . 11

3.1 Reconstruction results and Landweber parameters for the function $v_1$ 42
3.2 Reconstruction results and Landweber parameters for the functions $v_2$, $v_3$ and $v_4$ . . . . . 42

4.1 Value of the cost function for the training and validation set . . . . . 62
4.2 Reconstruction parameters and results for the functions $v_1$, $v_3$ and $v_5$ with the learned Landweber iteration . . . . . 66
4.3 Performance of a dense network, a regularized network, a pruned network and a frozen network applied to the MNIST database . . . . 69
4.4 Performance of two dense networks and a sparse one applied to the eikonal equation using a small data set . . . . . 70
4.5 Performance of the sparse network applied to the eikonal equation using the data set of Section 4.2 for different threshold parameters $\zeta$ and regularization parameters $\alpha$ . . . . . 72
4.6 Reconstruction results for the functions $v_1$ and $v_5$ with the sparsity-based learned Landweber iteration using $\tilde{\theta}_1^*$, $\tilde{\theta}_2^*$, $\tilde{\theta}_3^*$ . . . . . 73

# Introduction

If we multiply two natural numbers, we will obtain a unique result. Conversely, the representation of a natural number by two factors does not always consist of a unique solution and one has to fall back on the principle of prime numbers to obtain uniqueness. If it rains the whole day, the street will be wet. But, if we see a partially wet street, we can not guarantee that it has just rained. There could be other reasons for the wetness, and therefore, it is not possible to say for sure that the rain has just occurred. Maybe, it has snowed or hailed or someone has just washed his or her car. These presented examples, familiar to us from childhood, introduce us to the field of direct and inverse problems which are closely interrelated. Such problems are based on a causal connection between cause and effect that often can be described by a mathematical or physical model. When we talk about a direct problem, we know the cause and we can calculate the effect by given rules or depending on experience. Considering an inverse problem, on the other hand, the effect and the model are known, but the cause is unknown. The relation between cause and effect should be reversed, see Figure 1.

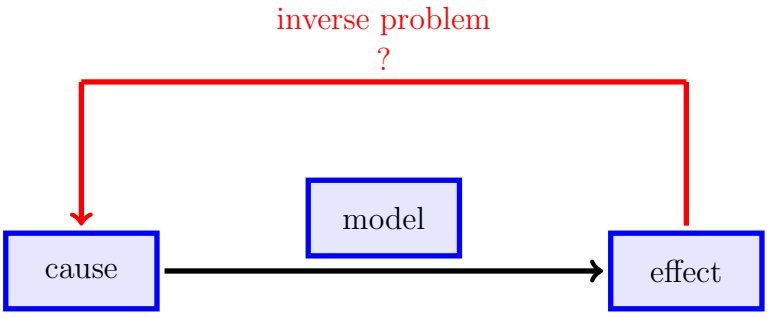



Figure 1: Structure of an inverse problem

To explain the idea of an inverse problem detached from mathematical concepts, one can quote Sir Arthur Conan Doyle who created the character of Sherlock Holmes. In his first detective novel '*A Study in Scarlet*' he wrote:

> Most people, if you describe a train of events to them, will tell you what the result would be. They can put those events together in their minds and argue with them that something will come to pass. There are few people, however, who, if you told them a result, would be able to evolve from their inner consciousness what the steps were which led up to a result. This power is what I mean when I talk of reasoning backward, or analytically ([17], p. 156).

The concept of an inverse problem is not just focused on mathematics but is also represented in the natural thinking of a person: Mr. Doyle describes in the first

lines the idea of direct thinking. Giving a starting point and some events, one can deduce the consequences and solve the direct problem. Many people are blessed with this gift. The more difficult way of thinking is the inverse one. Only a few people have the skill of drawing conclusions from results, i.e. '*reasoning backward*' like Sherlock Holmes and solving the inverse problem. They can reconstruct the causes. It is similar in mathematics: While the direct problem can often be described by functions or differential equations underlying physical laws solving the inverse problem is the more difficult task.

In the history of mathematics, inverse formulations have been established for a long time and extend from linear regression problems to highly complex applications related to scientific problems. However, the theory of inverse problems originates from the second half of the 20th century, see [35]. One of the most famous inverse problems of our society in the context of mathematics and physics can be found in medicine: the X-ray tomography. In 1917 Johann Radon deduced the mathematical background that could be involved to create the first computed tomography scanner in the 60s and 70s. The aim is to determine the density function of an object. Depending on their absorption coefficient, different materials can absorb the X-rays to different degrees. A detector is used to measure the decrease in the intensity of the penetrating radiation through the object. It follows the inverse problem of reconstructing the density distribution of an object under investigation, the cause, by the measured reduction of the intensity, the effect. The direct problem can be modeled by Beer-Lambert's law, see [15]. Today's studies on this subject are concerned with nano and limited-angle computed tomography. Further inverse problems based on physical phenomena can be found in vector or tensor tomography [92], magnetic particle imaging [50] or in heat conduction problems in industry [79], just to name a few.

Mathematically, an **inverse problem** is described by an operator equation of the form

$$F(x) = y. \tag{1}$$

The **forward operator** $F : \mathcal{D}(F) \to Y$ models the direct problem and maps an $x$ out of the source space $X$ to a $y$ out of the data space $Y$. The notation $\mathcal{D}(F)$ represents the domain of the operator $F$. The corresponding inverse problem deals with the case that $y \in Y$ is known and we have to reconstruct the respective $x \in X$. The topological spaces $X$ and $Y$ are mostly defined as Banach or Hilbert spaces. Depending on their mathematical modeling, inverse problems can be subdivided into two classes: linear and non-linear inverse problems. An inverse problem will be denoted as linear if the forward operator is linear. Then we write the operator equation to clarify as

$$Fx = y. \tag{2}$$

Accordingly, **non-linear inverse problems** possesses a non-linear operator $F$. In this work, we deal with a non-linear inverse problem. The mentioned X-ray tomography illustrates an example of a linear inverse problem.

In real applications, the available data $y$ include noise given by errors in the model, errors of observation or errors of measurement. In addition, we have to handle

rounding and discretization errors when moving on to finite-dimensional spaces for the implementation. We suppose that the noisy data $y^\delta$ fulfills the inequality

$$||y^\delta - y||_Y \leq \delta \tag{3}$$

where $||\cdot||_Y$ denotes the norm of the vector space $Y$. Due to the ill-posedness of such problems a direct inversion of the forward operator applied to noisy data $y^\delta$ often results in useless solutions $x^\delta$. For linear forward operators and any topological spaces Hadamard [27] defined that the problem $Fx = y$ is **well-posed** if

- the equation (2) has a solution for all $y \in Y$,
- the solution is unique,
- the inverse operator $A^{-1}$ is continuous, such that small disturbances in $y$ cause small disturbances in $x$.

If one of these conditions is violated, we call the problem **ill-posed**. Referring to Hilbert spaces, we can neglect the first two conditions by introducing the concept of the generalized inverse. According to Nashed, the definition of ill-posedness is reduced to stability, see [78]. While ill-posedness is a global property for linear inverse problems, it is locally defined in the non-linear setting. We call a non-linear operator equation **locally ill-posed** in $x^+ \in \mathcal{D}(F)$ if there is a sequence $\{x_k\}_{k\in\mathbb{N}}$ for every ball centered at $x^+$ that does not converge to $x^+$, but whose sequence of images $\{F(x_k)\}_{k\in\mathbb{N}}$ converges to $F(x^+)$.

The ill-posedness of an inverse problem is determined by the topology of the spaces $X$ and $Y$. This can be illustrated by the problem of differentiability, see [35, 78], where we have a well-posed problem for continuous functions and an ill-posed problem for bounded functions. The ill-posedness of inverse problems mostly results from the discontinuity of the inverse operator. Small perturbations in the observation increase strongly by using the inverse of the forward operator directly. The solution is useless. A way out is provided by so-called **regularization methods** that stabilize the inverting process. The inverse mapping is approximated by continuous operators and an optimum between stability and approximation is found. The total error depends on the regularization parameter. In the mentioned example of differentiability, one can overcome the ill-posedness by considering a regularization by the differential quotient. Until the end of the 20th century, a wide range of regularization methods for linear inverse problems in Banach and Hilbert spaces were developed, thanks among others to the work of Tikhonov, see [5]. The investigation of such problems is considered complete. An overview can be found in the book of Engl, Hanke and Neubauer [19]. Many of those methods, especially iterative ones, could be transferred to non-linear inverse problems. A systematic investigation of regularization methods for non-linear inverse problems started in the 80s. For this investigation, it must be taken into account that for non-linear inverse problems there is information about the non-linear operator only in a neighborhood of an element $x \in X$ and not on the whole domain $\mathcal{D}(F)$. As a consequence, for example, the initial value should be chosen close enough to the solution by giving an a priori information. Famous methods for the regularization of non-linear inverse problems are the iterative Tikhonov method [71], the regularized Gauss-Newton method [49] and the **Landweber iteration** [48].

The latter is a stable method but extremely time-consuming and will be focused on in our investigation: We deal with a non-linear inverse problem that is solved in the first part of this thesis by the iterative Landweber method. Respecting that in real applications, i.e. in industry, not just a stable and good approximation of the solution is necessary but also the reconstruction time plays an important role, we want to accelerate the normal Landweber iteration in a second step. Especially given the concept '*Industry 4.0*' where people, machines and products are directly networked with each other and the production and control of materials are desired inline, a fast reconstruction process is indispensable. For this purpose, we establish a connection to the area of artificial intelligence. By learning the solution of the parameter-to-solution operator of the inverse problem, we reduce the numerical effort of Landweber's method. Instead of solving a partial differential equation with a time-consuming numerical method, we involve a deep neural network to speed up the regularization method.

Techniques from the field of machine learning are gaining significant attention nowadays. The huge popularity is driven by the increasing computer power and the focus on big data in our society. Machine learning represents a subfield of artificial intelligence. The latter has been a scientific discipline since the 1950s. The name artificial intelligence originates from the '*Dartmouth Conference*' in 1956, whereas its vision was already anchored in people's minds beforehand, for example, in the literature '*I robot*' of Isaac Asimov [4]. Often the work of Alan Turing during the Second World War is cited as the historical starting point for the scientific discipline of artificial intelligence [28]. Meanwhile, machines beat us in games like '*Chess*' and '*Go*' and machine learning software has become a standard for many companies. Technologies such as '*ChatGPT*', a large language model developed by the company '*OpenAI*', have revolutionized the human mindset [6, 77]. Machine learning includes many algorithms and techniques like regression, classification or deep learning [25] that usually can be subdivided into supervised and unsupervised learning methods. In our research, we focus on the supervised context where algorithms learn their parameters from pairs of labeled input and output data. Generally speaking, a machine learning algorithm is trained by observing large data sets in the training process in order to make predictions about unobserved data. Our machine learning task involves so-called **convolutional neural networks (CNNs)** to approximate our forward operator. CNNs are mainly used for deep learning to recognize structures in the input matrix by convolution. These networks decrease the number of connections within a layer in comparison to fully-connected networks. Deep learning is a subsection of machine learning and provides the basis for some tasks like image and speech recognition and self-driving cars. Deep neural networks are neural networks with more than one layer.

To reduce the complexity of regularization methods and their computational time, the current investigation of inverse problems is taking into account more and more techniques from the field of machine learning. Furthermore, some mathematical concepts are developed to analyze the processes of deep learning, see [1]. In a publication, Maas et al. show that the learning of an inverse problem directly, i.e. the learning of the inverse mapping, already fails for simple investigations [63]. Further research on inverse problems in connection with neural networks can be found, for example, in publications of Haltmeier [29] or Kaltenbacher [47]. An important theorem for

neural networks, the universal approximation theorem by Cybenko [14], see Appendix B.4, states that a one-layer fully-connected network can approximate a continuous function on a compact subset $I \subseteq \mathbb{R}^n$ by using a sigmoidal activation function if the number of hidden units is large enough. In general, theorems on universal approximation set boundaries on the theoretical learning capacity of neural networks. The result of Cybenko was transferred to arbitrary $L^p$-spaces by Hornik [36] and can also be extended to the unbounded rectified linear unit as the activation function, see Appendix B.5. There exist further strong results on universal approximation, for example, for any function [59]. Recently, the fundamental approximation theorem has been generalized to infinite-dimensional function spaces, see [26]. Another work extends the result to CNNs, see [98].

In this work, we aim to combine the theory of inverse problems with the practical use of deep neural networks by accelerating the iterative Landweber method using a CNN as the forward operator. We apply the concept to the complex inverse problem of **terahertz (THz) tomography**. THz tomography is a relatively new technique and is gaining importance in science and industry. Because of the characteristics of the electromagnetic radiation within the THz range, THz tomography is perfectly suited as a technique to investigate plastics and ceramics. The object under investigation is penetrated by electromagnetic radiation from different positions. Receivers around the object collect information about the radiation. We want to reconstruct the refraction index $n(\mathbf{x})$, $\mathbf{x} \in \Omega \subseteq \mathbb{R}^2$, by measurements of the electromagnetic field. The application of THz inspection has expanded rapidly because the receiver and emitter of THz radiation have become cheaper, and consequently, has the chance to compete with other tomography techniques like X-ray, ultrasonic or microwaves. Starting with the non-linear eikonal equation as a physical model, we solve the inverse problem in the first step by the Landweber iteration, and in the second step, by a partially learned Landweber method. The eikonal equation can be seen as a high-frequency approximation of the Helmholtz equation, and more generally, of the wave equation. Using the Landweber iteration, we have to solve the eikonal equation as the parameter-to-solution operator in every step and for every angular position. We reduce the computing time by learning the forward operator.

In the first chapter, we justify our research and the given mathematical model and present the motivation for our work. We explain why we concentrate on the inverse problem of THz tomography, and above all, why we are interested in accelerating Landweber's method by machine learning techniques. In Chapter 2, we describe the physical basics of our research. Starting with the local Maxwell's equations for a special setting, we derive the eikonal equation that is used in the next chapters to model the propagation of THz radiation. We focus on the inverse problem of THz tomography in Chapter 3. We use the eikonal equation to describe our parameter-to-solution operator that is solved by the **Fast Marching Method (FMM)**. The forward operator is linearized, and afterwards, we calculate its adjoint. Both, the FMM and the solution of the adjoint linearized problem are implemented within the Landweber algorithm to solve the inverse problem. We aim to determine the refractive index $n(\mathbf{x})$ of an object in order to find defects like air inclusions. Numerical reconstructions are given at the end of the chapter. In Chapter 4, we accelerate Landweber's method by combining it with machine learning techniques. We train a convolutional network

that solves the eikonal equation and substitute the FMM. We result in a partially learned Landweber iteration. To reduce the amount of weights in our network we make the network sparse by an $\ell_1$-regularization of the cost function in the second part of Chapter 4. Numerical results for the partially learned Landweber method and the sparse one are presented and compared with the reconstructions from Chapter 3. The last chapter is based on a joint project with the **German Plastic Center (SKZ)** in Würzburg. This chapter can be seen as a kind of digression and the starting point of the underlying investigations. We deal with inline monitoring for plastics and ceramics and detect defects with a learned anomaly detection algorithm. In contrast to Chapter 3 and 4, we have access to real data for this part of our work. Finally, we conclude with a short overview and discussion. We give an outlook for future research.

Our investigation is a continuation of the research of Prof. Dr. Anne Wald and Dr. Jens Tepe who already worked in the field of THz tomography in the numeric group of Prof. Dr. Thomas Schuster at Saarland University.

# 1 Motivation

The term *'tomography'* covers imaging procedures for the examination of bodies. Tomographic methods are used in a wide variety of applications, particularly, in medical diagnostics and for the analysis of materials and components. The first experiments on imaging with THz radiation were published in 1976 by Hartwick et al. [32]. These investigations were followed by a period of twenty years during which the research on THz radiation stagnated. The technology for generating the radiation was too complicated and laborious. Neither suitable transmitters for the generation of THz radiation nor suitable receivers for the detection of the signal were available [97]. The concerned frequency range, which was hard to handle, is called the *'THz gap'*. In the last two to three decades, this gap was closed thanks to new developments, for instance, improvements in femtosecond optoelectronics [69]. Nowadays, there exist two prominent techniques to generate the radiation: the continuous and the pulsed THz systems. For more information about the THz systems we refer to Nüßler et al. [75] or Zhong [97].

The publication of Hu et al. [37] marks the starting point of the ongoing research. THz technologies have rapidly evolved. The senders and detectors have improved and become more efficient and economical for industry. While Hartwick investigated in his paper a wavelength between 0.3 mm to 1 mm, THz radiation is defined as a part of the electromagnetic spectrum with a wavelength of 0.03 mm to 3 mm. Consequently, it is placed in the electromagnetic spectrum between microwaves and infrared radiation with a corresponding frequency of 100 GHz to 10 THz, see Figure 1.1. Because of its position in the electromagnetic spectrum, THz radiation includes wave as well as ray character and describes the transition between electromagnetism and optics.

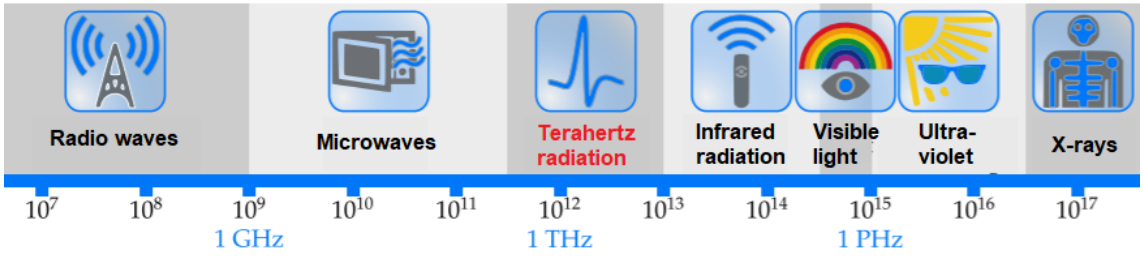



Figure 1.1: Electromagnetic spectrum (modified according to [68])

Driven by the development of technology, applications of THz radiation are widespread in industry. The radiation is utilized in body scanners for security purposes [91], car painting control and for the pharmaceutical industry [97]. In particular, the

radiation receives increasing attention in the field of non-destructive testing. The latter refers to techniques that provide information about the interior of an object without destroying it. Especially for plastics, ceramics, glass, wood or clothes, non-destructive testing with THz radiation is perfectly suited because the radiation can penetrate non-conductive materials. Liquids absorb the radiation while metal reflects it completely. In contrast to X-rays, THz radiation is non-ionizing and therefore not dangerous to health. It is applied as a non-contact technique that does not require a medium to couple with. Furthermore, the radiation achieves a better resolution compared to microwaves because of its shorter wavelength [74].

A variety of tomography techniques have been explored over the past few years to reconstruct images using THz radiation. Some techniques have been adapted and adopted from competing technologies like X-ray or ultrasonic imaging [41, 99]. In the last few years, Prof. Dr. Thomas Schuster's research group has been working intensively on new reconstruction methods in the field of THz tomography. Tepe et al. [88] investigated a modified algebraic reconstruction technique, the modified ART, to solve the inverse problem while Wald et al. [93] performed an inverse parameter identification for the Helmholtz equation taking into account the Gaussian profile of the THz beam. However, both methods have their limits: For the modified ART, preliminary information about the outer and inner boundaries of the investigated object is required that is not given in reality and has to be detected. The Helmholtz equation provides good results in a low-frequency range but fails for high frequencies. Furthermore, the method of Wald is very time-consuming. In addition, both reconstruction techniques were combined to generate a predictor-corrector-method [64]. The idea is to detect the boundaries with a few iterations of Wald's parameter identification problem and then use the modified ART to reconstruct. All methods mentioned aim to determine the complex refractive index $\tilde{n}(\mathbf{x})$ which depends on the position $\mathbf{x} \in \mathbb{R}^2$ (respectively, $\mathbf{x} \in \mathbb{R}^3$). If we reconstruct the complex refractive index, we can identify density differences, moisture contents and defects such as air inclusions, see [33]. To define the complex refractive index, we introduce the refractive index $n(\mathbf{x})$ and the extinction coefficient $\kappa(\mathbf{x})$.

**Definition 1.1** (**Refractive index**, [15], p. 220)**.**
*Let $\mathbf{x} \in \Omega \subseteq \mathbb{R}^2$. The refractive index $n : \Omega \to [1, +\infty)$ describes the ratio of the propagation speed $c_0 = 299792458$ m s$^{-1}$ in vacuum and the propagation speed $v : \Omega \to (0, c_0]$ inside a medium. It holds*

$$n(\mathbf{x}) := \frac{c_0}{v(\mathbf{x})}. \tag{1.1}$$

**Definition 1.2** (**Extinction coefficient**, [15], p. 223)**.**
*Let $\mathbf{x} \in \Omega \subseteq \mathbb{R}^2$. The extinction coefficient $\kappa : \Omega \to [0, +\infty)$ is proportional to the absorption coefficient $\alpha(\mathbf{x})$ and is defined via*

$$\kappa(\mathbf{x}) := \alpha(\mathbf{x}) \frac{c_0}{4\pi f}$$

*where $f$ is the frequency of the radiation. The absorption coefficient depends on the measured intensity $I_0$ without absorption, the measured intensity $I$ with absorption and the thickness $z$ of the absorbing object via Beer-Lambert's law and is given by*

$$I = I_0 e^{-\alpha(\mathbf{x})z}.$$

It follows the definition of the complex refractive index as a composition of the refractive index and the extinction coefficient:

**Definition 1.3** (**Complex refractive index**, [15], p. 224)**.**
*Let* $\mathbf{x} \in \Omega \subseteq \mathbb{R}^2$*. The complex refractive index* $\tilde{n} : \Omega \mapsto \mathbb{C}$ *is given by*

$$\tilde{n}(\mathbf{x}) := n(\mathbf{x}) + i \cdot \kappa(\mathbf{x})$$

*where* $n(\mathbf{x})$ *is the refractive index and* $\kappa(\mathbf{x})$ *the absorption coefficient at position* $\mathbf{x}$*.*

In THz tomography, the object under investigation is illuminated by electromagnetic radiation in the corresponding frequency range. As illustrated in Figure 1.2, there is one sender, $E_1$, emitting the radiation. Several receivers around the object, in the case shown $R_1$ to $R_{18}$, collect information about the radiation. This information is used to reconstruct the complex refractive index of the considered domain $\Omega$. After one measuring process, the measuring system is shifted such that the object is investigated from different positions.

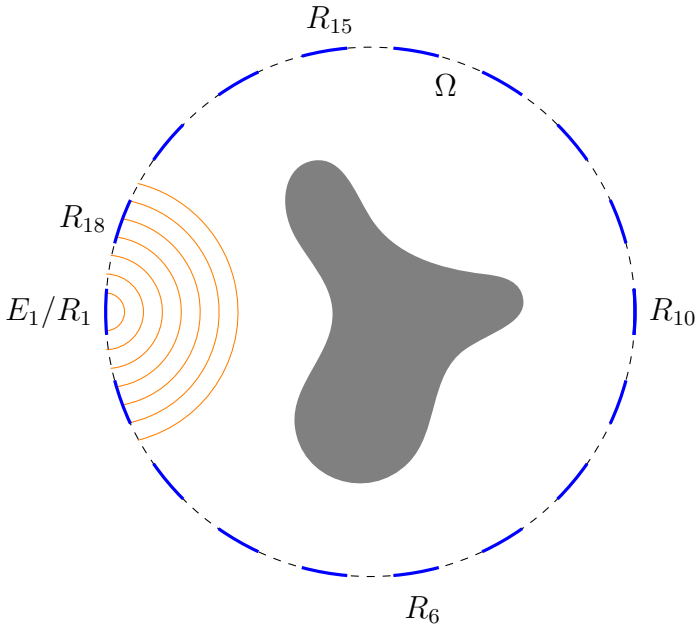



Figure 1.2: Schematic two-dimensional representation of a THz tomograph

In Figure 1.3, we illustrate an example of the measured THz signal: The figure depicts a reference signal of the THz pulse drawn in blue and the measured signal drawn in red. We notice a vertical reduction of the amplitude because the object absorbs some radiation. An increase in the absorption coefficient of the object, i.e. the extinction coefficient, results in a greater vertical shift of the signal. Further, we notice a horizontal shift of the signal. Penetrating through an object, the travel time of the radiation is increased due to the lower propagation speed inside the object. An increase in the refractive index inside the object, i.e. a reduction of the propagation speed, results in a greater horizontal shift of the signal. The electromagnetic wave reaches the receiver later. The complex refractive index thus exhibits a direct relationship with the propagation delay of the signal given by the real part and with the attenuation of the intensity given by the imaginary part.

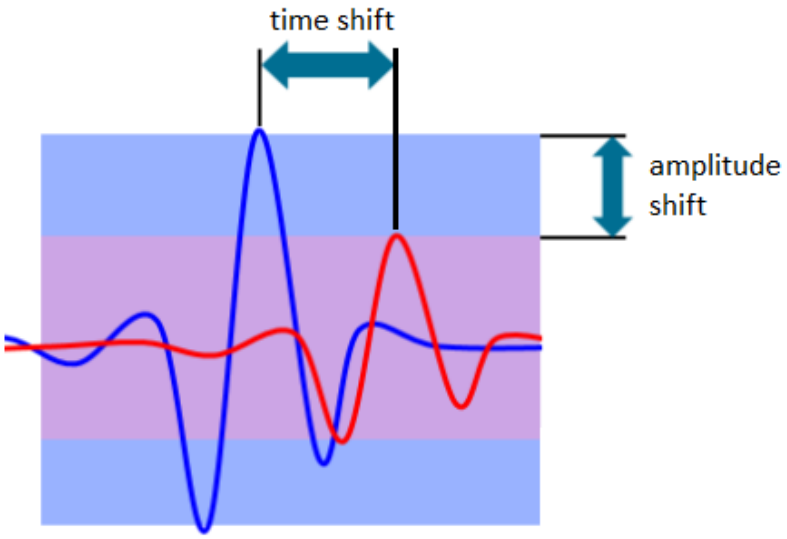



Figure 1.3: Time and phase shift of the amplitude of a measured THz signal in red compared to a reference signal in blue (modified according to [62])

In general, the inverse problem of THz tomography can be formulated as the following problem:

**Problem 1.4** (**Inverse problem of the two-dimensional THz tomography**). *Determine the complex refractive index* $\tilde{n}(\mathbf{x})$ *for a domain* $\Omega \subseteq \mathbb{R}^2$ *by measurements of the travel time* $T_i$ *and the transmittance* $\tau_i = \frac{I_i}{I_{i_0}}$ *of an electromagnetic wave with* $i = 1, ..., N$ *receivers.*

Motivated by the fact that Wald et al. failed for high frequencies, this thesis deals with a new approach to the inverse problem of THz tomography. We use a high-frequency approximation of the Helmholtz equation which results in the eikonal equation and reduce the problem to reconstruct only the refractive index $n(\mathbf{x})$ of an object. The eikonal equation models the physical relationship between the refractive index for a referred domain and the travel time of an electromagnetic wave for high frequencies. Consequently, we focus on the time-of-flight measurements of the signal and neglect the measurement information about the absorption and reflection. Further, we concentrate on a two-dimensional reconstruction. It results the reduced inverse problem of the two-dimensional THz tomography:

**Problem 1.5** (**Reduced Inverse problem of the two-dimensional THz tomography**). *Determine the refractive index* $n(\mathbf{x})$ *for a domain* $\Omega \subseteq \mathbb{R}^2$ *by measurements of the travel time* $T_i$ *of an electromagnetic wave with* $i = 1, ..., N$ *receivers.*

For our investigation, we focus on plastics and ceramics since THz radiation can propagate through these materials. Especially in the plastics industry, many products are produced during an extrusion process. Inline monitoring plays an important role to intervene directly during the production process. The rate of rejected parts should be reduced. Fast feedback about defects is desirable. So, a second motivation of the underlying thesis, besides the reconstruction of the refractive index, is to reduce the reconstruction time by integrating concepts of machine learning. Starting with the mathematical and numerical investigation of the reduced inverse problem

of THz tomography and solving it via Landweber's method, we aim in a second step to accelerate the iteration of our reconstruction algorithm. For this purpose, we study CNNs to learn the solution of the parameter-to-solution operator, given by the eikonal equation, with a deep neural network instead of using the FMM. The numerical solution of this first-order non-linear differential equation is the most time-consuming step using Landweber's iteration. By substituting it with a deep neural network, we reduce the numerical effort of one reconstruction through an a prior learning process.

A third and final motivation for the doctoral thesis includes public discussions: One of the most debated topics in this day and age is the climate crisis. How can we generate clean energy? How can we store energy? This discussion must also be considered in the context of artificial intelligence. On the one hand, artificial intelligence can be used in the context of smart grid design, developing low-emission infrastructure and modeling climate change predictions to reduce the effect of the climate crises [16]. On the other hand, the carbon footprint of training a deep neural network can be enormous. Using a deep neural network consumes a high amount of energy. Strubell et al. [87] from the University of Massachusetts investigated that the converted carbon dioxide ($CO_2$) emission for training a deep neural network is about eight times higher than the average $CO_2$ footprint consumed by a car during its lifetime. Compared to the $CO_2$ emission of the learning process of a deep neural network that is about 400 kg, a human being consumes an average of 5 kg during one year, see Table 1.1. There are three factors resulting in high energy consumption for a deep neural network: the size of the training data set, the execution of the model for one single input and the tuning of the hyperparameters of the model which results in a huge amount of training processes. Up to now, there are no mathematical rules for the used hyperparameters like, e.g. the depth and width of a network and they are determined via a trial and error process.

Table 1.1: The $CO_2$ footprint for training a deep neural network compared to the energy consumption of a car during its lifetime and the average $CO_2$ emission produced by a human being within one year (modified according to [87])

| | $CO_2$ emission |
|---|---|
| training DNN (model architecture, hyperparameter) | $\approx$ 424 kg |
| car, avg incl. fuel | $\approx$ 57 kg |
| human life, avg, 1 year | $\approx$ 5 kg |

The greater computer power in the modern world is accompanied by an increase in the number of parameters used for a deep neural network to learn some tasks. The $CO_2$ footprint continues to grow. So, respecting the actual discussions in the context of artificial intelligence, we want to learn the parameters of the network in a sparse way in the last step of our thesis. We aim to make the algorithm sustainable by deleting connections in the network. If we set some parameters in the network equal to zero, we do not need to calculate their gradients during the optimization process. Further, we obtain sparse matrices for the network parameters that require

less storage.

In summary, the motivation for the dissertation can be stated in the following three points:

- We want to solve Problem 1.5, the reduced inverse problem of the two-dimensional THz tomography, via the eikonal equation and Landweber's method within a mathematical framework.
- We want to accelerate the reconstruction algorithm including a CNN by substituting the solution of the forward operator.
- We want to save energy in the learning process of the network by replacing the CNN with a sparse CNN.

Note, that the motivation arose from a previous project in collaboration with the SKZ in Würzburg promoted by the **Arbeits-Gemeinschaft industrieller Forschungs-Vereinigungen** (AiF) in which we focused on the detection of anomalies in an inline process using THz radiation. The results of the project are described in the last chapter of the thesis.

# 2 Physical background

In this chapter, we introduce the physical basis of our investigations. Dealing with THz tomography, i.e. electromagnetic radiation, we need to study our direct problem by electromagnetic laws. The propagation of electromagnetic radiation can be described by partial differential equations. Starting with the Lorentz force law and Maxwell's equations we derive step by step the first-order non-linear eikonal equation. As a high-frequency approximation of the Helmholtz equation, the eikonal equation is used to model the parameter-to-solution operator that is a part of the forward operator of the reduced two-dimensional inverse problem of THz tomography, see Problem 1.5. We obtain a physical relation between the process parameter to be determined, the refractive index $n(\mathbf{x})$, and the THz data acquired by the measuring process, the travel time $T(x)$ of the radiation. The Helmholtz equation results from the wave equation that is an implication of Maxwell's equations. The Lorentz force law and Maxwell's equations describe the interaction between electric and magnetic fields, and more generally, of all electromagnetic phenomena. The following sections are based on several standard literature on physical electromagnetic field theory. We mainly focus on the books of Born and Wolf [8], Demtröder [15] and González-Acuna and Chaparro-Romo [24].

## 2.1 Maxwell's equations

Maxwell's equations can be written in an integral form, but also in a differential one respecting the local character. Both versions are converted into each other by using the Divergence theorem and Stokes' theorem. For our derivation, we focus on the differential case. To formulate Maxwell's equations, we start with the electromagnetic force, the Lorentz force law, to introduce the electric and the magnetic field.
We define the **electric field** in position $\mathbf{x} = (x_1, x_2, x_3)^\intercal \in \mathbb{R}^3$ at time $t$ by $E(t, \mathbf{x})$ as a vector field in which a particle of electric charge $q$ experiences an **electric force**

$$F_E(t, \mathbf{x}) = qE(t, \mathbf{x}). \tag{2.1}$$

Further, we denote the **induction field** in position $\mathbf{x} \in \mathbb{R}^3$ at time $t$ by $B(t, \mathbf{x})$ as a vector field in which a stationary electric charge $q$ suffers no magnetic influence while a moving electric charge experiences a **magnetic force** perpendicular to its velocity $v(t, \mathbf{x})$ by

$$F_B(t, \mathbf{x}) = q\Big(v(t, \mathbf{x}) \times B(t, \mathbf{x})\Big). \tag{2.2}$$

The **Lorentz force law** describes the total electromagnetic force for a point electric charge exposed to an electric field and an induction field by combining equation (2.1)

and (2.2) to

$$F(t,\mathbf{x}) = q\Big(E(t,\mathbf{x}) + v(t,\mathbf{x}) \times B(t,\mathbf{x})\Big). \tag{2.3}$$

If the charge moves parallel to the induction field, the total force depends only on the electric field, see equation (2.3).

In linear, isotropic media, we associate the **electric displacement field** $D$ with the electric field $E$ by

$$D(t,\mathbf{x}) = \epsilon(\mathbf{x})E(t,\mathbf{x})$$

and the **magnetic field** $H$ with the induction field $B$ by

$$B(t,\mathbf{x}) = \mu(\mathbf{x})H(t,\mathbf{x}).$$

The parameters $\epsilon(\mathbf{x})$ and $\mu(\mathbf{x})$ depend on the position $\mathbf{x}$ and are called **dielectric permittivity** and **magnetic permeability** respectively. We assume that both parameters are time-independent. The dielectric permittivity describes the influence of the material on the electric field. It is determined by the relative permittivity $\epsilon_r(\mathbf{x})$ and the permittivity of vacuum $\epsilon_0$ via

$$\epsilon(\mathbf{x}) = \epsilon_r(\mathbf{x})\epsilon_0.$$

In general, if $\epsilon_r(\mathbf{x}) > 1$ the electric field is weakened. Identically, the magnetic permeability is defined via the relative permeability $\mu_r(\mathbf{x})$ and the permeability of free space $\mu_0$:

$$\mu(\mathbf{x}) = \mu_r(\mathbf{x})\mu_0.$$

To formulate Maxwell's equations, we define the **charge density** $\rho(t,\mathbf{x})$ as the charge per volume that effects on the **electric current density** $j(t,\mathbf{x})$ via

$$j(t,\mathbf{x}) = \rho(t,\mathbf{x})v(t,\mathbf{x})$$

where $v$ is again the speed of the charged particle.

**Notation 2.1** ([46], p. 22 et seq.)**.**
*For a three-dimensional vector field $f : \mathbb{R}^3 \to \mathbb{R}^3$, we define the **nabla operator** as $\nabla := (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3})^\intercal$, such that the inner product*

$$\nabla \cdot f := \frac{\partial}{\partial x_1} f_1 + \frac{\partial}{\partial x_2} f_2 + \frac{\partial}{\partial x_3} f_3$$

*represents the **divergence** of the vector field $f = (f_1, f_2, f_3)^\intercal$ and $\nabla \times f$ the **curl** of the vector field $f$ defined by the cross product*

$$\nabla \times f := \begin{pmatrix} \frac{\partial}{\partial x_2} f_3 - \frac{\partial}{\partial x_3} f_2 \\ \frac{\partial}{\partial x_3} f_1 - \frac{\partial}{\partial x_1} f_3 \\ \frac{\partial}{\partial x_1} f_2 - \frac{\partial}{\partial x_2} f_1 \end{pmatrix}.$$

After introducing all relevant physical parameters, we can formulate the **local Maxwell equations** as follows:

$$\nabla \cdot D(t, \mathbf{x}) = \rho(t, \mathbf{x}) \tag{2.4}$$
$$\nabla \cdot B(t, \mathbf{x}) = 0 \tag{2.5}$$
$$\nabla \times E(t, \mathbf{x}) = -\frac{\partial}{\partial t} B(t, \mathbf{x}) \tag{2.6}$$
$$\nabla \times H(t, \mathbf{x}) = j(t, \mathbf{x}) + \frac{\partial}{\partial t} D(t, \mathbf{x}). \tag{2.7}$$

The four partial differential equations are accepted as axioms and describe the propagation of electromagnetic waves in space in general. The first equation (2.4), known as **Gauss's law**, describes the total electric flux through a closed surface. The analog for the magnetic flux is represented in equation (2.5) and is called **Gauss's law for magnetism**. It claims that the total magnetic flux passing through a closed surface is equal to zero. **Faraday's law**, equation (2.6), tells that a magnetic field that changes in time produces a circulating electric field. According to that, the last equation (2.7), **Ampère's law**, declares that a circulating magnetic field is generated by a changing electric displacement field and a current density.

For our purposes, we consider a special case of investigation corresponding to the study of plastics and ceramics: The space contains no charge. As a consequence, the charge density $\rho$ and the electric current density $j$ are equal to zero. Additionally, we assume that our material is non-conductive and non-magnetizable, i.e. the magnetic permeability $\mu$ is set to $\mu_0$. With these assumptions, one can transform Maxwell's equations to

$$\nabla \cdot E(t, \mathbf{x}) = 0 \tag{2.8}$$
$$\nabla \cdot B(t, \mathbf{x}) = 0 \tag{2.9}$$
$$\nabla \times E(t, \mathbf{x}) = -\frac{\partial}{\partial t} B(t, \mathbf{x}) \tag{2.10}$$
$$\nabla \times B(t, \mathbf{x}) = \mu_0 \epsilon(\mathbf{x}) \frac{\partial}{\partial t} E(t, \mathbf{x}) \tag{2.11}$$

which is the basis for the further derivation. To result in equation (2.8), we set

$$\nabla \cdot \Big( \epsilon(\mathbf{x}) E(t, \mathbf{x}) \Big) \approx \epsilon(\mathbf{x}) \nabla \cdot E(t, \mathbf{x}),$$

i.e. the dielectric permittivity varies weakly.

## 2.2 Wave equation

To obtain the eikonal equation, it is necessary to derive the wave equation that is a consequence of the transformed Maxwell equations. We take Faraday's law (2.10) and apply the curl operator on both sides:

$$\nabla \times \Big( \nabla \times E(t, \mathbf{x}) \Big) = -\nabla \times \Big( \frac{\partial}{\partial t} B(t, \mathbf{x}) \Big). \tag{2.12}$$

**Notation 2.2** ([46], p. 25)**.**
*For a scalar field $f : \mathbb{R}^3 \to \mathbb{R}$, we define the* ***gradient*** *of $f$ as*

$$\nabla f = \Big(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}\Big)^\intercal$$

*and the* ***Laplacian oparator*** $\Delta$ *as*

$$\Delta f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \frac{\partial^2 f}{\partial x_3^2}.$$

In the next step, we use the identity

$$\nabla \times \Big(\nabla \times E(t, \mathbf{x})\Big) = \nabla\Big(\nabla \cdot E(t, \mathbf{x})\Big) - \Delta E(t, \mathbf{x}).$$

Combined with equation (2.11) and (2.12), we obtain

$$\nabla\Big(\nabla \cdot E(t, \mathbf{x})\Big) - \Delta E(t, \mathbf{x}) = -\mu_0 \epsilon(\mathbf{x}) \frac{\partial^2}{\partial t^2} E(t, \mathbf{x}).$$

The last line is reduced to the **wave equation**

$$\Delta E(t, \mathbf{x}) - \mu_0 \epsilon(\mathbf{x}) \frac{\partial^2}{\partial t^2} E(t, \mathbf{x}) = 0 \tag{2.13}$$

for the electric field $E$ by applying equation (2.8). In vacuum, i.e. $\epsilon(\mathbf{x}) = \epsilon_0$, it holds

$$\Delta E(t, \mathbf{x}) - \frac{1}{c_0^2} \frac{\partial^2}{\partial t^2} E(t, \mathbf{x}) = 0. \tag{2.14}$$

The constant $c_0 = \sqrt{\mu_0 \epsilon_0}^{-1}$ is called **speed of light**, see Definition 1.1. The refractive index is coupled with the relative permittivity and the relative permeability by

$$n(\mathbf{x}) = \sqrt{\mu_r(\mathbf{x}) \epsilon_r(\mathbf{x})}. \tag{2.15}$$

We transform the wave equation (2.13) using Definition 1.1 and equation (2.15) to

$$\Delta E(t, \mathbf{x}) - \frac{1}{v^2(\mathbf{x})} \frac{\partial^2}{\partial t^2} E(t, \mathbf{x}) = 0. \tag{2.16}$$

The wave equation, a second-order linear partial differential equation, describes the temporal and local behavior of the electric field $E$ in the considered medium depending on the refractive index. The equation is used to describe the propagation of many other waves, such as sound waves or water waves.

**Remark 2.3.**
*If we apply the curl operator to Ampère's law, we obtain the same result for the induction field, i.e.*

$$\Delta B(t, \mathbf{x}) = \frac{1}{v^2(\mathbf{x})} \frac{\partial^2}{\partial t^2} B(t, \mathbf{x}).$$

## 2.3 Helmholtz equation

To solve the wave equation, we focus on time harmonic electric waves $E(t, \mathbf{x})$ and separate them into a factor that depends on the position $\mathbf{x}$ and a part that is a harmonic function of the time $t$. We write

$$E(t, \mathbf{x}) = u(\mathbf{x})e^{i\omega t} \tag{2.17}$$

and insert this separation of variables into the wave equation:

$$\begin{aligned} e^{i\omega t}\Delta u(\mathbf{x}) - \frac{1}{v^2(\mathbf{x})}u(\mathbf{x})\frac{\partial^2}{\partial t^2}e^{i\omega t} &= 0 \\ \Leftrightarrow e^{i\omega t}\Delta u(\mathbf{x}) + \frac{\omega^2}{v^2(\mathbf{x})}u(\mathbf{x})e^{i\omega t} &= 0. \end{aligned}$$

Dividing by $e^{i\omega t}$, we derive the partial differential equation

$$\Delta u(\mathbf{x}) + \frac{\omega^2}{v^2(\mathbf{x})}u(\mathbf{x}) = 0 \tag{2.18}$$

for the spatial component of equation (2.17) which is known as the **Helmholtz equation**. The parameter $\omega$ refers to the **angular frequency** of the electric wave that is related to the **frequency f** by

$$\omega = 2\pi\mathbf{f}.$$

The Helmholtz equation is often given in the form

$$\Delta u(\mathbf{x}) + k(\mathbf{x})^2 u(\mathbf{x}) = 0$$

with the **wave number**

$$k(\mathbf{x}) = \frac{\omega}{v(\mathbf{x})}.$$

## 2.4 Eikonal equation

The eikonal equation can finally be seen as a high-frequency approximation of the Helmholtz equation (2.18). In the last step, we consider a time shift in the solution of the form

$$u(\mathbf{x}) = A(\mathbf{x})e^{-i\omega T(\mathbf{x})},$$

such that we write the electric field as

$$E(t, \mathbf{x}) = A(\mathbf{x})e^{i\omega\left(t - T(\mathbf{x})\right)}.$$

The function $A(\mathbf{x})$ represents the local wave amplitude which may, in general, be a complex-valued function. In our case, we consider linear polarized electromagnetic waves resulting in a real-valued function $A(\mathbf{x})$. The duration $T(\mathbf{x})$ determines the time of the wave until it reaches position $\mathbf{x}$ from the source. We calculate

$$\nabla u(\mathbf{x}) = e^{-i\omega T(\mathbf{x})}\Big(\nabla A(\mathbf{x}) - i\omega A(\mathbf{x})\nabla T(\mathbf{x})\Big)$$

and consequently

$$\begin{aligned}\Delta u(\mathbf{x}) &= -i\omega\nabla T(\mathbf{x})e^{-i\omega T(\mathbf{x})}\Big(\nabla A(\mathbf{x}) - i\omega A(\mathbf{x})\nabla T(\mathbf{x})\Big)\\ &\quad + e^{-i\omega T(\mathbf{x})}\Big(\Delta A(\mathbf{x}) - i\omega A(\mathbf{x})\Delta T(\mathbf{x}) - i\omega\nabla A(\mathbf{x})\nabla T(\mathbf{x})\Big)\\ &= e^{-i\omega T(\mathbf{x})}\Big(\Delta A(\mathbf{x}) - \omega^2 A(\mathbf{x})\big(\nabla T(\mathbf{x})\big)^2\Big)\\ &\quad - ie^{-i\omega T(\mathbf{x})}\Big(2\omega\nabla T(\mathbf{x})\nabla A(\mathbf{x}) - \omega A(\mathbf{x})\Delta T(\mathbf{x})\Big).\end{aligned}$$

Inserting this result in equation (2.18) and dividing by $e^{-i\omega T(x)}$, we obtain

$$\begin{aligned}&\Big(\Delta A(\mathbf{x}) - \omega^2 A(\mathbf{x})\big(\nabla T(\mathbf{x})\big)^2 + \frac{\omega^2}{v^2(\mathbf{x})}A(\mathbf{x})\Big)\\ &- i\Big(2\omega\nabla T(\mathbf{x})\nabla A(\mathbf{x}) - \omega A(\mathbf{x})\Delta T(\mathbf{x})\Big) = 0.\end{aligned}$$

To be a solution of the Helmholtz equation, both, the real part and the imaginary part, have to be equal to zero. We focus on the real part

$$\Delta A(\mathbf{x}) - \omega^2 A(\mathbf{x})\big(\nabla T(\mathbf{x})\big)^2 + \frac{\omega^2}{v(\mathbf{x})^2}A(\mathbf{x}) = 0.$$

Dividing by $\omega^2$ and using a high-frequency approximation, i.e. $\Delta A(\mathbf{x})\omega^{-2}$ tends to zero for high frequencies, we deduce the eikonal equation as a first-order non-linear partial differential equation since $A(\mathbf{x})$ does not vanish everywhere as

$$|\nabla T(\mathbf{x})|^2 = \frac{1}{v^2(\mathbf{x})}. \tag{2.19}$$

We use the standard notation for the **Euclidean norm**:

$$\big(\nabla T(\mathbf{x})\big)^2 = \Big(\frac{\partial T}{\partial x_1}\Big)^2 + \Big(\frac{\partial T}{\partial x_2}\Big)^2 + \Big(\frac{\partial T}{\partial x_3}\Big)^2 = ||\nabla T(\mathbf{x})||_2^2 = |\nabla T(\mathbf{x})|^2. \tag{2.20}$$

It results a direct correlation between the travel time $T(\mathbf{x})$ of the electromagnetic wave required to reach a point $\mathbf{x}$ and the material-dependent velocity of the wave $v(\mathbf{x})$. Equation (2.19) is an adequate choice for modeling the propagation of THz radiation, and so, for the definition of the parameter-to-solution operator of the considered inverse problem in the next step. Note that we assumed $\mathbf{x} \in \mathbb{R}^3$ throughout this chapter; however, the same result is obtained for $\mathbf{x} \in \mathbb{R}^2$ that we utilize in the further course of the thesis. It should also be mentioned that, apart from Chapter 5, we do not model the Gaussian beam to stay within a manageable framework.

# 3 Inverse problem of terahertz tomography

After deducing the eikonal equation in the previous chapter, we use it as a model to construct the forward operator to solve the reduced inverse problem of THz tomography, see Problem 1.5. The eikonal equation is a first-order non-linear partial differential equation that results in a non-linear operator. We define the forward operator $F$ as a composition of three operators by subdividing $F$ into

$$F = \mathcal{Q} \circ \gamma \circ S.$$

By $S$ we understand the parameter-to-solution operator for which the eikonal equation is chosen. The operator $S$ simulates the propagation of an electromagnetic wave and maps the material-dependent velocity $v(\mathbf{x})$ to the travel time $T(\mathbf{x})$. The measured data are only available on the boundary of the considered domain. We need an operator, the trace operator $\gamma$, that restricts the solution of the eikonal equation to this boundary. In reality, we only measure discrete data. An observation operator $\mathcal{Q}$ is applied. The observation operator models the setting of the measuring process and maps to an $N$-dimensional vector, where $N$ describes the number of receivers. To prove that the operator $S$ is well-defined, we consider a special setting for the refractive index and the travel time to apply the results of the paper '*Generalized solution of the Hamilton-Jacobi equations of eikonal type*' written by S. N. Kruzhkov [57].

In the second step, after defining the non-linear forward operator $F$, we apply the iterative Landweber method to stabilize the resulting ill-posed problem. For this purpose, the forward operator has to be linearized. Then, the adjoint of the linearized forward operator is calculated resulting again in a partial differential equation. To execute the iterative Landweber method, we numerically solve the forward operator implementing the FMM. The linearized adjoint problem is solved numerically via a fast sweeping method including a Gauss-Seidel iteration. Finally, first reconstructions are presented in this chapter for various refractive indices.

## 3.1 The forward operator based on the eikonal equation

The objective of the reduced inverse problem of THz tomography is to reconstruct either the refractive index $n(\mathbf{x})$ or the velocity $v(\mathbf{x})$. This reconstruction is based on discrete measurements of the travel time $T(\mathbf{x})$ taken by an electromagnetic wave

to reach the boundary, originating from the initial position $\mathbf{x_0} = (x_0, y_0)^\intercal \in \Omega$. We limit to the two-dimensional space, i.e. we consider a domain $\Omega \in \mathbb{R}^2$. Consequently, we neglect the change in $n$ in $z$-direction. We have

$$n(\mathbf{x}) = n(x, y, z) = n(x, y, 0) = n(x, y)$$

for all $z \in \mathbb{R}$. Furthermore, we note that outside the object, the refractive index $n(\mathbf{x})$ is equal to 1. Therefore, we can reasonably assume that we are operating in an environment approximating a vacuum where $v(\mathbf{x}) \approx c_0$, see Definition 1.1. It is worth noting that the refractive index of air only deviates from this assumption at the fourth decimal place.

The first part of the direct problem of THz tomography is obtained via the eikonal equation:

$$|\nabla T_v(x, y)|^2 = \frac{1}{v^2(x, y)}, \quad \text{for all } (x, y)^\intercal \in \Omega. \tag{3.1}$$

To show the existence and uniqueness of a solution for the eikonal equation with Dirichlet boundary condition, we adopt the results from Kruzhkov, see [57]. We deal with a Hamilton-Jacobi equation of the form

$$H(\mathbf{x}, T, \nabla T) := f(\mathbf{x}, T, p) - n^2(x, y) = 0, \tag{3.2}$$

where

$$p := \nabla T \text{ and } f(\mathbf{x}, T, p) := |\nabla T|^2.$$

According to Kruzhkov, we assume $n \in C^{2,\alpha}(\Omega)$, $0 < \alpha < 1$, to have Hölder-continuous derivatives up to second order and formulate the following four assumptions that are obviously satisfied:

**Assumption 3.1** ([57], p. 418)**.**

*a) The function $f(\mathbf{x}, T, p)$ has Hölder-continuous derivatives up to second order in $\bar{\Omega} \times \mathbb{R} \times \mathbb{R}^2$: $f \in C^{2,\alpha}$, $0 < \alpha < 1$. Further, we have $f(\mathbf{x}, T, 0) \equiv 0$ and $n(x, y) > 0$.*

*b) Uniformly with respect to $(\mathbf{x}, T, p) \in Q$, where $Q$ is a compactum in $\Omega \times \mathbb{R} \times \mathbb{R}^2$:*

$$\inf_{\xi \in \mathbb{R}^2, |\xi|=1} (\xi, f_{pp}(\mathbf{x}, T, p)\xi) \geq \lambda = const > 0.$$

*c) For any $(\mathbf{x}, T, p) \in \Omega \times \mathbb{R} \times \mathbb{R}^2$*

$$H_T(\mathbf{x}, T, p) \leq 0, \quad H(\mathbf{x}, T, p) \equiv f(\mathbf{x}, T, p) - n^2(x, y).$$

*d) Uniformly with respect to $\mathbf{x} \in B_r(\mathbf{x}_0) \cap \bar{\Omega}$, for any fixed $\mathbf{x}_0$, $r$ and $M$*

$$f(\mathbf{x}, M, p) \to +\infty \text{ as } |p| \to \infty.$$

*Proof.*

a) Clear, according to the definition of $f$ and $n$.

b) We have $f_{pp} = 2I$ and it follows $\lambda = 2$.

c) We have $H_T \equiv 0$.

d) It holds $f = |p|^2 \to +\infty$ for $|p| \to \infty$.

□

We now suppose that a function $\varphi \in Lip(\partial\Omega)$ is given on the boundary $\partial\Omega$ satisfying the Dirichlet boundary condition

$$T|_{\partial\Omega} = \varphi. \tag{3.3}$$

By Rademacher's theorem, see e.g. [20], the gradient $\nabla\varphi$ exists at almost every interior point of $\Omega$. Further, we define a generalized solution of problem (3.2) and (3.3) as follows:

**Definition 3.2** (**Generalized solution**)**.**
*A function $u \in Lip(\overline{\Omega}) \cap E(\Omega)$ is called generalized solution if it satisfies the equations (3.2) and (3.3) almost everywhere in $\Omega$. A function $u$ is an element of $E(\Omega)$ if the following inequation is satisfied $\forall x, x + \Delta x, x - \Delta x \in K_\delta(y)$ (where $K_\delta(y)$ is a ball centered at $y$ with the radius $\delta$):*

$$\frac{u(x + \Delta x) - 2u(x) + u(x - \Delta x)}{|\Delta x|^2} \geq -C(y, \delta).$$

**Theorem 3.3** (**Existence of a solution**, [57], p. 418)**.**
*Given the conditions of Assumption 3.1, there exists a generalized solution of problem (3.2), (3.3) if and only if there exists an extension of the boundary function $\varphi$ into $\Omega$ such that $\varphi \in Lip(\bar{\Omega})$ (preserving the notation $\varphi$ for it) and almost everywhere in $\Omega$*

$$f(\mathbf{x}, \varphi, \nabla\varphi) - n^2(x, y) \leq 0.$$

*Proof.* See [57], p. 418 et seqq. □

**Theorem 3.4** (**Uniqueness of a solution**, [57], p. 411)**.**
*The generalized solution of the problem (3.2) and (3.3) is unique.*

*Proof.* See [57], p. 411 et seqq. □

At this point, we define the first part of the forward operator, the parameter-to-solution operator $S$, that is essential for our investigation of the direct and inverse problem of THz tomography. We suppose that there exists a function $\varphi_v(x, y) \in Lip(\partial\Omega)$ such that there exists an extension $\varphi_v(x, y) \in Lip(\bar{\Omega})$ (we preserve the notation $\varphi_v$ for it). Overall, we assume

$$\varphi_v(x, y) \in Lip(\bar{\Omega}), \quad \varphi_v(x_0, y_0) = 0, \tag{3.4}$$

$$|\nabla\varphi_v(x, y)|^2 - \frac{1}{v^2(x, y)} \leq 0 \tag{3.5}$$

**Definition 3.5 (Parameter-to-solution operator).**
*Let $v \in C^{2,\alpha}(\Omega)$. We define the non-linear parameter-to-solution operator*

$$\begin{aligned} S : \mathcal{D}(S) \subseteq C^{2,\alpha}(\Omega) &\to Lip(\bar{\Omega}) \\ v &\mapsto S(v) := T_v, \end{aligned}$$

*where $0 < \alpha < 1$ and $T_v$ solves the partial differential equation (3.1) with the Dirichelet boundary condition*

$$T_v(x, y) = \varphi_v(x, y) \text{ on } \partial\Omega \tag{3.6}$$

*and the initial condition*

$$T_v(x_0, y_0) = 0, \text{ for one } (x_0, y_0)^\intercal \in \partial\Omega. \tag{3.7}$$

*We have*

$$\mathcal{D}(S) = \{v \in C^{2,\alpha}(\Omega) : \varphi_v \in Lip(\overline{\Omega}),\ \varphi_v(x_0, y_0) = 0,\ |\nabla\varphi_v|^2 - \frac{1}{v^2} \leq 0\}.$$

**Theorem 3.6.**
*The parameter-to-solution operator $S$ is well-defined under the assumptions (3.4) and (3.5) for $\varphi_v$.*

*Proof.* The statement follows directly from Theorem 3.3 and 3.4. □

The operator $S$ calculates the travel time $T_v$ required for the THz radiation to reach the point $(x, y)^\intercal$ from its starting point $(x_0, y_0)^\intercal$ for a given refractive index $n$ and thus a given velocity $v$. Due to the fact that the travel time is increasing in time, we can suppose that $\varphi_v(x, y) \not\equiv 0$ on $\partial\Omega$.

**Remark 3.7.**

a) *One possibility to solve non-linear first-order partial differential equations is the method of characteristics. The partial differential equation is transformed into a system of ordinary differential equations. In this work, we solve the eikonal equation by the FMM, see Subsection 3.4.1. For more information about the method of characteristics, we refer to Evans [21].*

b) *The uniqueness of the solution is investigated in the viscosity sense in [13]. For our investigation, we focus on the discretized case where we have existence and uniqueness by Theorem 3.24.*

c) *Since the partial derivatives of the unknown function $T$ appear in quadratic form in equation (3.1), it is a non-linear differential equation. Consequently, the unique operator $S$ is indeed non-linear, see Appendix B.7.*

In the context of tomographic processes, the emission and measurement of radiation by both, the sender and receivers, typically occur along a curve encircling the object under investigation, refer to Figure 1.2. The domain denoted as $\Omega$ is defined as the interior of the measurement setup. The data is generated along the boundary $\partial\Omega$ of this region. To characterize our forward operator, we must confine the operator $S$ to the boundary of our specified region. We propose that the boundary $\partial\Omega$ possesses the

property of being $\mathcal{C}^1$ class, enabling the establishment of a trace operator known for its continuity and linearity. The trace operator restricts an element of the continuous functions to the boundary of the domain $\Omega$.

Let $\Omega \subseteq \mathbb{R}^n$ be an open, bounded domain of class $C^1$ and $u \in C(\bar{\Omega})$. Then, $\gamma : C(\bar{\Omega}) \to C(\partial\Omega)$ defined by

$$\gamma u = u|_{\partial\Omega}.$$

is a linear and continuous mapping. We have

$$\sup_{\mathbf{x}\in\partial\Omega} |u(\mathbf{x})| \leq \sup_{\mathbf{x}\in\bar{\Omega}} |u(\mathbf{x})|.$$

**Definition 3.8** (**Trace operator**)**.**
*The operator*

$$\gamma : Lip(\bar{\Omega}) \subset C(\bar{\Omega}) \to C(\partial\Omega), \;\; T_v \mapsto T_v|_{\partial\Omega}$$

*denotes the trace operator.*

To define the complete forward operator, the measuring process must be taken into account. We use a linear observation operator to model the measuring process and assume that the surfaces of all receivers are contained in $\partial\Omega$. We are guided by the work of Wald et all, see [93]. Before introducing the observation operator, we need some notation:

**Notation 3.9.**
*Let $N \in \mathbb{N}$ be the number of receivers. By*

$$E_\nu \subseteq \partial\Omega, \;\; \nu = 1, 2, ..., N$$

*we denote the surfaces of the $N$ receivers. Further, we introduce the* ***sensor characteristic*** *of the receiver $E_\nu$ as a real-valued linear functional $e_\nu \in C(\partial\Omega)^*$*

$$e_\nu : C(\partial\Omega) \to \mathbb{R}.$$

**Definition 3.10** (**Observation operator**)**.**
*The operator $\mathcal{Q} : C(\partial\Omega) \to \mathbb{R}^N$ with*

$$\begin{aligned} \varphi \mapsto \mathbf{y} := (\mathcal{Q}\varphi)_{\nu=1,...,N} &= \Big( \big\langle \varphi, e_\nu \big\rangle_{C(\partial\Omega)\times C(\partial\Omega)^*} \Big)_{\nu=1,...,N} \\ &= \left( \int_{\partial\Omega} e_\nu(\mathbf{x})\varphi(\mathbf{x}) \, ds_{\mathbf{x}} \right)_{\nu=1,...,N} \end{aligned} \tag{3.8}$$

*denotes the observation operator modeling the measurement process of THz tomography.*

**Remark 3.11.**

*a) The observation operator is a continuous linear operator.*

*b) The observation operator maps the restricted travel time $\gamma T_v \in C(\partial\Omega)$ to the measured values $\mathbf{y} = (y_1, ..., y_N)^\intercal \in \mathbb{R}^N$.*

*c) For our investigations, we use point evaluations, i.e.*

$$e_\nu(\mathbf{x}) := \delta(\mathbf{x} - \mathbf{x}_{E_\nu}),\ \nu = 1, 2, ..., N.$$

*The point $\mathbf{x}_{E_\nu} \in \partial\Omega$ indicates the position of the receiver $E_\nu$. Consequently, we can rewrite*

$$\mathbf{y} = (\gamma T_v(\mathbf{x}_{E_1}), ..., \gamma T_v(\mathbf{x}_{E_N}))^\intercal = (T_v(\mathbf{x}_{E_1}), ..., T_v(\mathbf{x}_{E_N}))^\intercal.$$

Finally, we summarize our derivation by defining the forward operator as a composition of the observation operator $\mathcal{Q}$, the trace operator $\gamma$ and the parameter-to-solution operator $S$:

**Definition 3.12** (**Forward operator**).
*The operator $F : \mathcal{D}(S) \subset C^{2,\alpha}(\Omega) \to \mathbb{R}^N$, $0 < \alpha < 1$, with*

$$F(v) := \mathcal{Q}\gamma S(v) = (T_v(\mathbf{x}_{E_1}), ..., T_v(\mathbf{x}_{E_N}))^\intercal$$

*is called forward operator of the reduced THz setting.*

## 3.2 Linearization of the forward operator

In the previous section, we deduced the forward operator $F$, an operator that describes the mathematical model of the direct problem of THz tomography. Because $S$, and consequently $F$, is non-linear, we deal with a non-linear inverse problem. To solve this problem with the iterative Landweber method the linearization of the forward operator $F$ is required. We approximate the non-linear forward operator by a linear operator. For this, we need the Gâteaux derivative:

**Definition 3.13** (**Gâteaux derivative** [94], p. 112 et seq.).
*Let $X$, $Y$ be normed spaces, $U \subseteq X$ an open set and $F : U \to Y$. Note that $F$ is in general not linear. The mapping $F$ is called **Gâteaux-differentiable** in $x_0 \in U$, if there exists a continuous linear operator $A_{x_0} \in \mathcal{L}(X, Y)$ fulfilling*

$$\lim_{h \to 0} \frac{F(x_0 + h\nu) - F(x_0)}{h} = A_{x_0}\nu \quad \textit{for all } \nu \in X.$$

*By $F'(x_0) : \nu \mapsto A_{x_0}\nu$ we denote the Gâteaux derivative of $F$ in $x_0 \in U$ in the direction $\nu \in X$.*

To calculate the linearization of the operator $S$, we consider the perturbed problem

$$|\nabla T_{v_h}(x, y)|^2 = \frac{1}{\Big(v(x, y) + \beta h(x, y)\Big)^2}, \quad \text{for all } (x, y)^\intercal \in \Omega, \tag{3.9}$$

$$T_{v_h}(x_0, y_0) = 0, \quad \text{for one } (x_0, y_0)^\intercal \in \partial\Omega, \tag{3.10}$$

$$T_{v_h}(x, y) = \varphi_{v_h}(x, y), \quad (x, y)^\intercal \in \partial\Omega, \tag{3.11}$$

for an $\beta \in \mathbb{R}^+$ with $T_{v_h} := S(v + \beta h)$ and $\varphi_{v_h}$ fulfilling the assumptions (3.4) and (3.5).

**Notation 3.14.**
*For simplicity, we neglect the arguments in the notation and write, for example, $T$ instead of $T(x, y)$.*

We further assume, that the limit

$$\lim_{\beta\to 0} \frac{S(v+\beta h) - S(v)}{\beta} = \lim_{\beta\to 0} \frac{T_{v_h} - T_v}{\beta} =: S'(v)h$$

exists for all $h \in C^{2,\alpha}(\Omega)$.

To calculate the linearization of our operator $S$, we subtract the eikonal equation for $T_v$, see equation (3.1), form the perturbed one for $T_{v_h}$, see equation (3.9):

$$\begin{aligned}(T_{v_h})_x^2 - (T_v)_x^2 + (T_{v_h})_y^2 - (T_v)_y^2 &= \frac{1}{v^2 + 2\beta hv + \beta^2 h^2} - \frac{1}{v^2} \\ \Leftrightarrow (T_{v_h} - T_v)_x (T_{v_h} + T_v)_x + (T_{v_h} - T_v)_y (T_{v_h} + T_v)_y &= \frac{-2\beta hv - \beta^2 h^2}{v^4 + 2v^3\beta h + v^2\beta^2 h^2}.\end{aligned}$$

Dividing by $\beta \neq 0$ and considering the limit $\beta \to 0$ yield

$$\begin{aligned}\Big(\frac{T_{v_h} - T_v}{\beta}\Big)_x (T_{v_h} + T_v)_x + \Big(\frac{T_{v_h} - T_v}{\beta}\Big)_y (T_{v_h} + T_v)_y &= \frac{-2hv - \beta h^2}{v^4 + 2v^3\beta h + v^2\beta^2 h^2} \\ \overset{\beta\to 0}{\Rightarrow} 2[S'(v)h]_x (T_v)_x + 2[S'(v)h]_y (T_v)_y &= -\frac{2h}{v^3} \\ \Leftrightarrow \nabla(S'(v)h) \cdot \nabla T_v &= -\frac{h}{v^3}.\end{aligned}$$

For these steps, we assume the continuity of the operator $S$. The initial condition for $(x_0, y_0)^\intercal$ results by equation (3.7) and (3.10) in

$$\lim_{\beta\to 0} \frac{T_{v_h}(x_0, y_0) - T_v(x_0, y_0)}{\beta} = 0.$$

Based on the previous calculations, we define the linearization of the operator $S$ that depends linear on $h \in C^{2,\alpha}(\Omega)$.

**Theorem 3.15** (**Linearized parameter-to-solution operator**).
*For $v \in \mathcal{D}(S)$ and $T_v = S(v)$, we define the Gâteaux derivative*

$$S'(v) : C^{2,\alpha}(\Omega) \to C^1(\bar{\Omega})$$

*of the parameter-to-solution operator $S$ as the operator, that maps $h \in C^{2,\alpha}(\Omega)$ to the solution $z$ of the partial differential equation*

$$\begin{aligned}\nabla z \cdot \nabla T_v &= -\frac{h}{v^3} \text{ in } \Omega \\ z(x_0, y_0) &= 0 \text{ for one } (x_0, y_0)^\intercal \in \partial\Omega \\ z(x, y) &\not\equiv 0 \text{ on } \partial\Omega.\end{aligned} \tag{3.12}$$

*It holds $S'(v)h = z$.*

**Lemma 3.16.**
*Let $X, Y, Z$ denote normed spaces and $U \subseteq X$, $V \subseteq Y$ open sets.*

*a) If $F : X \to Y$ is linear, we have that $F$ is Gâteaux-differentiable if and only if $F$ is continuous with*

$$F'(x_0) = F, \ x_0 \in X.$$

*b) If $H : U \to Y$ and $G : V \to Z$ with $H(U) \subseteq V$ are Gâteaux-differentiable in $x_0 \in U$, respectively $H(x_0) \in V$, then $G \circ H$ is Gâteaux-differentiable in $x_0$ with*

$$(G \circ H)'(x_0) = G'(H(x_0)) \circ H'(x_0).$$

*Proof.* See Werner [94], p. 121. □

To obtain the linearization of the forward operator $F$, we observe the composition $\mathcal{Q}\gamma S(v)$. The trace operator and the observation operator are linear, continuous and thus, see Lemma 3.16 a), Gâteaux-differentiable. Together with the chain rule, see Lemma 3.16 b), we obtain for the **linearized forward operator** $F'(v) : C^{2,\alpha}(\Omega) \to \mathbb{R}^N$

$$\begin{aligned}\Big[F'(v)\Big]h &= \Big[(\mathcal{Q}\gamma S)'(v)\Big]h \\ &= \Big[\mathcal{Q}'(\gamma S(v)) \circ \gamma'(S(v)) \circ S'(v)\Big]h \\ &= \Big[\mathcal{Q}\gamma S'(v)\Big]h = \Big(z(\mathbf{x}_{E_1}), ..., z(\mathbf{x}_{E_N})\Big)^{\mathsf{T}}.\end{aligned}$$

## 3.3 Adjoint linearized problem

Considering the Landweber iteration, see Section 3.4, we have to investigate the adjoint of the linearized forward operator $F'(v)^*$, that is given by the mapping

$$F'(v)^* : \mathbb{R}^N \to C^{2,\alpha}(\Omega)^*, \ \ \tilde{\sigma} \mapsto \big[(\gamma S'(v))^* \mathcal{Q}^*\big]\tilde{\sigma}.$$

For $\tilde{\sigma} \in \mathbb{R}^N$, we want to find a function $\tilde{\mu} \in C^{2,\alpha}(\Omega)$ such that

$$F'(v)^*\tilde{\sigma} = \tilde{\mu}.$$

We split our calculations into two parts and start with the investigation of the adjoint operator $(\gamma S'(v))^*$. While $\gamma S'(v)$ maps a Hölder-continuous real-valued function $\eta \in C^{2,\alpha}(\Omega)$ to a continuous function on $\partial\Omega$, the adjoint of it maps an element $\mathcal{Q}^*\tilde{\sigma} =: \sigma \in C(\partial\Omega)^*$ to a function $\mu \in C^{2,\alpha}(\Omega)^*$.

We obtain

$$\begin{aligned}\mu(\eta) = \Big(\eta, \mu\Big)_{C^{2,\alpha}(\Omega)\times C^{2,\alpha}(\Omega)^*} &= \Big(\eta, \big(\gamma S'(v)\big)^*\sigma\Big)_{C^{2,\alpha}(\Omega)\times C^{2,\alpha}(\Omega)^*} \\ &= \Big\langle \gamma S'(v)\eta, \sigma \Big\rangle_{C(\partial\Omega)\times C(\partial\Omega)^*}.\end{aligned}$$

Before we formulate Theorem 3.19 defining the adjoint operator $(\gamma S'(v))^*$, we introduce the normal derivative and the product rule for divergence:

**Notation 3.17.**
*The* ***normal derivative*** $\frac{\partial}{\partial \mathbf{n}}$ *denotes the partial directional derivative in the direction of the outward normal vector* $\mathbf{n}$ *of the boundary* $\partial\Omega$*. We write for a function* $f$

$$\frac{\partial f}{\partial \mathbf{n}} := \nabla f \cdot \mathbf{n}.$$

**Lemma 3.18** (**Product rule for divergence**)**.**
*For a scalar-valued function* $\varphi$ *and a vector-valued function* $v$*, we have the following identity:*

$$\operatorname{div}(\varphi v) = \varphi \operatorname{div} v + v\nabla\varphi.$$

*Proof.* See Appendix B.6. □

**Theorem 3.19.**
*Let* $v \in \mathcal{D}(S)$*,* $\sigma \in C(\partial\Omega)^*$*. The adjoint of* $\gamma S'(v)$ *is given by*

$$(\gamma S'(v))^*\sigma := \frac{\phi}{v^3}$$

*where* $\phi \in C^{2,\alpha}(\Omega)^*$ *solves the partial differential equation*

$$\begin{aligned} \Delta T\phi + \nabla T \nabla \phi = 0 \quad &\text{in } \Omega \\ \phi(\nabla T \cdot \boldsymbol{n}) = -\sigma \quad &\text{on } \partial\Omega. \end{aligned} \tag{3.13}$$

*Proof.* Consider the dual pairing of equation (3.12) with some test function $\phi \in C^{2,\alpha}(\Omega)^*$:

$$\begin{aligned} (\nabla z \nabla T_v, \phi)_{C^{2,\alpha}(\Omega)\times C^{2,\alpha}(\Omega)^*} &= -\Big(\frac{h}{v^3}, \phi\Big)_{C^{2,\alpha}(\Omega)\times C^{2,\alpha}(\Omega)^*} \\ \Leftrightarrow \int_\Omega (\nabla z \nabla T_v)\phi \, dxdy &= -\int_\Omega \frac{h}{v^3}\phi \, dxdy. \end{aligned}$$

By setting $(\gamma S'(v))^*\sigma = \frac{\phi}{v^3}$ the right-hand side is reformulated to

$$\begin{aligned} -\int_\Omega \frac{h}{v^3}\phi \, dxdy &= -\Big(h, \frac{\phi}{v^3}\Big)_{C^{2,\alpha}(\Omega)\times C^{2,\alpha}(\Omega)^*} \\ &= -\Big(h, (\gamma S'(v))^*\sigma\Big)_{C^{2,\alpha}(\Omega)\times C^{2,\alpha}(\Omega)^*} \\ &= -\langle \gamma S'(v)h, \sigma\rangle_{C(\partial\Omega)\times C(\partial\Omega)^*} \\ &= -\int_{\partial\Omega} z\sigma \, ds_{\mathbf{x}} \end{aligned}$$

To transform the left-hand side, we use the product rule for divergence. We substitute $\varphi$ and $v$ in Lemma 3.18 by $\varphi := z$ and $v := \nabla T_v\phi$ to obtain

$$\begin{aligned} \int_\Omega (\nabla z \nabla T_v)\phi \, dxdy &= \int_\Omega \operatorname{div}(z\nabla T_v\phi) \, dxdy - \int_\Omega z\big(\operatorname{div}(\nabla T_v\phi)\big) \, dxdy \\ &= \int_\Omega \operatorname{div}(z\nabla T_v\phi) \, dxdy + \int_\Omega z\big(\Delta T_v\phi + \nabla T_v \nabla\phi\big) \, dxdy \end{aligned}$$

We use the Divergence theorem of Gauß to rewrite the first term

$$\int_\Omega \operatorname{div}(z\nabla T_v\phi)\, dxdy = \int_{\partial\Omega} \mathbf{n}\cdot\nabla T_v\phi z\, ds_{\mathbf{x}}.$$

Finally, by summarizing the calculations, we obtain:

$$\int_{\partial\Omega} z\big(\mathbf{n}\cdot\nabla T_v\phi\big)\, ds_{\mathbf{x}} = -\int_{\partial\Omega} z\sigma\, ds_{\mathbf{x}}$$

and

$$\int_\Omega z\big(\Delta T_v\phi + \nabla T_v\nabla\phi\big)\, dxdy = 0.$$

This yields (3.13) since the solution $z$ of the linearized problem typically does not vanish either on $\Omega$ or its boundary $\partial\Omega$. □

In the second step, we formulate the adjoint of the observation operator:

**Lemma 3.20.**
*The adjoint of the observation operator (3.8) is defined by $\mathcal{Q}^* : \mathbb{R}^N \to C(\partial\Omega)^*$ with*

$$\tilde\sigma := (\tilde\sigma_\nu)_{\nu=1,\dots,N} \mapsto \mathcal{Q}^*\tilde\sigma = \sum_{\nu=1}^N \tilde\sigma_\nu\delta(\mathbf{x}-\mathbf{x}_{E\nu}). \tag{3.14}$$

*Proof.* For $\varphi \in C(\partial\Omega)$ and $\tilde\sigma \in \mathbb{R}^N$, we use the Euclidean scalar product on $\mathbb{R}^N$ respecting $(\mathbb{R}^N)^* \cong \mathbb{R}^N$ such that we have

$$\begin{aligned}\langle\mathcal{Q}\varphi,\tilde\sigma\rangle_{\mathbb{R}^N} &= \langle\varphi,\mathcal{Q}^*\tilde\sigma\rangle_{C(\partial\Omega)\times C(\partial\Omega)^*}\\ \Leftrightarrow \sum_{\nu=1}^N \varphi(x_{E_\nu})\tilde\sigma_\nu &= \int_{\partial\Omega}\varphi\mathcal{Q}^*\tilde\sigma\, ds_{\mathbf{x}}.\end{aligned}$$

by using the definitions of the scalar products. We identify

$$\mathcal{Q}^*\tilde\sigma = \sum_{\nu=1}^N \tilde\sigma_\nu\delta(\mathbf{x}-\mathbf{x}_{E\nu}).$$

□

At this point, we introduce the **adjoint of the linearized forward operator** $F'(v)^*$:

**Theorem 3.21.**
*Let $\tilde\sigma \in \mathbb{R}^N$. The adjoint of $F'(v)$ is given by*

$$F'(v)^*\tilde\sigma := \tilde\mu$$

*where $\tilde\mu \in C^{2,\alpha}(\Omega)^*$ is defined as*

$$\tilde\mu := \frac{\phi}{v}$$

*and $\phi \in C^{2,\alpha}(\Omega)^*$ solves the boundary value problem*

$$\begin{aligned}\Delta T\phi + \nabla T\nabla\phi = 0 \quad &in\ \Omega\\ \phi(\nabla T\cdot\boldsymbol{n}) = -\sum_{\nu=1}^N \tilde\sigma_\nu\delta(\mathbf{x}-\mathbf{x}_{E\nu}) \quad &on\ \partial\Omega.\end{aligned} \tag{3.15}$$

*Proof.* The statement is a consequence of Theorem 3.19, Lemma 3.20 and substituting $\sigma$ by $\mathcal{Q}^*\tilde{\sigma}$. □

**Remark 3.22.**

*a) Note that equation (3.13) as well as equation (3.15) is to be understood in the dual space setting tested against $\phi \in C^{2,\alpha}(\Omega)^*$ as in the proof of Lemma 3.19.*

*b) We do not prove the existence and uniqueness of the solution $\phi$. Again, it is a Hamilton-Jacobi equation for which the theory has already been well studied, see Remark 3.7. We recommend the work of Crandall and Lions [13]. For our thesis, we focus on the discretized case where we obtain convergence for our numerical solution by Section 3.4.2.*

## 3.4 Solving the inverse problem using Landweber iteration

Due to the previous sections, we are faced with a non-linear operator equation. The challenge of the resulting ill-posed problems can be addressed in the following ways: One possibility is to include prior knowledge to make the problem more stable while the other entails employing specific numerical approaches known as regularization techniques. The latter involves the point-wise approximation of the discontinuous inverse. The regularization theory is extensively comprehended and numerous regularization methods have been analyzed. A common method to solve non-linear inverse problems is the Landweber method. We give a short overview of the method referring to Kaltenbacher et al. [48].

In general, the Landweber method can be understood as a gradient descent method to minimize the least square functional

$$\phi^\delta(x) = \frac{1}{2}||F(x) - y^\delta||^2 \tag{3.16}$$

for a non-linear operator equation

$$F(x) = y$$

where $F : \mathcal{D}(F) \to Y$ with $\mathcal{D}(F) \subseteq X$ is a non-linear, continuous and Fréchet differentiable operator between Hilbert spaces $X$ and $Y$.

Considering that the exact data $y$ is rarely available, we represent the measured perturbed data as the noisy data $y^\delta$. We operate under the assumption that these noisy observations fulfill the inequation

$$||y^\delta - y|| \leq \delta.$$

The Landweber method is motivated by the normal equation and a fixed-point iteration: In the context of a linear problem, denoted here as

$$Kx = y,$$

where $K$ represents a linear operator, numerous iterative methods aim to approximate $K^+y$ where $K^+$ signifies the **Moore–Penrose inverse**, as explained in Rieder [78]. The methods rely on a transformation that converts the normal equation into an equivalent fixed-point equation. This transformation is exemplified as follows leading to a descent method:

$$x = x + K^*(y - Kx).$$

The idea of a descent method is to calculate the new iteration in the direction of the steepest descent. For the least square functional (3.16), its gradient $\partial_x \phi^\delta(x)$ with respect to $x$ is given by

$$\partial_x \phi^\delta = F'(x)^*(F(x) - y^\delta) \tag{3.17}$$

where $F'(x)^*$ denotes the adjoint of the Fréchet derivative, in our case the Gâteaux derivative, of $F$ in $x \in \mathcal{D}(F)$. With an initial guess $x_0^\delta$, we obtain the Landweber iteration

$$x_{k+1}^\delta = x_k^\delta - \omega F'(x_k^\delta)^*(F(x_k^\delta) - y^\delta), \quad k = 0, 1, 2, ... \tag{3.18}$$

as a descent method where $\omega > 0$ is a relaxation parameter. When the Landweber iteration is applied to exact data, meaning that we use $y$ instead of $y^\delta$, we denote the iterated solutions as $x_k$ instead of $x_k^\delta$.

Dealing with noisy data, it is necessary to incorporate a stopping criterion into the iteration process to act as a regularization method. The **discrepancy principle** is a suitable criterion to stop the iteration after $k^* = k^*(y^\delta, \delta)$ steps fulfilling

$$||y^\delta - F(x_{k_*}^\delta)|| \leq \tau\delta < ||y^\delta - F(x_k^\delta)||, \;\; 0 \leq k < k_*. \tag{3.19}$$

where $\tau > 1$.

**Remark 3.23** ([48], p. 5 et seqq.)**.**
*In the context of non-linear problems, iteration methods such as (3.18) typically do not converge globally. However, we can establish a guarantee of convergence within a limited local scope by imposing certain conditions on operator F. Given some assumptions, one can prove that the stopping index $k^*$ is finite. An appropriate choice is*

$$\tau > 2\frac{1+\eta}{1-2\eta} > 2$$

*where $\eta < \frac{1}{2}$ ensures the local cone condition*

$$||F(x) - F(\tilde{x}) - F'(x)(x - \tilde{x})|| \leq \eta||F(x) - F(\tilde{x})||$$

*for $x$, $\tilde{x} \in \mathcal{B}_{2\rho}(x_0) \subseteq \mathcal{D}(F)$. The notation $\mathcal{B}_{2\rho}(x_0)$ defines a closed ball of radius $2\rho$ around $x_0$.*

For more information about the non-linear Landweber iteration, we recommend the article of Hanke et al. [31] and the publication of Andreas Neubauer [72]. For our investigation, we focus on iteration (3.18).

### 3.4.1 Numerical solution of the forward operator using the Fast Marching Method

So far, the forward operator $F$ has been derived as a composition of the observation, the trace and the parameter-to-solution operator. Further, the Landweber method has been introduced. We need a numerical procedure to calculate the forward operator $F$ to obtain the residual

$$R^\delta = (F(v) - y^\delta)$$

which is an essential part of the gradient (3.17) and iteration (3.18). For this purpose, we solve the partial differential equation (3.1) with the initial condition (3.7) via a finite difference method, the so-called **Fast Marching Method (FMM)**. The idea of the FMM has been introduced by Sethian [82]. In this section, we describe the FMM based on a publication by Sethian [83]. Additional information can be found in the research of Chacon et al. [11]. In the following, the Fast Marching algorithm is presented to simulate the propagation of THz radiation and thus to solve the forward operator on a uniform Cartesian grid.

The FMM is an iterative procedure that, beginning at an initial value or a boundary condition, calculates the travel time that the THz radiation requires to reach a certain point $\mathbf{x} \in \Omega$. The differential equation is solved starting from the initial condition, i.e. from smaller values of the travel time $T$ to larger ones. Such methods are called **upwind schemes** [83]. In our case, the initial value models the source point at which the signal is emitted. To solve the eikonal equation numerically with the FMM, we have to discretize our domain. We set

$$\Omega = [-a, a] \times [-a, a],\ a \in \mathbb{R},$$

and are interested in a solution on a uniform Cartesian grid with step size $h$ that we define as

$$\Omega_h := \{(x_i, y_j)^\intercal \in \Omega : x_i = -a + (i-1)h,\ y_j = -a + (j-1)h,\ 1 \leq i, j \leq M\}.$$

The step size $h$ is given by

$$h := \frac{2a}{M-1}$$

where $M - 1$ is the number of equidistant steps in $x$- and $y$-direction and $M^2$ the number of pixels. The FMM also works for triangulated domains or unstructured meshes, see [52, 84].

We approximate the partial derivative at position $(x_i, y_j)^\intercal$ by the left-hand and the right-hand difference quotient and define

$$D_{i,j}T_x^+ := \frac{T(x_{i+1}, y_j) - T(x_i, y_j)}{h},\ \ 1 \leq i \leq M-1,\ 1 \leq j \leq M \tag{3.20}$$

$$D_{i,j}T_x^- := \frac{T(x_{i-1}, y_j) - T(x_i, y_j)}{-h},\ \ 2 \leq i \leq M,\ 1 \leq j \leq M \tag{3.21}$$

$$D_{i,j}T_y^+ := \frac{T(x_i, y_{j+1}) - T(x_i, y_j)}{h},\ \ 1 \leq i \leq M,\ 1 \leq j \leq M-1 \tag{3.22}$$

$$D_{i,j}T_y^- := \frac{T(x_i, y_{j-1}) - T(x_i, y_j)}{-h},\ \ 1 \leq i \leq M,\ 2 \leq j \leq M \tag{3.23}$$

We take the initial value problem (3.1) and an upwind scheme by Rouy and Tourin [80] to obtain

$$\left(\max(D_{i,j}T_x^-, -D_{i,j}T_x^+, 0)\right)^2 + \left(\max(D_{i,j}T_y^-, -D_{i,j}T_y^+, 0)\right)^2 = \frac{1}{v_{i,j}^2} \tag{3.24}$$

for $(x_i, y_j)^\intercal \in \Omega_h$ with

$$v_{i,j} = v(x_i, y_j).$$

Starting with the initial condition

$$T(x_i, y_j) = 0 \text{ for one } (i,j) \in \{1, ..., M\} \times \{1, ..., M\},$$

we calculate iteratively the travel time for all pixels. For the FMM in general, the starting point can be chosen arbitrarily. We notice that in our case the initial condition is assigned to a pixel on the boundary of our domain, that is

$$\partial\Omega_h := \{(x_i, y_j)^\intercal \in \Omega_h : i = 1 \vee j = 1 \vee i = M \vee j = M\}.$$

For all other pixels, we set the travel time to $+\infty$ meaning that it is currently unknown. In addition, we add some pixels to our domain by defining

$$T(x_0, y_j) := T(x_{M+1}, y_j) := T(x_i, y_0) := T(x_i, y_{M+1}) := +\infty, \;\; 1 \leq i, j \leq M,$$

such that the Definition (3.20) - (3.23) of the left-hand and right-hand difference quotients holds for all elements of the domain $\Omega_h$. The refractive index and thus the velocity $v_{i,j}$ is assumed to be known when solving the considered forward problem. The FMM searches for the smallest travel time value and then calculates the travel time for its neighboring pixels using the discretized eikonal equation (3.24). We want to calculate a solution such that the pixel values increase starting from the initial value.

The next theorem states that the discretized eikonal equation (3.24) is uniquely solvable given some preconditions. When calculating the travel time $T_{i,j}$ at position $(x_i, y_j)^\intercal$, we are confronted with several cases. We set $T_H := \min(T_{i-1,j}, T_{i+1,j})$ and $T_V := \min(T_{i,j-1}, T_{i,j+1})$ and consider $T \geq 0$.

**Theorem 3.24.**
*Let $\Omega_h$ be a Cartesian grid with equidistant step width $h$ and $M^2$ pixels. Further the travel times for the neighboring pixel*

$$\{(x_{i-1}, y_j), (x_i, y_{j-1}), (x_{i+1}, y_j), (x_i, y_{j+1})\}$$

*of the pixels $(x_i, y_j)$ are given and at least one travel time is finite. Then we can construct a unique solution $T_{i,j}$, $(i,j) \in \{1, ..., M\} \times \{1, ..., M\}$, such that*

*a) $T_{i,j} > \max\{T_H, T_V\}$ for $T_H \neq +\infty$ and $T_V \neq +\infty$ and $|T_H - T_V| \leq \frac{h}{v_{i,j}}$.*

*b) $T_{i,j} > T_H$ for $T_V = +\infty$.*

*c) $T_{i,j} > T_V$ for $T_H = +\infty$.*

*d)* $T_H > T_{i,j} > T_V$ *or* $T_V > T_{i,j} > T_H$ *for* $T_H \neq +\infty$ *and* $T_V \neq +\infty$ *and* $|T_H - T_V| > \frac{h}{v_{i,j}}$.

*Proof.* The situation that $T_H = \infty$ and $T_V = \infty$ at the same time is not possible due to the assumption that at least one neighboring pixel is finite.

a) Let $T_H \neq +\infty$ and $T_V \neq +\infty$ and $|T_H - T_V| \leq \frac{h}{v_{i,j}}$.
We construct a solution $T_{i,j} > \max\{T_H, T_V\}$. As a consequence we have

$$(T_{i,j} > T_{i-1,j}) \vee (T_{i,j} > T_{i+1,j}), \text{ respectively } (T_{i,j} > T_{i,j-1}) \vee (T_{i,j} > T_{i,j+1})$$

and it follows

$$\max(D_{i,j}T_x^-, -D_{i,j}T_x^+, 0) = \max\left(\frac{T_{i,j} - T_{i-1,j}}{h}, \frac{T_{i,j} - T_{i+1,j}}{h}, 0\right) > 0$$

and

$$\max(D_{i,j}T_y^-, -D_{i,j}T_y^+, 0) = \max\left(\frac{T_{i,j} - T_{i,j-1}}{h}, \frac{T_{i,j} - T_{i,j+1}}{h}, 0\right) > 0.$$

The discretized eikonal equation can be transformed to

$$(T_{i,j} - T_H)^2 + (T_{i,j} - T_V)^2 = \frac{h^2}{v_{i,j}^2}$$

yielding

$$T_{i,j} = \frac{T_H + T_V}{2} \pm \frac{1}{2}\sqrt{(T_H + T_V)^2 - 2\Big(T_H^2 + T_V^2 - \frac{h^2}{v_{i,j}^2}\Big)}.$$

This solution exists because of $|T_H - T_V| \leq \frac{h}{v_{i,j}}$. The solution is unique because

$$\begin{aligned} T_{i,j} &= \frac{T_H + T_V}{2} - \frac{1}{2}\sqrt{(T_H + T_V)^2 - 2\Big(T_H^2 + T_V^2 - \frac{h^2}{v_{i,j}^2}\Big)} \\ &\leq \max\{T_H, T_V\} - \frac{1}{2}\sqrt{(T_H + T_V)^2 - 2\Big(T_H^2 + T_V^2 - \frac{h^2}{v_{i,j}^2}\Big)} \\ &\leq \max\{T_H, T_V\}. \end{aligned}$$

leads to a contradiction to $T_{i,j} > \max\{T_H, T_V\}$.

b) We set $T_V = \infty$ and construct a unique solution $T_{i,j} > T_H$.
It follows that

$$\max(D_{i,j}T_y^-, -D_{i,j}T_y^+, 0) = 0$$

and

$$\max(D_{i,j}T_x^-, -D_{i,j}T_x^+, 0) \neq 0.$$

We transform the discrete differential equation (3.24) to

$$\begin{aligned} (T_{i,j} - T_H)^2 &= \frac{h^2}{v_{i,j}^2} \\ \Leftrightarrow T_{i,j} &= \frac{h}{v_{i,j}} + T_H. \end{aligned}$$

c) Same argumentation as in b).

d) It remains $T_H \neq +\infty$ and $T_V \neq +\infty$ and $|T_H - T_V| \geq \frac{h}{v_{i,j}}$. W.l.o.g we set $T_V > T_{i,j} > T_H$ and we use the same argumentation as in b) having

$$\max(D_{i,j}T_y^-, -D_{i,j}T_y^+, 0) = 0.$$

□

**Remark 3.25.**
*Theorem 3.24 states that we can calculate the travel time $T_{i,j}$ for all $1 \leq i, j \leq M$ iteratively in an upwind procedure if a source point condition is given. It can be shown that the solution converges to the viscosity solution [11]. To apply the forward operator $F$ completely, we have to restrict the solution to the boundary pixel of our domain $\Omega_h$ where the receivers are placed.*

Building on the previous theorem, we can now describe the Fast Marching algorithm. The FMM solves the discretized eikonal equation (3.24) starting with an initial value $T(x_0, y_0)$. The steps of Algorithm 3.26 are illustrated in Figure 3.1 for a better understanding.

**Algorithm 3.26.** *(FMM)*

**STEP 1:**
*Choose a finite start value $T_{i_0,j_0}$ with $(x_{i_0}, y_{i_0})^\intercal \in \partial\Omega_h$ and mark it as 'reached = 1'.*
*Set all other pixel values to $+\infty$ and mark them as 'unreached = 0'.*

**STEP 2:**
*Starting from the initial value, determine the four neighboring pixels and mark them as 'active =2'.*
*Calculate for each neighboring pixel with Theorem 3.24 the travel time. The neighbor pixel acts for $T_{i,j}$ in each case.*

**STEP 3:**
*From the active pixels, select the one with the smallest travel time and mark it as 'reached = 1'.*
*Repeat step 2, starting from this value. Skip a neighboring pixel if it is already marked as 'reached = 1'.*

**STEP 4:**
*Repeat step 3 until all pixels are marked as 'reached=1'.*

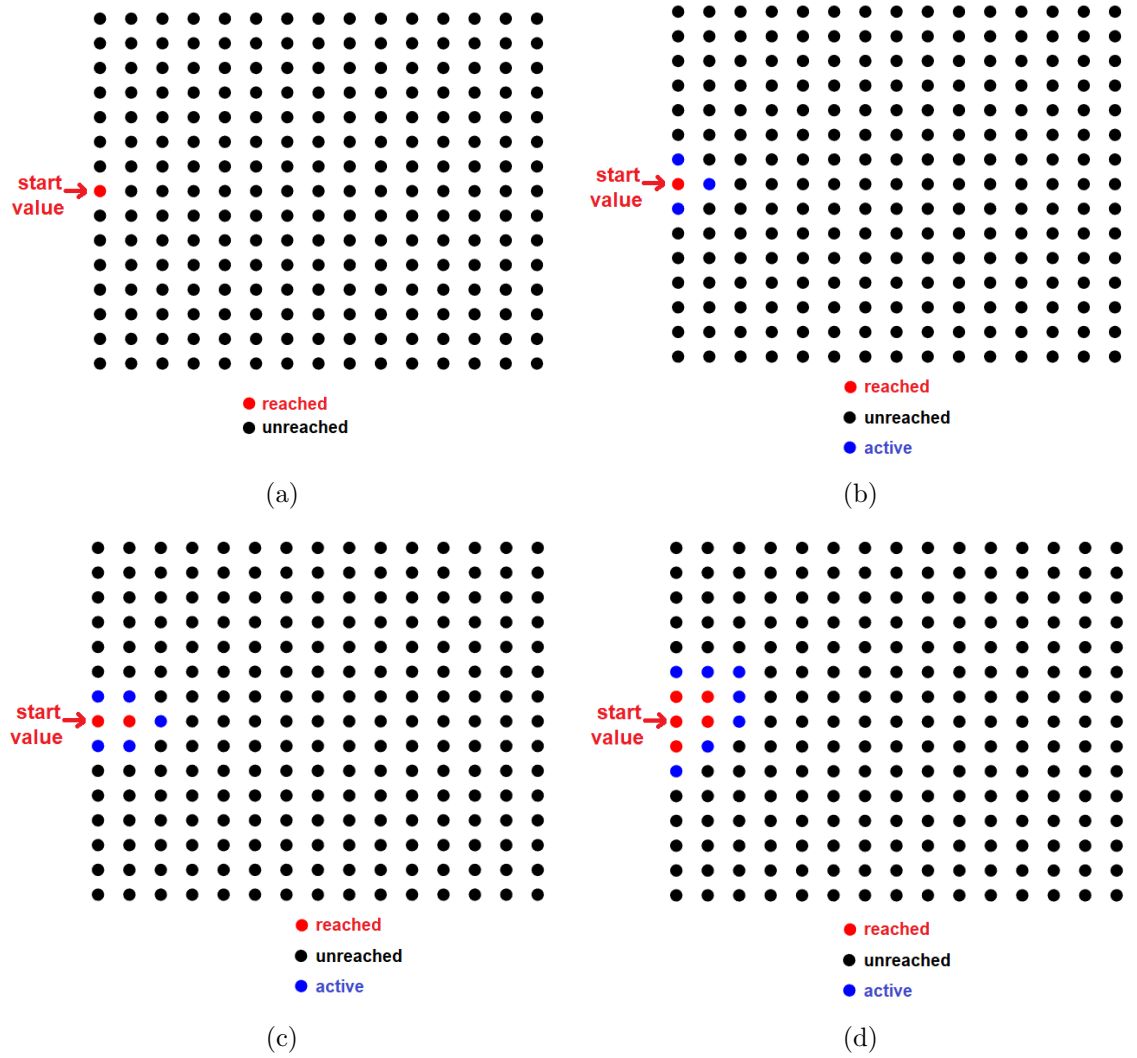



Figure 3.1: Visualization of Algorithm 3.26 showing Step 1 (a), Step 2 (b) and Step 3 (c) as well as the fifth iteration (d)

### 3.4.2 Numerical solution of the adjoint linearized problem

To implement iteration (3.18), a numerical solution of the adjoint of $F'$ is needed in addition to the numerical solution of $F$ derived in the last section. Consequently, we are interested in a numerical solution to the adjoint linearized problem (3.15), see Theorem 3.21. The considered partial differential equation is a first-order linear equation, again a Hamilton-Jacobi equation of the form

$$H(x, \phi(x), \nabla\phi(x)) = 0, \ x \in \Omega,$$

with Dirichlet boundary conditions. We solve the discretized version of this equation with a **fast sweeping** scheme motivated by Kao, Osher and Qian [51]. For a more detailed description of the underlying method, we refer to [60] that is a basis for this subsection.

The idea is to calculate the solution of a grid point $(x_i, y_j)^\intercal$ with the help of its neighbors, and then, to use a Gauss-Seidel-iteration by following the characteristics.

For this purpose, we divide our domain as in Section 3.4.1 and transform equation (3.15) to

$$(T_x\phi)_x + (T_y\phi)_y = 0 \tag{3.25}$$

respecting the product rule. We start with a discretization using the central difference on our equidistant grid and set for simplification

$$\begin{aligned}
a_{i+\frac{1}{2},j} &:= T_x(x_{i+\frac{1}{2}}, y_j) = \frac{T(x_{i+1}, y_j) - T(x_i, y_j)}{h},\\
a_{i-\frac{1}{2},j} &:= T_x(x_{i-\frac{1}{2}}, y_j) = \frac{T(x_i, y_j) - T(x_{i-1}, y_j)}{h},\\
b_{i,j+\frac{1}{2}} &:= T_y(x_i, y_{j+\frac{1}{2}}) = \frac{T(x_i, y_{j+1}) - T(x_i, y_j)}{h},\\
b_{i,j-\frac{1}{2}} &:= T_y(x_i, y_{j-\frac{1}{2}}) = \frac{T(x_i, y_j) - T(x_i, y_{j-1})}{h}.
\end{aligned}$$

We rewrite equation (3.25) using again the central difference for $(T_x\phi)_x$ and $(T_y\phi)_y$ to

$$\frac{1}{h}\left(a_{i+\frac{1}{2},j}\phi_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j}\phi_{i-\frac{1}{2},j} + b_{i,j+\frac{1}{2}}\phi_{i,j+\frac{1}{2}} - b_{i,j-\frac{1}{2}}\phi_{i,j-\frac{1}{2}}\right) = 0 \tag{3.26}$$

In the second step, we introduce the notations

$$\begin{aligned}
a^{\pm}_{i+\frac{1}{2},j} &= \frac{a_{i+\frac{1}{2},j} \pm |a_{i+\frac{1}{2},j}|}{2},\\
a^{\pm}_{i-\frac{1}{2},j} &= \frac{a_{i-\frac{1}{2},j} \pm |a_{i-\frac{1}{2},j}|}{2},\\
b^{\pm}_{i,j+\frac{1}{2}} &= \frac{b_{i,j+\frac{1}{2}} \pm |b_{i,j+\frac{1}{2}}|}{2},\\
b^{\pm}_{i,j-\frac{1}{2}} &= \frac{b_{i,j-\frac{1}{2}} \pm |b_{i,j-\frac{1}{2}}|}{2},
\end{aligned} \tag{3.27}$$

such that we can define the values $\phi_{i\pm\frac{1}{2},j}$ and $\phi_{i,j\pm\frac{1}{2}}$ according to the propagation of the radiation: If, for instance, the characteristic goes from left to right, we will define

$$\phi_{i+\frac{1}{2},j} := \phi_{i,j} \text{ for } a_{i+\frac{1}{2},j} > 0.$$

Analogously, we set

$$\phi_{i+\frac{1}{2},j} := \phi_{i+1,j} \text{ for } a_{i+\frac{1}{2},j} < 0$$

and define the values $\phi_{i,j\pm\frac{1}{2}}$ in a similar way. According to notation (3.27), we write equation (3.26) as follows:

$$\begin{aligned}
&(a^{+}_{i+\frac{1}{2},j}\phi_{i,j} + a^{-}_{i+\frac{1}{2},j}\phi_{i+1,j}) - (a^{+}_{i-\frac{1}{2},j}\phi_{i-1,j} + a^{-}_{i-\frac{1}{2},j}\phi_{i,j})\\
+&(b^{+}_{i,j+\frac{1}{2}}\phi_{i,j} + b^{-}_{i,j+\frac{1}{2}}\phi_{i,j+1}) - (b^{+}_{i,j-\frac{1}{2}}\phi_{i,j-1} + b^{-}_{i,j-\frac{1}{2}}\phi_{i,j}) = 0
\end{aligned}$$

A reformulation of the last equation results in an expression for $\phi_{i,j}$ in the interior of our domain:

$$\begin{aligned}
&(a^{+}_{i+\frac{1}{2},j} - a^{-}_{i-\frac{1}{2},j} + b^{+}_{i,j+\frac{1}{2}} - b^{-}_{i,j-\frac{1}{2}})\phi_{i,j}\\
=&a^{-}_{i+\frac{1}{2},j}\phi_{i+1,j} - a^{+}_{i-\frac{1}{2},j}\phi_{i-1,j} + b^{-}_{i,j+\frac{1}{2}}\phi_{i,j+1} - b^{+}_{i,j-\frac{1}{2}}\phi_{i,j-1}.
\end{aligned} \tag{3.28}$$

Further, we include the boundary conditions

$$\begin{aligned} \phi_{1,j} &= \frac{\tilde{\sigma}_{1,j}}{T_{x_{1,j}}}, \quad \phi_{M,j} = -\frac{\tilde{\sigma}_{M,j}}{T_{x_{M,j}}}, \quad j = 2, ..., M-1 \\ \phi_{i,1} &= \frac{\tilde{\sigma}_{i,1}}{T_{y_{i,1}}}, \quad \phi_{i,M} = -\frac{\tilde{\sigma}_{i,M}}{T_{y_{i,M}}}, \quad i = 2, ..., M-1 \end{aligned} \tag{3.29}$$

by using equation (3.15) and the outer normal vectors of a square, see Figure 3.2. For the edges, we set

$$\begin{aligned} \phi_{1,1} &= \frac{\tilde{\sigma}_{1,1}}{\sqrt{2}(T_{x_{1,1}} + T_{y_{1,1}})}, \\ \phi_{1,M} &= \frac{\tilde{\sigma}_{1,M}}{\sqrt{2}(T_{x_{1,M}} - T_{y_{1,M}})}, \\ \phi_{M,1} &= -\frac{\tilde{\sigma}_{M,1}}{\sqrt{2}(T_{x_{M,1}} - T_{y_{M,1}})}, \\ \phi_{M,M} &= -\frac{\tilde{\sigma}_{M,M}}{\sqrt{2}(T_{x_{M,M}} + T_{y_{M,M}})}. \end{aligned} \tag{3.30}$$

with $T_{x_{i,j}} := T_x(x_i, y_j)$ and $T_{y_{i,j}} := T_y(x_i, y_j)$.

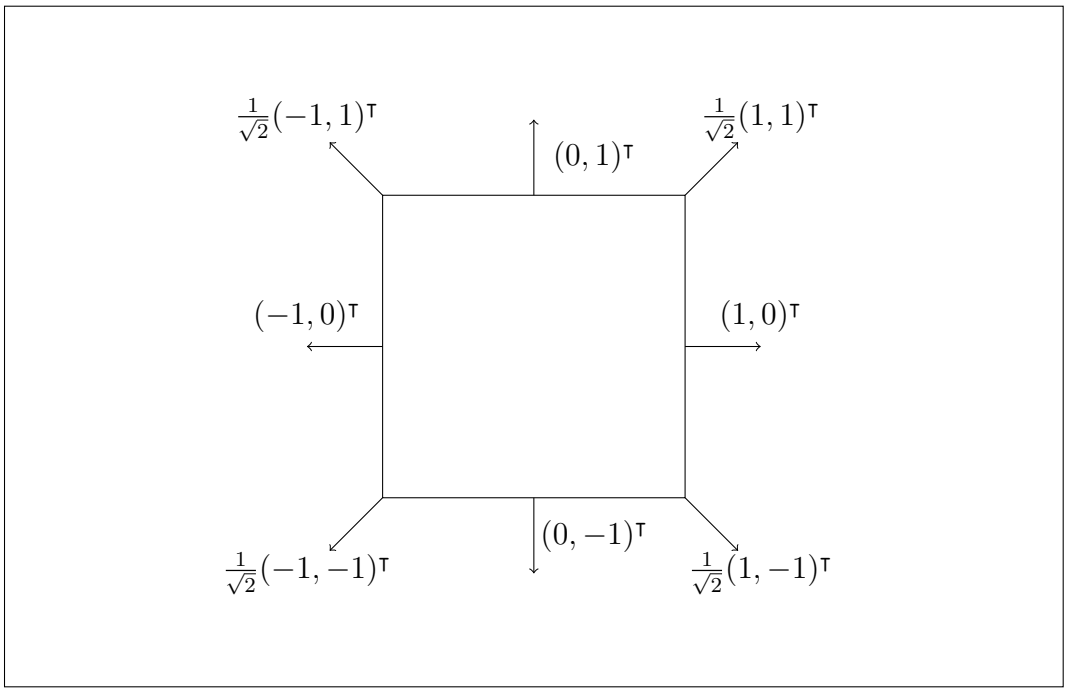



Figure 3.2: Outer normal vector of a square

Finally, we can use an iterative scheme and the boundary conditions to solve equation (3.15) via a fast sweeping iterative method, that is a Gauss-Seidel iteration using different directions. We obtain convergence because the resulting matrix of the linear system for the calculation of $\phi$ is irreducibly diagonally dominant, see [60]. The Gauss-Seidel method can be looked up in the book '*Introduction to Numerical Analysis*' by Stoer and Bulirsch [86]. We have an iteration mechanism characterized by the following structure

$$x^{(k+1)} = \Phi(x^{(k)})$$

to solve the linear system

$$Ax = b.$$

The value $x_i^{(k+1)} = \Phi_i(x^{(k)})$ is calculated via

$$x_i^{(k+1)} = -\frac{1}{a_{ii}}\left(\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} - b_i\right). \tag{3.31}$$

We obtain the following algorithm to solve the differential equation (3.25):

**Algorithm 3.27.** *(Fast sweeping method)*

**STEP 1:**
*Compute the boundary conditions (3.29) and (3.30). These values are fixed during the iteration. Set all other values* $\phi_{i,j}^{(old)}$ *to a larger value.*

**STEP 2:**
*Calculate* $\tilde{\phi}_{i,j}$ *with equation (3.28) using Gauss-Seidel iteration 3.31. Sweep from* $i = 1, ..., M$ *and* $j = 1, ..., M$ *and set* $\phi_{i,j}^{(new)} = \min\{\phi_{i,j}^{(old)}, \tilde{\phi}_{i,j}\}$.

**STEP 3:**
*Repeat Step 2 with the other three sweeping combinations:*

- $i = 1, ..., M$ *and* $j = M, ..., 1$
- $i = M, ..., 1$ *and* $j = M, ..., 1$
- $i = M, ..., 1$ *and* $j = 1, ..., M$.

**STEP 4:**
*Repeat Step 2 and 3 until*

$$||\phi_{i,j}^{(new)} - \phi_{i,j}^{(old)}|| \leq \epsilon.$$

### 3.4.3 Numerical reconstruction with the Landweber iteration

Finally, we have all the ingredients to implement the iterative Landweber method (3.18). In the further course of the dissertation, we work on a domain

$$\Omega = \{\mathbf{x} \in \mathbb{R}^2 : ||\mathbf{x}||_\infty \leq 0.15 \text{ m}\}$$

and use an equidistant grid of

$$M \times M = 57 \times 57$$

pixels with a resulting step size of

$$h = \frac{2 \cdot 15 \text{ cm}}{57 - 1} \approx 0.5357 \text{ cm}.$$

**Remark 3.28.**

*a) The domain is chosen based on the measurement setup in Chapter 5.*

*b) An increasing amount of pixels results in higher numerical costs because of the iterative PDE solvers introduced in the previous subsection. Further, the number of pixels influences the learning process of the neural network in Chapter 4.*

Our start pixel $(x_0^{(j)}, y_0^{(j)})^\intercal$, $j = 1, ..., J$, is rotated around the object of interest on the boundary $\partial\Omega_h$. The parameter $J$ denotes the total number of start positions. For each starting position, we obtain a forward operator $F^{(j)}$ and a resulting system of non-linear operator equations

$$F^{(j)}(v) = (T_v^{(j)}(\mathbf{x}_{E_1}), ..., T_v^{(j)}(\mathbf{x}_{E_N}))^\intercal = (y_1^{(j)}, ..., y_N^{(j)})^\intercal.$$

According to equation (3.17), we get a gradient for each starting condition defined by

$$g^{j,\delta}(v) := \partial_v \phi^{j,\delta}(v) = (F^{(j)})'(v)^*(F^{(j)}(v) - y^{j,\delta}). \tag{3.32}$$

Motivated by Wald [93], we define the gradient for each Landweber step as the average of all available gradients, such that each starting condition contributes the same amount to the solution of the underlying inverse problem. We set

$$g^\delta(v) := \frac{1}{J} \sum_{j=1}^{J} g^{j,\delta}(v) \tag{3.33}$$

and obtain the implemented Landweber iteration given by

$$v_{k+1}^\delta = v_k^\delta - \omega g^\delta(v_k^\delta), \quad k = 0, 1, 2, ... \tag{3.34}$$

We generate synthetic measurement data for a given velocity $v$ by the FMM introduced in the Subsection 3.4.1 and store them in a matrix $Y$. We have

$$Y := (F^{(j)}(v))_{j=1,...,J} = \begin{pmatrix} y_1^{(1)} & \cdots & y_1^{(J)} \\ \vdots & \ddots & \vdots \\ y_N^{(1)} & \cdots & y_N^{(J)} \end{pmatrix} \in \mathbb{R}^{N \times J}.$$

We solve the initial value problem

$$\begin{aligned} &|\nabla T_v(x, y)|^2 = \frac{1}{v^2(x, y)}, \quad \text{for all } (x, y) \in \Omega \\ &T_v(x_0^{(j)}, y_0^{(j)}) = 0 \end{aligned} \tag{3.35}$$

for a fixed velocity $v$ and each initial value $(x_0^{(j)}, y_0^{(j)})^\intercal$, $j = 1, ..., J$. The vector $F^{(j)}(v)$ is composed of the boundary values of the solution due to Definition 3.10 of the observation operator and Remark 3.11.

In the second step, we add a uniformly distributed noise to the synthetic data. We use a norm that is independent of the number of starting positions, i.e. we set

$$||Y^\delta - Y|| = \max_{j=1,...,J} ||y^{j,\delta} - y^j||_{\mathbb{R}^N} \leq \delta$$

where $Y^\delta$ is defined as

$$Y^\delta := \begin{pmatrix} y_1^{(1,\delta)} & \cdots & y_1^{(J,\delta)} \\ \vdots & \ddots & \vdots \\ y_N^{(1,\delta)} & \cdots & y_N^{(J,\delta)} \end{pmatrix} \in \mathbb{R}^{N\times J}.$$

By choosing random numbers from

$$I_\delta = [-0.001, 0.001],$$

we obtain

$$\max_{j=1,\dots,J} ||y^{j,\delta} - y^j||_{\mathbb{R}^N} \leq 0.001\sqrt{N} =: \delta.$$

Given by the number of edge pixels, we end up with a total number of $N = 224$ and thus

$$\delta \approx 0.015.$$

Finally, we use Landwebers method together with the average of all available gradients, see equation (3.34), to reconstruct the velocity $v^\delta$.

To test the algorithm, we start with a Gaussian function given by

$$v_1 : \Omega \to \mathbb{R}^+,\ v_1(x,y) = (3 - \mathrm{e}^{\frac{-x^2-y^2}{100}}) \cdot 10^8. \tag{3.36}$$

We define the test function $v_1$ in such a way that we are within a reasonable range of values for THz tomography corresponding to a refractive index $n_1 \in [1, 1.5]$. For our evaluations, we measure the amount of iteration steps $n$, the relative error

$$\epsilon_v = \frac{||v - v_n^\delta||_F}{||v||_F}$$

and the norm of the residual given by

$$||R_n^\delta|| = ||F(v_n^\delta) - y^\delta|| = \max_{j=1,\dots,J} ||F^{(j)}(v_n^\delta) - y^{j,\delta}||.$$

We use the latter value to apply the discrepancy principle as a stopping rule. By the norm $||\cdot||_F$ we understand the Frobenius norm

$$||v||_F = \sqrt{\sum_{i,j=1}^{N} |v_{i,j}|^2}.$$

The calculated values are presented in Table 3.1. The goal of this first reconstruction is to obtain an impression of the Landweber method and the underlying inverse problem. In Figure 3.3, we illustrate the reconstructed velocity $v_{1,k^*}^\delta$ in comparison to the ground truth velocity $v_1$ by visualizing both together with the reconstruction error. Further, we plot the relative error and the residual during the iterations in Figure 3.4. Note, that for each iteration step, we not only have to apply the FMM $J$-times, but also the fast sweeping method for the calculation of the adjoint.

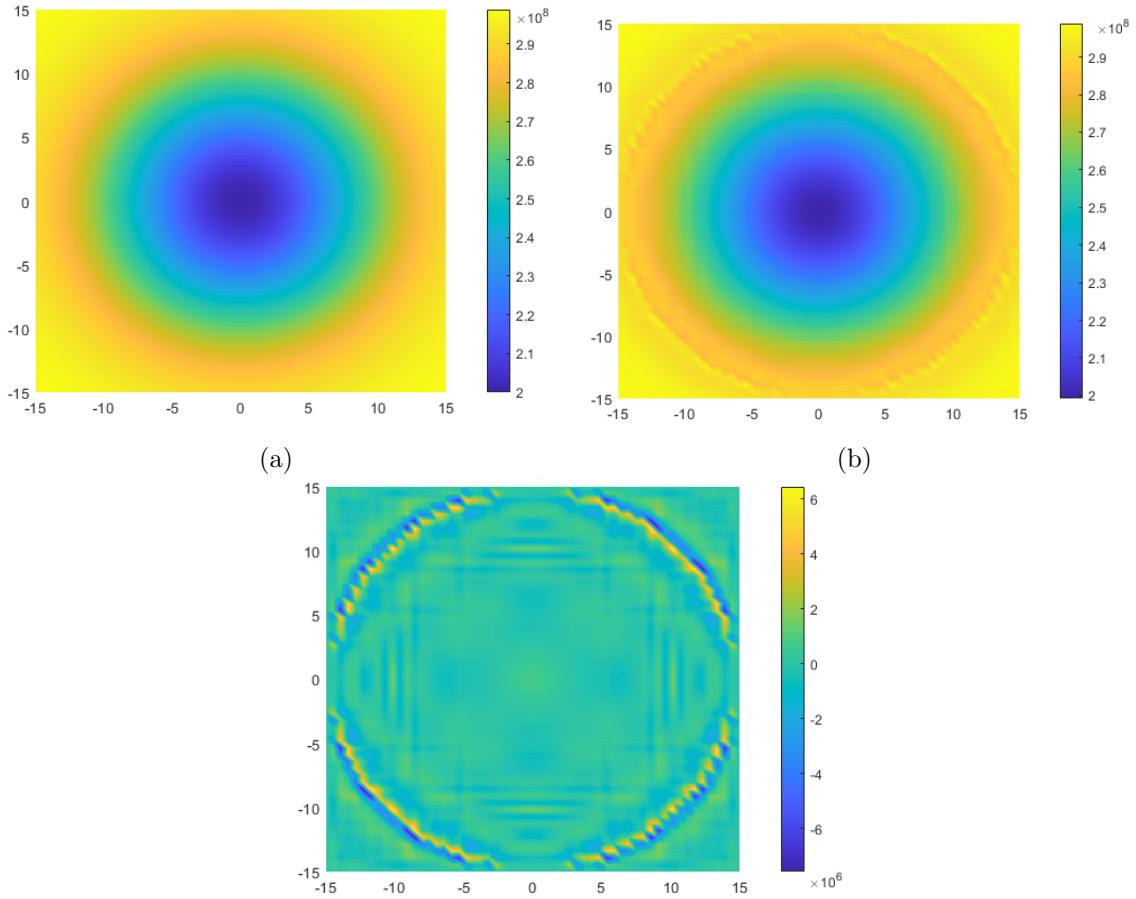



Figure 3.3: Reconstruction $v^{\delta}_{1,k^*}$ (b) of the ground truth velocity $v_1$ (a) with the Landweber iteration showing the error plot $v_1 - v^{\delta}_{1,k^*}$ (c)

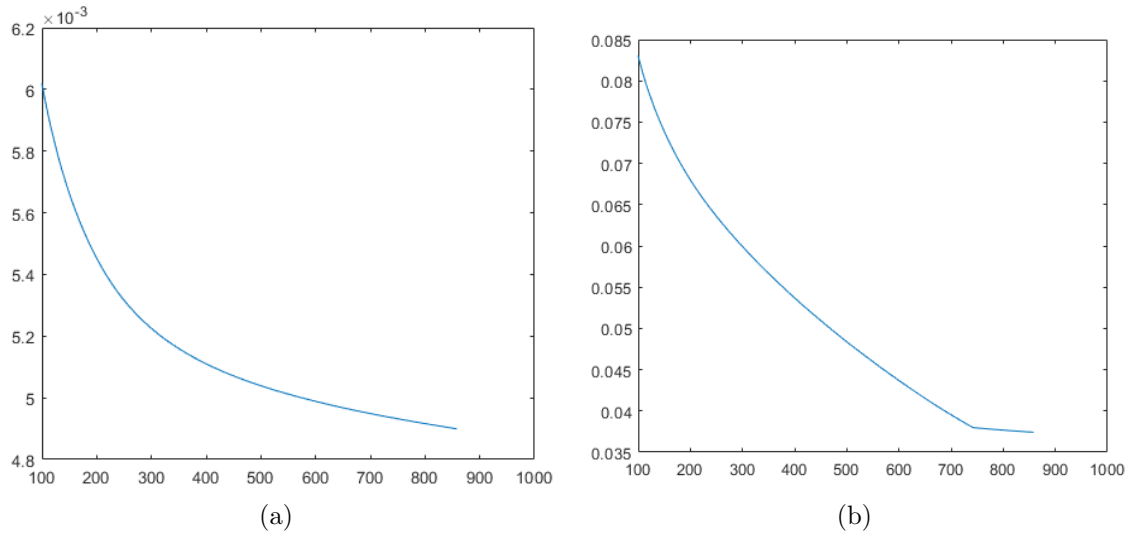



Figure 3.4: Relative error (a) and residual (b) during the Landweber iteration for the reconstruction of $v_1$

Table 3.1: Reconstruction results and Landweber parameters for the function $v_1$

| | |
|---|---|
| Amount of iterations $k^*$ | 858 |
| Relative error $\epsilon_v$ | 0.49 % |
| Residual $\|\|R^\delta_{k^*}\|\|$ | 0.037414 |
| Relaxation parameter $\omega$ | 1 |
| $\tau$ | 2.5 |

We continue with more complex functions to test the algorithm. We aim to approximate the refractive index of a plastic block by the function

$$v_2 : \Omega \to \mathbb{R}^+,\ v_2(x,y) = (3 - \mathrm{e}^{\frac{-x^8-y^8}{10^8}}) \cdot 10^8$$

and a plastic block with an air inclusion by the function

$$v_3 : \Omega \to \mathbb{R}^+,\ v_3(x,y) = (3 - \mathrm{e}^{\frac{-x^8-y^8}{10^8}} + \mathrm{e}^{-\frac{(x+2)^8+(y+2)^8}{300}}) \cdot 10^8. \tag{3.37}$$

The function

$$v_4 : \Omega \to \mathbb{R}^+,\ v_4(x,y) = (3 - \mathrm{e}^{\frac{-x^8-y^8}{10^8}} + \mathrm{e}^{-\frac{(x-3)^8+(y-3)^8}{300}} + \mathrm{e}^{-\frac{(x+3)^8+(y+3)^8}{100}}) \cdot 10^8.$$

describes a plastic block with two inclusions, one air inclusion and a second inclusion of different material. We still investigate continuous functions that approach the non-continuous transitions. We add noise to the synthetic data to obtain $Y^\delta$. The reconstruction results presented in Table 3.2 and illustrated in Figure 3.5 to 3.7 demonstrate good characteristics with a low relative error and robust qualitative and quantitative detection of inclusions. However, a significant drawback lies in the high numerical costs associated with the reconstruction process, lasting for hours. This limitation poses challenges in scenarios where efficiency is a critical factor as it is necessary for inline monitoring. In the upcoming chapter, we will explore ways to strike a balance between the achieved accuracy and the associated computational effort. This will involve optimizing the algorithm through the integration of neural networks taking into account energy efficiency.

Table 3.2: Reconstruction results and Landweber parameters for the functions $v_2$, $v_3$ and $v_4$

| | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|
| Amount of iterations $k^*$ | 3466 | 26594 | 120000 |
| Relative error $\epsilon_v$ | 1.41 % | 1.82 % | 2.13 % |
| Residual $\|\|R^\delta_{k^*}\|\|$ | 0.031428 | 0.014967 | 0.0181 |
| Relaxation parameter $\omega$ | 1 | 1 | 0.5 |
| $\tau$ | 2.1 | 1 | 1.25 |

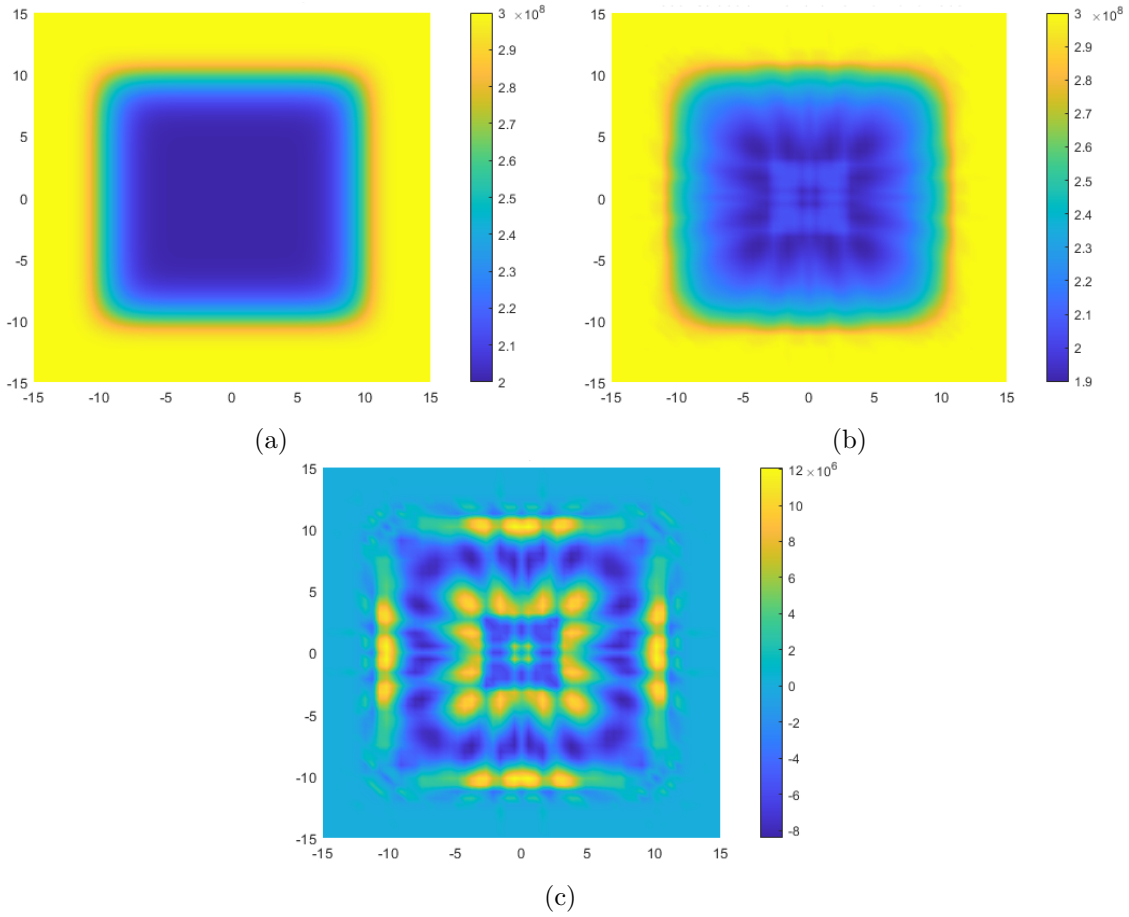
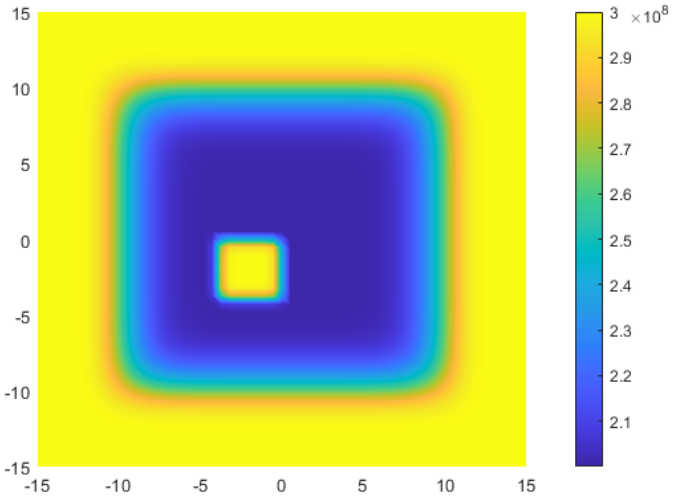
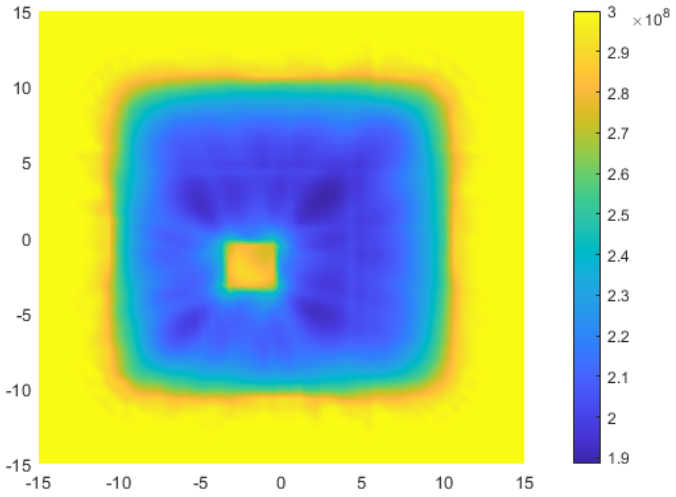


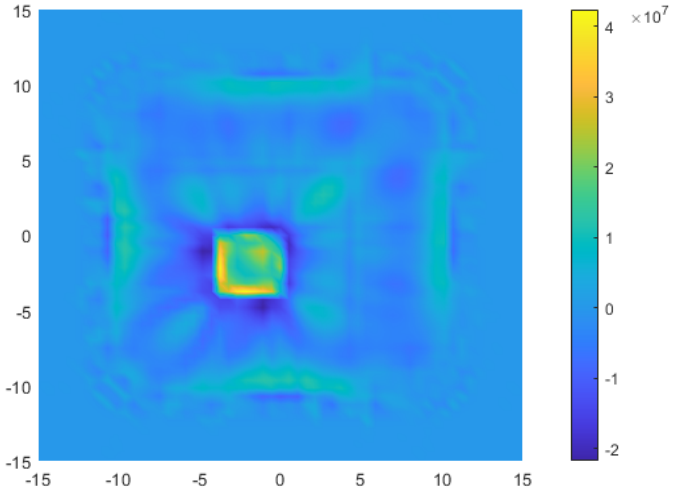

Figure 3.5: Reconstruction $v^{\delta}_{2,k^*}$ (b) of the ground truth velocity $v_2$ (a) with the Landweber iteration showing the error plot $v_2 - v^{\delta}_{2,k^*}$ (c)




(a)

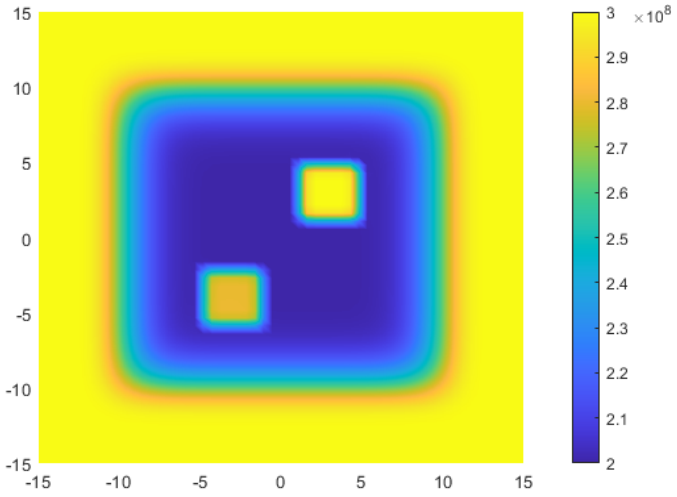



(b)

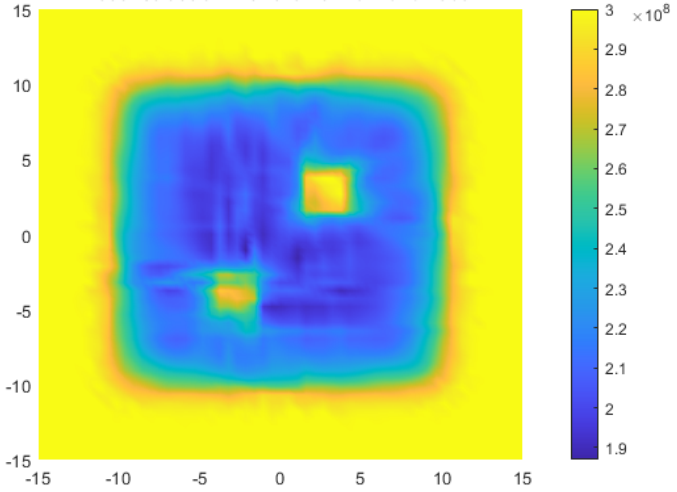



(c)

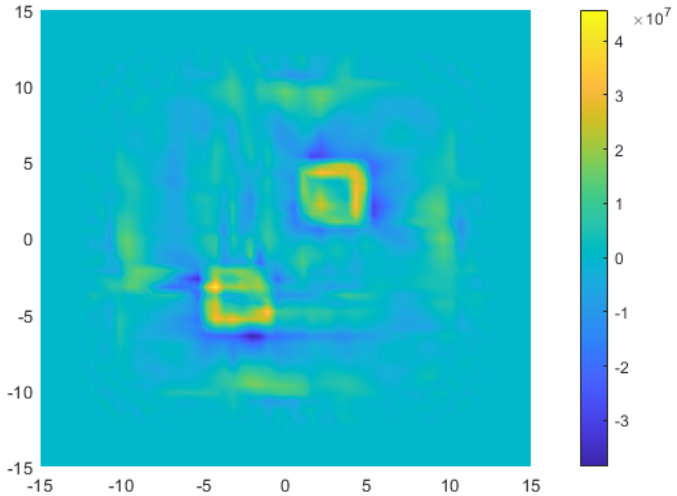
Figure 3.6: Reconstruction $v^{\delta}_{3,k^*}$ (b) of the ground truth velocity $v_3$ (a) with the Landweber iteration showing the error plot $v_3 - v^{\delta}_{3,k^*}$ (c)




(a)




(b)




(c)

Figure 3.7: Reconstruction $v^{\delta}_{4,k^*}$ (b) of the ground truth velocity $v_4$ (a) with the Landweber iteration showing the error plot $v_4 - v^{\delta}_{4,k^*}$ (c)

# 4 Learned Landweber iteration for terahertz tomography

In the previous chapter, we implemented the Landweber iteration to handle the non-linear inverse problem of THz tomography including the underlying physical model of the eikonal equation. We obtained promising results. The main problem of the presented algorithm concerns the computational time: For every Landweber step, we have to solve two partial differential equations for every angular position, one for the forward operator and one for the adjoint of the linearized forward operator. Since we have 180 angular positions and thousands of iterations within one reconstruction the numerical solutions of the partial differential equations have to be fast to avoid high numerical efforts.

In the next stage of this thesis, we aim to accelerate the given Landweber method by involving concepts of machine learning. More precisely, we replace the numerical solution of the forward operator $F$, which is currently performed via the FMM, by a CNN $\Phi_{\theta^*}(\cdot)$ such that the Landweber method (3.18) is transformed to a partially learned Landweber method

$$v_{i+1}^{\delta} = v_i^{\delta} - \omega F'(v_i^{\delta})^*(\Phi_{\theta^*}(v_i^{\delta}) - y^{\delta}).$$

We speed up the reconstruction process because of saving time for the evaluation of the forward operator. For this advantage, we accept a worse reconstruction accuracy and an upstream learning process to determine the optimal parameters $\theta^*$ of the network.

Afterwards, we opt to transform the network into a sparse one because the learning process consumes a significant amount of energy. This modification aims to conserve parameters, and consequently, investigates the energy reduction during the network's learning process. By setting weights to zero, it is not necessary to calculate its gradients during the optimization process of the network. We obtain a sparse learned Landweber method, where the learned optimal weights $\tilde{\theta}^*$ are sparse, meaning some entries are equal to zero. We gain a third version of Landweber's method:

$$v_{i+1}^{\delta} = v_i^{\delta} - \omega F'(v_i^{\delta})^*(\Phi_{\tilde{\theta}^*}(v_i^{\delta}) - y^{\delta}).$$

To achieve our goals, we first introduce the terms convolutional network and sparse network in the context of machine learning. Then, we learn the solution of the eikonal equation via synthetic supervised training data for different network structures and compare the computing time for the learned eikonal solver and the FMM. Finally, we include the dense learned forward operator in the Landweber method and compare

the results to those obtained from Chapter 3, and in the last step, to the ones of the sparse learned Landweber method. To generate sparsity, we include a regularization term in the learning process together with a threshold procedure.

## 4.1 Introduction to convolutional neural networks and sparse networks

In this section, we introduce the terms **convolutional neural networks (CNNs)** and **sparse neural networks (SNNs)**. We start with a short introduction to machine learning in general but skip the machine learning basics. Readers, that are not familiar with the fundamentals of machine learning are referred to the books of Aggarwal, Goodfellow and Nielsen [2, 25, 73] which we use as a guide in this section, if not otherwise stated. We assume that the terms **supervised** and **unsupervised** learning, **hyperparameters** in the context of neural networks, **perceptron**, **training** and **test data** are known. By the term **backpropagation** we understand a method to compute the gradients of the weights in the network using the chain rule to optimize the parameters. We speak of **underfitting** if the error in the training set is too high and we speak of **overfitting** if the difference between the training and the test error is too large. To counteract underfitting, the number of layers can be increased. In the case of overfitting, the training data set should be adjusted [34].

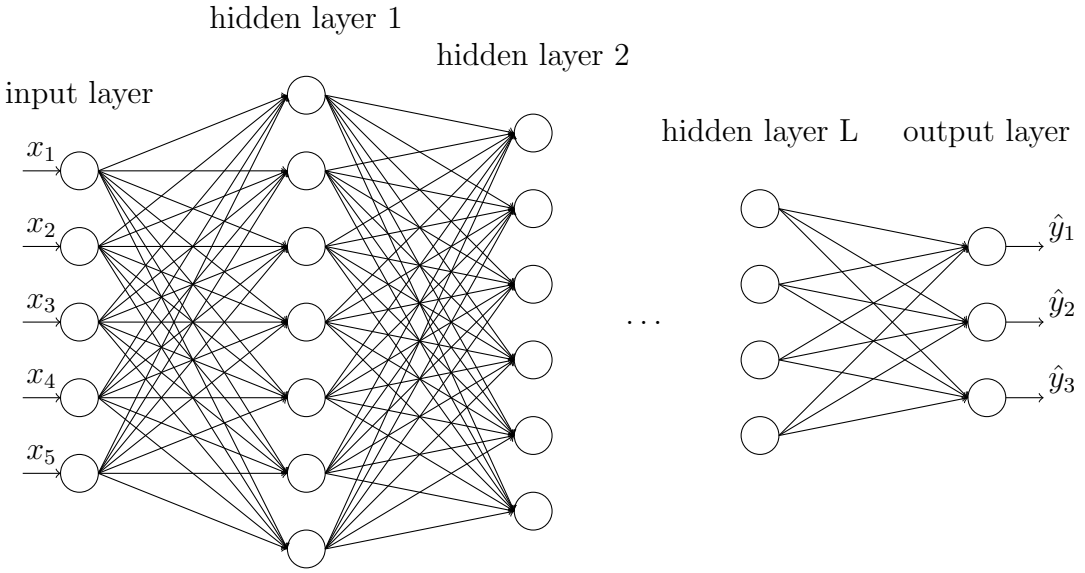



Figure 4.1: Example of a multilayer perceptron including $L$ hidden layers.

To introduce some notations, we observe Figure 4.1 which is an example of a fully connected **deep neural network**, also known as a **multilayer perceptron**. The network is composed of an **input layer**, some **hidden layers** and an **output layer**. Each layer consists of a different amount of **units** and the layers are interconnected with **weight matrices**. It is fully connected because every unit of the previous layer is connected with every unit of the next layer. Starting on the left side with the input layer, we take some input vector $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ denotes the **domain set** containing the objects we want to predict a certain behavior. The input vector is

multiplied with the weight matrice $W_1$ that connects the input layer with the first hidden layer. We add some bias $b_1$ and apply a non-linear **activation function** $\tau_1(\cdot)$ to obtain the values of the first hidden layer:

$$\varphi_1(\mathbf{x}) = \tau_1(W_1\mathbf{x} + b_1).$$

We repeat the procedure for the next hidden layers and end up in the output layer with an output vector $\hat{\mathbf{y}} \in \mathcal{Y}$ out of the **label set** $\mathcal{Y}$. The activation function $\tau : \mathbb{R}^m \to \mathbb{R}^m$ is defined component-wise by a non-linear operator $\tau^* : \mathbb{R} \to \mathbb{R}$, i.e.

$$\tau(\mathbf{z}) := \begin{pmatrix} \tau^*(z_1) \\ \vdots \\ \tau^*(z_m) \end{pmatrix}.$$

An activation function characterizes a particular threshold-like behavior observed in neurons. In other words, it signifies that a specific input threshold must be surpassed for a neuron to generate an output. In the field of neural networks, there are several activation functions, e.g.:

- Sigmoid: $\tau(z) = \frac{1}{1+e^{-z}}$.
- Rectified linear unit (ReLU): $\tau(z) = \max\{0, z\}$.
- Heaviside: $\tau(z) = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0. \end{cases}$

The three presented activation functions are illustrated in Figure 4.2. We see that the bias can shift the activation functions horizontally, and so, determines the threshold level.

We sum up the explanations and define a feedforward neural network for a regression process as follows:

**Definition 4.1** (**Feedforward neural network for regression**).
*Let $L \in \mathbb{N}$ denote the number of hidden layers and $u_i \in \mathbb{N}$ the number of units in layer $i$. Let $W_i \in \mathbb{R}^{u_i \times u_{i-1}}$ be the weight matrix that connects the $i$-1-th layer with the $i$-th layer and $b_i \in \mathbb{R}^{u_i}$ the corresponding bias. Further, let $\tau_i$ be a non-linear activation function, $\mathcal{X} := \mathbb{R}^{u_0}$ the domain set and $\mathcal{Y} := \mathbb{R}^{u_{L+1}}$ the label set. The values of layer $i$, $i = 1, ..., L$, are represented by $\varphi_i : \mathbb{R}^{u_{i-1}} \to \mathbb{R}^{u_i}$,*

$$\varphi_i(\tilde{\mathbf{x}}) = \varphi_{i,W_i,b_i}(\tilde{\mathbf{x}}) := \tau_i(W_i\tilde{\mathbf{x}} + b_i).$$

*Then $\Phi_L : \mathcal{X} \to \mathcal{Y}$ with*

$$\mathbf{x} \mapsto \Phi_L(\mathbf{x}) = W_{L+1}\Big(\varphi_L \circ \varphi_{L-1} \circ ... \circ \varphi_1(\mathbf{x})\Big) + b_{L+1} = \hat{\mathbf{y}}$$

*is a feedforward neural network for regression with $L$ hidden layers.*

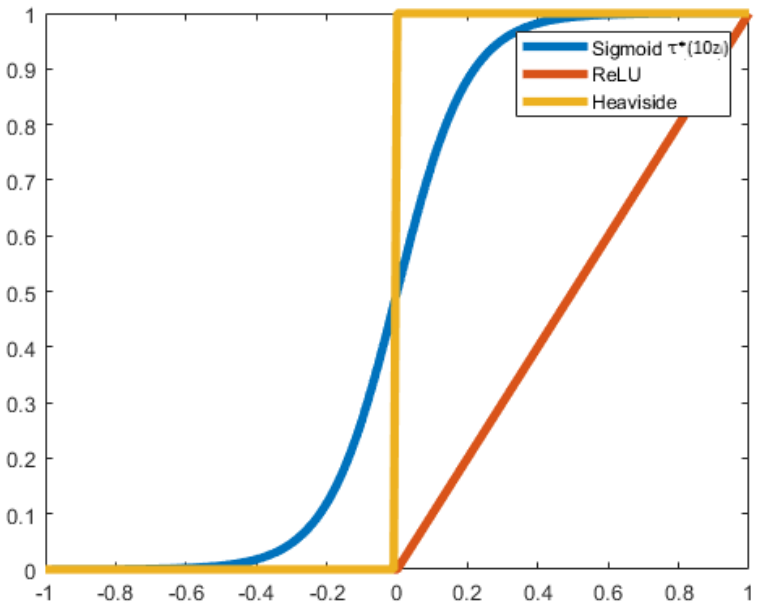



Figure 4.2: Activation functions

Note, that we neglect the activation function for the output layer since we are in a regression analysis. The network under consideration is a feedforward one, that compared to recurrent networks means that information only proceeds in one direction. For our investigations, we only focus on finite-dimensional domain and label sets $\mathcal{X}$ and $\mathcal{Y}$.

Given a supervised training set $\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(N_T)}, \mathbf{y}^{(N_T)})\} \subseteq \mathbb{R}^{u_0} \times \mathbb{R}^{u_{L+1}}$ with $N_T \in \mathbb{N}$ training data, we are interested in the weights and bias that minimize the error between the output of the network $\hat{\mathbf{y}}^{(i)}$ and the correct output $\mathbf{y}^{(i)}$ for all $i = 1, ..., N_T$. We introduce the cost function to measure the distance.

**Notation 4.2.**
*In the following, we comprise the parameters of the $i$-th layer in $\theta_i = (W_i, b_i)$. We denote by $\Theta_i$ the parameter space parameterizing the $i$-th layer. We define by $\Theta := \Theta_1 \times ... \times \Theta_{L+1}$ the parameter space of all weights $\theta = (\theta_1, ..., \theta_{L+1})$ in the network. Further, we write $\mathbf{X}_T = (\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N_T)})$ for all input training data, $\mathbf{Y}_T = (\mathbf{y}^{(1)}, ..., \mathbf{y}^{(N_T)})$ for all output training data and $\Phi_\theta$ for a network with the parameters $\theta$.*

**Definition 4.3 (Cost function).**
*Let $\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(N_T)}, \mathbf{y}^{(N_T)})\} \subseteq \mathbb{R}^{u_0} \times \mathbb{R}^{u_{L+1}}$ be a training set of $N_T$ training data. The cost function*

$$J : \Theta \times \mathbb{R}^{u_0 \times N_T} \times \mathbb{R}^{u_{L+1} \times N_T} \to \mathbb{R}$$

*is defined as*

$$J(\theta, \mathbf{X}_T, \mathbf{Y}_T) := \frac{1}{N_T} \sum_{i=1}^{N_T} l\left(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}\right) = \frac{1}{N} \sum_{i=1}^{N_T} l\left(\Phi_\theta(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}\right), \tag{4.1}$$

*that measures the average distance between the output of the network* $\hat{\mathbf{y}}$ *and the correct data* $\mathbf{y}$*. The function* $l : \mathbb{R}^{u_0} \times \mathbb{R}^{u_{L+1}} \to \mathbb{R}$ *is the so-called* ***loss function*** *given, for instance, by the mean squared error.*

The task of training a neural network is to minimize the cost function with respect to the parameters $\theta \in \Theta$, i.e, find $\theta^* \in \Theta$ with

$$\theta^* := \arg\min_{\theta \in \Theta} J(\theta, \mathbf{X}_T, \mathbf{Y}_T). \tag{4.2}$$

Most machine learning models are biologically inspired. They try to reproduce the functioning of the nervous system and the brain. The multilayer perceptron, see Figure 4.1, is based on the structure and interaction of nerve cells. Considering a nerve cell, we have a stimulus captured by dendrites. The stimulus can be identified as the input of the network. If a threshold potential is reached, an action potential will be triggered. This is simulated by the activation function acting in each unit. The signal is transmitted via the axon to the synapses where it is transferred to the next cell by neurotransmitters. This process is represented by the weighting matrices and the bias. The nervous system consists of an interaction of all nerve cells. Stimuli are received and processed which ultimately lead to a reaction illustrated by the output of the network. Consequently, a deep neural network consists not only of one perceptron but of an interaction of several layers. By increasing the number of layers and augmenting the units within each layer, a deep neural network has the capacity to encode functions and stimuli of growing complexity.

The idea of CNNs originates from a neuroscientific experiment about the mammalian vision system. Hubel et al. investigated the activity of neurons in cats in response to images [38, 39]. Unlike in our work, convolutional networks are mostly used for classification tasks and gained increasing interest by winning the '*ImageNet*' contest named **Large Scale Visual Recognition Challenge** (LSVRC). The network **AlexNet** was the first deep neural network producing an error lower than 20%. Today's CNNs generate an error less than 5% which is the human error rate for this task [3].

When talking about convolutional networks, we deal with a special kind of feedforward network to do learning processes with data that has a grid-like topology. Mostly, we make predictions for two-dimensional or three-dimensional images. As the name implies, the networks involve a mathematical operator on two functions with a real-valued argument called convolution.

**Definition 4.4** (**Convolution**, [23], p. 100)**.**
*The convolution* $f * g$ *of two functions* $f, g : \mathbb{R}^n \to \mathbb{C}$ *is defined as*

$$(f * g)(\mathbf{x}) := \int_{\mathbb{R}^n} f(\mathbf{y}) g(\mathbf{x} - \mathbf{y})\, dy. \tag{4.3}$$

In general, the convolution can be considered as a weighted average of a function, where the function $f$ is weighted by the function $g$. The operator used in CNNs does not correspond exactly to this definition. Usually, we deal with discrete data. We need the discrete convolution operator:

**Definition 4.5** (**Discrete convolution**, [23], p. 52)**.**
*For discrete two-dimensional real-valued functions $I$ and $H$, we define the discrete convolution operator $I * H$ as*

$$(I * H)(u, v) := \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(u - i, v - j) \cdot H(i, j). \tag{4.4}$$

**Remark 4.6.**
*The convolution is commutative. We have*

$$(I * H)(u, v) = (H * I)(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) \cdot H(u - i, v - j).$$

Convolutional networks use an operator, the linear filter, that is given in the next definition. All properties of the (discrete) convolution can be transferred to the linear filtering, see Appendix B.9.

**Definition 4.7** (**Linear filter**, [9], p. 96)**.**
*Let $I$ and $H$ be discrete real-valued functions and $R$ a filter region, such that $H(i, j) = 0$ outside of $R$. The linear filter $I \hat{*} H$ is defined as*

$$(I \hat{*} H)(u, v) := \sum_{(i,j) \in R} I(u + i, v + j) \cdot H(i, j). \tag{4.5}$$

In a CNN, the first argument of Definition 4.7 is the **input** and the second argument is the **kernel**. The output is stated as the **feature map**. We often work with multidimensional arrays of data and parameters. So, we result in a tensor.

**Remark 4.8.**

*a) The linear filter represents a discrete convolution with the kernel $H^*$ creating from vertical and horizontal mirroring of $H$, see Appendix B.8.*

*b) When we speak of convolution in the following, we understand it as Definition 4.7 and use the operator $*$ instead of $\hat{*}$.*

*c) When we apply several kernels to a two-dimensional input, we obtain a multidimensional feature map, see Figure 4.4. In the next convolutional layer, we consequently have to use a multidimensional kernel. The convolution operator then is amplified to the corresponding dimension, see Definition 4.9.*

In summary, we can define a convolutional layer as follows:

**Definition 4.9** (**Convolutional layer**)**.**
*Let $d \in \mathbb{N}$ be the dimension of the input, $n_f \in \mathbb{N}$ the number of resulting feature maps, $X \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ the input and $W \in \mathbb{R}^{q_1 \times \ldots \times q_d \times n_f}$ the multidimensional convolution kernel. The convolution operator*

$$* : \mathbb{R}^{n_1 \times \ldots \times n_d} \times \mathbb{R}^{q_1 \times \ldots \times q_d \times n_f} \to \mathbb{R}^{(n_1 - q_1 + 1) \times \cdots \times (n_d - q_d + 1) \times n_f}$$

*is given by*

$$X * W = \left( \sum_{k_1=1}^{q_1} \cdots \sum_{k_d=1}^{q_d} w^{[j]}_{k_1 \cdots k_d} x_{l_1 + k_1 - 1 \cdots l_d + k_d - 1} \right)_{l_1 = 1, \ldots, n_1 - q_1 + 1, \cdots, l_d = 1, \ldots, n_d - q_d + 1}^{j = 1, \ldots, n_f}. \tag{4.6}$$

*Then* $\Phi_{W,B} : \mathbb{R}^{n_1 \times \ldots \times n_d} \to \mathbb{R}^{(n_1-q_1+1)\times\cdots\times(n_d-q_d+1)\times n_f}$

$$X \mapsto \tau(X * W + B)$$

*with bias* $B \in \mathbb{R}^{(n_1-q_1+1)\times\cdots\times(n_d-q_d+1)\times n_f}$ *and activation function*

$$\tau : \mathbb{R}^{(n_1-q_1+1)\times\cdots\times(n_d-q_d+1)\times n_f} \to \mathbb{R}^{(n_1-q_1+1)\times\cdots\times(n_d-q_d+1)\times n_f}$$

*defined component-wise is called a convolutional layer.*

The following algorithm describes the application of the convolution operator with the kernel $W$ applied to a two-dimensional input and generating a multidimensional feature map. A visual example generating a one-dimensional feature map is given in Figure 4.3. The considered output pixel $y_{11}$ is given by

$$\begin{aligned} y_{11} =& w_{11}x_{11} + w_{12}x_{12} + w_{13}x_{13} + w_{21}x_{21} + w_{22}x_{22} + w_{23}x_{23} \\ &+ w_{31}x_{31} + w_{32}x_{32} + w_{33}x_{33}. \end{aligned} \tag{4.7}$$

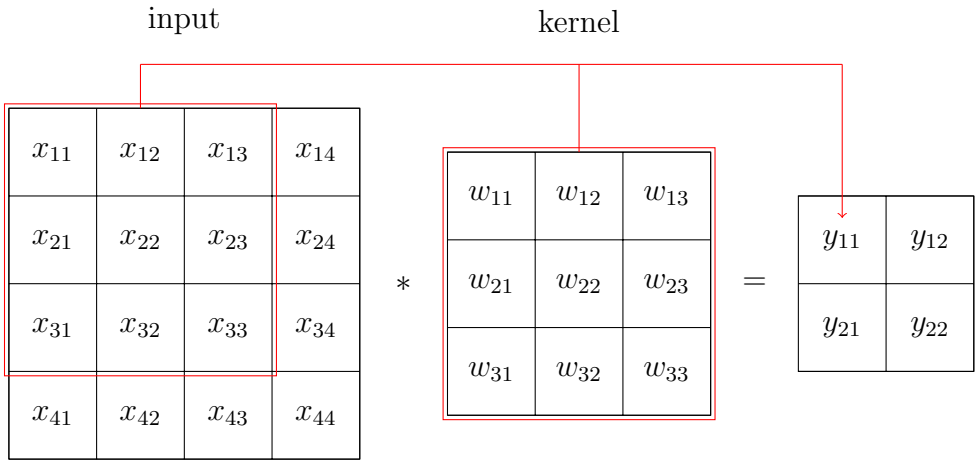



Figure 4.3: Visual Example of Algorithm 4.10 with $n_1 = n_2 = 4$, $j = 1$, $q_1 = q_2 = 3$ and $l_1 = l_2 = 1$ with the considered pixel $y_{11}$ given by equation (4.7)

**Algorithm 4.10.** ***(Discrete convolution)***

**INPUT/OUTPUT:**

*The kernels* $w^{[j]}_{k_1k_2}$, $k_1 = 1, ..., q_1$, $k_2 = 1, ..., q_2$, $j = 1, \cdots, n_f$, *and the input* $x_{zr}$, $z = 1, ..., n_1$, $r = 1, ..., n_2$ *are given.*
*We generate an output* $y^{[j]}_{l_1l_2}$, $l_1 = 1, ..., n_1 - q_1 + 1$, $l_2 = 1, ..., n_2 - q_2 + 1$.

**STEP 1:**

*Set* $l_1 = l_2 = 1$. *The kernel* $W^{[1]}$ *is positioned over the input so that entry* $w^{[1]}_{11}$ *falls on entry* $x_{l_1l_2}$ *of the input.*

**STEP 2:**
*The values of* $w^{[1]}_{k_1k_2}$, $k_1 = 1, ..., q_1$, $k_2 = 1, ..., q_2$, *are multiplied with the underlying pixels*

$$x_{l_1+k_1-1,l_2+k_2-1} = x_{1+k_1-1,1+k_2-1} = x_{k_1k_2}$$

*and summed up.*

**STEP 3:**
*The sum is saved in a new point of the feature map* $y^{[1]}_{11}$.

**STEP 4:**
*Repeat Step 1 to Step 3 for all input entries* $x_{l_1l_2}$ *with* $l_1 = 1, ..., n_1 - q_1 + 1$ *and* $l_2 = 1, ..., n_2 - q_2 + 1$.

**STEP 5:**
*Repeat Step 1 to Step 4 for all kernels* $W^{[j]}$.

At this point, we can summarize that CNNs are

> simple neural networks that use convolution in place of general matrix multiplication in at least one of their layers ([25], p. 330).

In such networks, the convolution operator is often combined with other functions. A layer consists typically of three steps: first, a convolution, then an activation function, see Definition 4.9, and finally a **pooling operator**, where the size of the grid is reduced and the output of the activation function at a specific location is substituted by a representation derived from the surrounding outputs. For example, one can use the **max-pooling** operator where one takes the maximum out of a rectangular neighborhood each time to reduce the size. Some commonly used pooling functions involve computing the average within a rectangular neighborhood, calculating the $\ell^2$-norm within such a neighborhood or performing a weighted average that takes into account the distance from the central pixel. Further, it is possible to use a fully connected layer following a convolutional layer. Then, the given tensor is flattened into a vector and the fully connected layer is applied to generate, for example the output of the network, see Figure 4.4.

**Remark 4.11.**
*There are two processes that we can use within a convolution:*

*a) When we skip some positions in the convolution process, we are talking about the term* ***stride****. Using, for example a stride of two, we shift our overlapping kernel each time by two entries. By doing this, we reduce the computational cost and result in a down-sampling of the output. Mathematically, we can transform equation (4.6) to*

$$X *_s W = \left( \sum_{k_1=1}^{q_1} \cdots \sum_{k_d=1}^{q_d} w^{[j]}_{k_1,\cdots,k_d} x_{l_1+k_1-1,\cdots,l_d+k_d-1} \right)^{j=1,...,n_f}_{l_i \in I_i} \tag{4.8}$$

*using* $I_i = \{sj + 1 | j \in \mathbb{N}_0, sj + 1 \leq n_i - q_i + 1\}$.
*Unless otherwise stated, we use a stride of* $s = 1$.

*b) Another process, that is often used within a convolutional layer, is* ***padding*** *to change the input size and to give more weight to the edge pixel of the input. Normally, the periphery of the layer will receive a lower level of representation in contrast to the central pixels by using the convolution operator. Padding is done by adding some pixels around the borders of the input, respectively, the output of a hidden layer before using the convolution. The values of the added pixels are set to zero. By a padding of* $p$ *the feature map is amplified by* $p$ *in each dimension. The computational costs increase.*
*If not otherwise specified, we will use a padding of* $p = 0$.

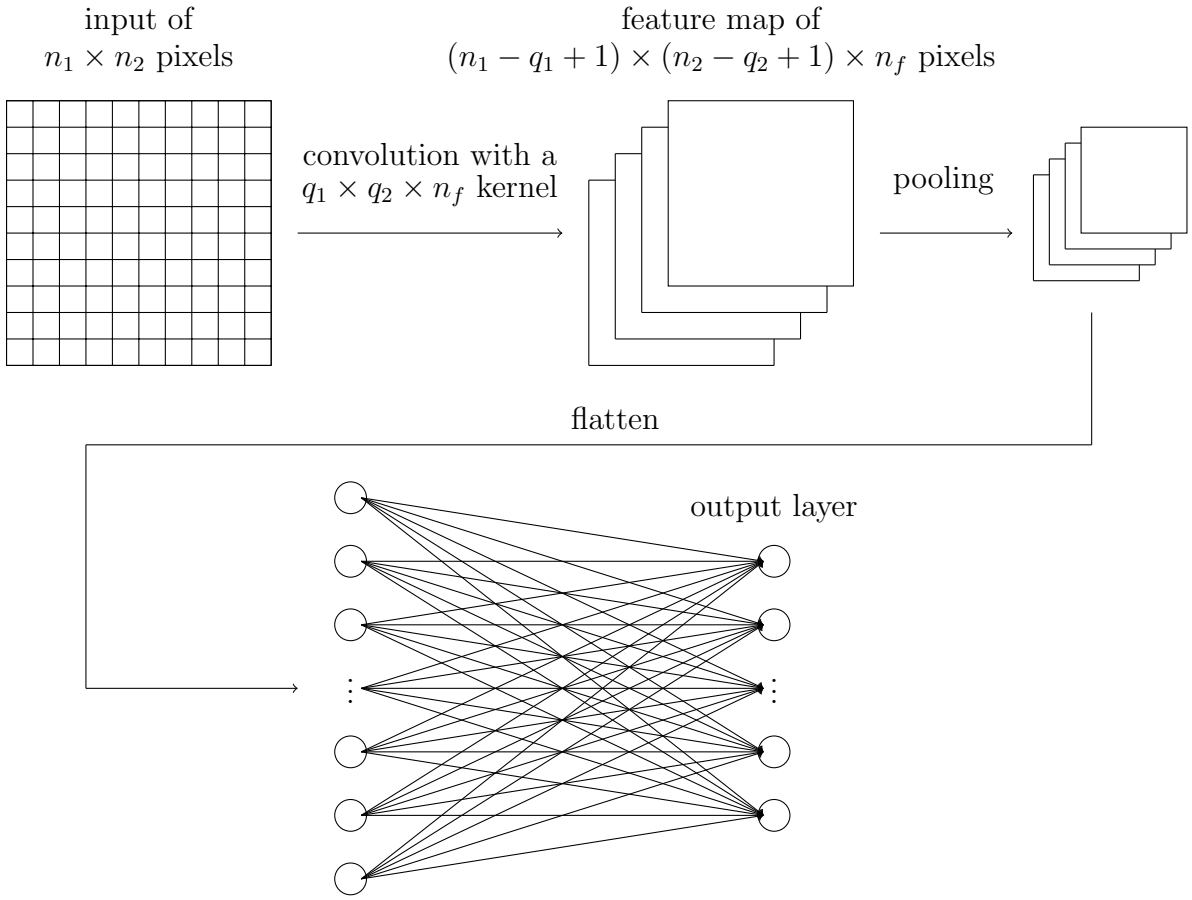



Figure 4.4: Example of a CNN including convolution, pooling, flattening and a fully connected layer

When we build a deep neural network, we often result in complex architectures with a high amount of parameters. The approximation theorems show that we can approximate $L^p$-functions with multilayer feedforward networks if we use enough hidden units and a bounded and non-constant activation function. Note that the ReLU is unbounded. We can show that it is discriminatory, and thus, we obtain the universal approximation theorem, see Appendix B.4. For more information about the universal approximation theorems, we refer to [14, 26, 36, 59, 98].

Training a deep neural network consequently leads to solve high-dimensional optimization problems because the weights should minimize the cost functional given by equation (4.2). To reduce the computational costs, the aim is to reduce the number of parameters which results in sparsity generating regularization. One wants to set parameters that have a low influence on the performance of the network to zero in order to save time in the optimization process. If a parameter is equal to zero, connections will be suppressed and consequently less derivatives have to be calculated for the optimization step. These efforts are additionally bolstered by the recently proposed '*lottery ticket hypothesis*' which suggests that within dense, feedforward networks, there exist sub-networks with fewer neurons that, when trained in isolation, can attain the same level of test accuracy as the original network [22]. Sparsity generating regularization has already proven itself in the context of inverse problems, see e.g. [43].

To induce sparsity, a regularization term is introduced into the cost function, effectively acting as a penalty. This regularization term aims to reduce the magnitudes of weights and connections, thereby promoting a sparse architecture and guaranteeing a faster optimization process. Thus,

$$\arg\min_{\theta\in\Theta}\Big(J(\theta,\mathbf{X}_T,\mathbf{Y}_T)+\alpha R(\theta)\Big) \tag{4.9}$$

is solved with the functional $R(\theta)\in\mathbb{R}$ that leads to a sparse minimizer. The regularization parameter $\alpha\in\mathbb{R}_+$ controls the influence of the functional on the weights. By using an iterative optimizer more and more weights are removed from the network. The idea of sparsity is illustrated in Figure 4.5. On the left side, we see the dense architecture. Through the optimization process, connections are removed and we result in the sparse setting on the right side. The more weights equal to zero exist, the less complex the model is.

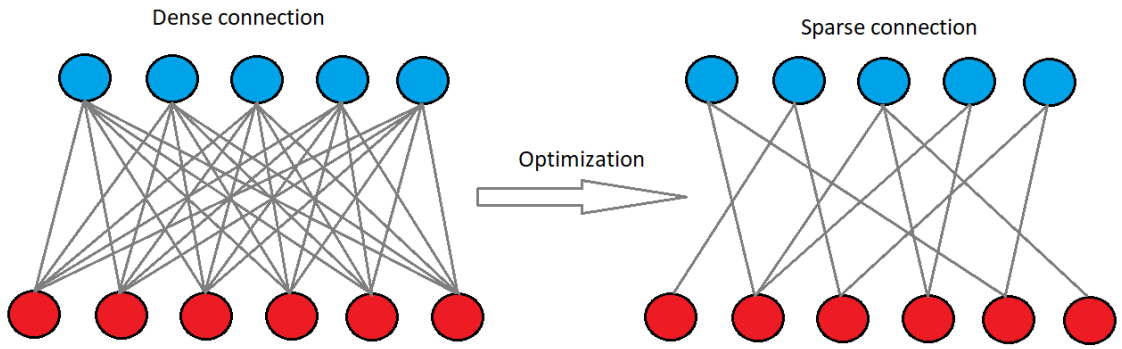



Figure 4.5: Idea of sparsity in the context of neural networks

Another objective of deleting parameters is to prevent overfitting. High values of the weights are suppressed by introducing a penalty term into the minimization process. We aim to find an equilibrium between minimizing the cost function $J(\theta,\mathbf{X}_T,\mathbf{Y}_T)$ and controlling the penalty term $R(\theta)$, where the regularization parameter $\alpha$ controls the influence of the penalty term. This balance helps the model to generalize effectively to unfamiliar data, enhancing its robustness and reliability. One prominent regularization method is the $\ell^1$-regularization, also known as **Lasso** regularization [90]. As the name suggests, the $\ell^1$-norm is used, such that we include the absolute

values of our network parameters $\theta$. So, large coefficients are strongly penalized. We set

$$R_1(\theta) = ||\theta||_1 = \sum_{i=1}^{L+1} ||\theta_i||_1. \tag{4.10}$$

The $\ell^1$-regularization opts for fewer parameters, respectively, for driving some weights to zero and thus, reduces the influence of less important weights. At the same time, the most crucial parameters are identified resulting in a more efficient model. The second idea is to use **weight decay** by substituting the regularization term by the squared $\ell^2$-norm, see e.g. [56]. We have

$$R_2(\theta) = ||\theta||_2^2. \tag{4.11}$$

**Remark 4.12.**
*While the $\ell^1$-regularization pushes each weight towards zero by a constant amount, the $\ell^2$-regularization decreases the weights in proportion to their magnitude. The $\ell^1$-penalty is not differentiable at zero and we have to include subgradients in the optimization process. In reality, a thresholding step is required for both, the $\ell^1$- and the $\ell^2$-regularization to obtain sparsity. If a weight is smaller than a threshold parameter in absolute value, it will be forced to zero.*

Another well-known regularization term is the so-called **group lasso** introduced by Yuan and Lin in 2006 [96]. Building upon the notation of reference [81] and using the symbol $\mathcal{G}$ to represent the set of all parameter groups, e.g. given by kernels of convolutional layers, weight matrices between dense layers or bias terms, we can establish a sparsity promoting regularization term acting on groups of parameters

$$R_{1,2}(\theta) = \sum_{g \in \mathcal{G}} \sqrt{n_g} ||g||_2. \tag{4.12}$$

In this context, the variable $n_g$ represents the number of elements in group $g$, such that $\sqrt{n_g}$ ensures that we apply a corresponding weight across all groups.

The three regularization terms introduced here represent just a limited subset of the available options. It is also feasible to combine these terms in various ways to create additional regularization techniques. To generate sparse neural networks, it is also possible to use techniques like **dropout**, also known as **pruning**, or **freezing** [10, 95]. Dropout or pruning involves the random removal of specific network units, as implied by their names. Freezing is the practice of fixing certain weights while performing the optimization process solely with the other weights. For both techniques, fewer derivatives have to be calculated and computing time is saved accordingly.

## 4.2 Learned solution of the eikonal equation

With several thousands of iterations and a reconstruction over days, it is unimaginable to use the given algorithm from Chapter 3 in an economy where an efficient reconstruction process is desirable. It is not a problem that arises specifically from our inverse problem but appears in general whenever multiple partial differential

equations have to be solved for multiple angular positions.

As commonly acknowledged, the Landweber method is quite slow necessitating the implementation of acceleration techniques to enhance its practical applicability. One possibility is the **steepest descent method** where the relaxation parameter $\omega$ is not fixed and depends on the current iteration [48, 76]. As an alternative, one can skip to the **Levenberg–Marquardt method** [30, 44] or the iteratively regularized **Gauss–Newton method** [7, 45] that are second-order iterative methods to solve a non-linear inverse problem. The notable benefit of these approaches lies in their ability to achieve the specified stopping criteria with significantly fewer iterations compared to the Landweber iteration or the steepest descent method. However, each of these iterations consumes more time than a single step in the Landweber iteration, and again, results in a high numerical effort.

Another way to obtain faster convergence, especially for small perturbations $\delta > 0$, is the **Nesterov acceleration** first introduced by Nesterov [70]. The Nesterov acceleration is effective even when applied to non-linear inverse problems. In comparison to the Landweber iteration (3.18), the Nesterov acceleration applies the Landweber step to a linear combination of the last two iterates given by

$$z_k^\delta = x_k^\delta + \frac{k-1}{k+\alpha-1}(x_k^\delta - x_{k-1}^\delta),$$

such that the new iterate is calculated via

$$x_{k+1}^\delta = z_k^\delta - \omega F'(z_k^\delta)^*(F(z_k^\delta) - y^\delta).$$

Here, the parameter $\omega$ is again a relaxation parameter and the scaling parameter $\alpha \geq 3$ is normally chosen as $\alpha = 3$, see [40].

For our interest, we investigate a new approach to speed up the Landweber iteration. Recall, that in our setting the gradient is given as an average over all directions, see equation (3.33). This means we have to evaluate $J$-times the forward operator $F$ to calculate one iteration step. In this section, we substitute this process with a deep neural network given by a CNN. So, given an input, the task of the network is to calculate the travel time for $J = 180$ starting positions of the signal only using one evaluation, respectively, one forward pass. The input in our case is a $57 \times 57$ pixels image of the refractive index $n$. The output is available as the travel time $T$ of the radiation for $57 \times 57 \times 180$ pixels. The total supervised data set can be written as

$$\mathbf{X} \times \mathbf{Y} := \{(X^{(1)}, Y^{(1)}), ..., (X^{(\tilde{N})}, Y^{(\tilde{N})})\} \subseteq \mathcal{X} \times \mathcal{Y} := \mathbb{R}^{57\times57} \times \mathbb{R}^{57\times57\times180}. \quad (4.13)$$

We generate this synthetic training set using simulated FMM data as output data for our network training. The values of the input training matrix $X^{(i)} \in \mathbb{R}^{57\times57}$, $i = 1, ..., \tilde{N}$, resemble the external structure of the objects under investigation. We vary the refractive index of the objects and the inclusions in a range of one to two having $x_{zr}^{(i)} \in [1, 2]$, $z, r = 1, ..., 57$. Further, we shift the position of the defect and modify its size. The second part of the input training data consists of some iterates within some reconstructions, i.e. we do a reconstruction with the normal Landweber algorithm and add some iterates $n_k^\delta$ to our training set. The last part of the input

training data is given by a data augmentation using a relative noise of 2 % and 5 %. Figure 4.6 illustrates the refractive index of some input training data $X^{(i)} \in \mathbb{R}^{57\times 57}$. Part (a) shows a possible object under investigation, part (c) a noisy one and picture (b) and (d) some iterates of a reconstruction.

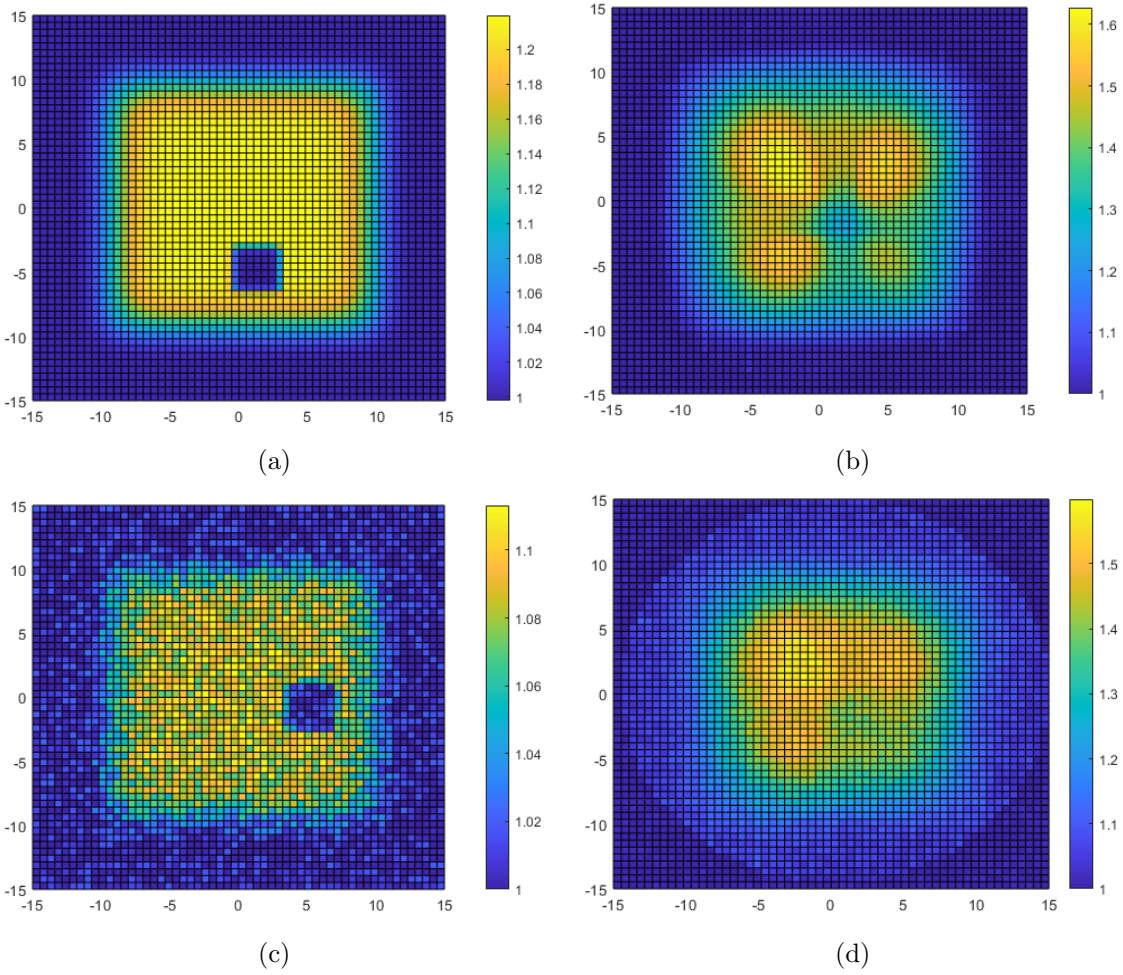



Figure 4.6: Surface plot of the refractive index of some training input data $X^{(i)}$

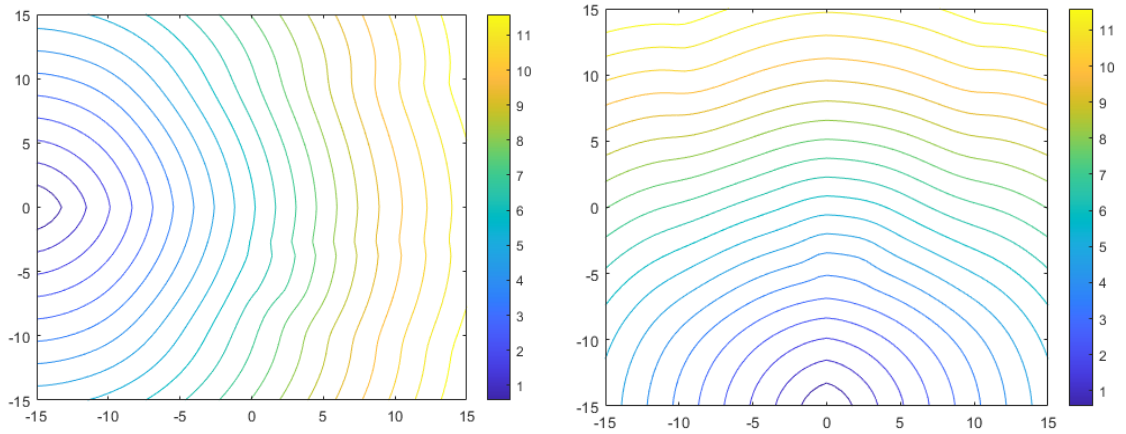
**Remark 4.13.**
*At this point, we would like to mention that there are many ways to compose the training data set. Of course, the items have a direct influence on the training process, and thus, on the performance of the reconstruction in the next chapter. The training set presented here yields the best results for our setting. Also, the amount of training data can be increased arbitrarily. Our goal was to keep the investigations within a manageable framework ending up with $\tilde{N} = 1800$ supervised training data.*

The output training data $Y^{(i)} \in \mathbb{R}^{57\times 57\times 180}$ are received via the FMM. We initialize in Algorithm 3.26, more precisely in Theorem 3.24

$$v_{z,r} = \frac{c_0}{x_{zr}^{(i)}}, \quad z, r = 1, ..., 57,$$

for every input training data $X^{(i)}$ and solve the initial value problem (3.35) for every starting position $j = 1, ..., J$ to obtain the supervised training set. If we take, e.g.

the input training data $X^{(1)}$ given in Figure 4.6 (a), we will obtain 180 matrices of the dimension $57 \times 57$ summarized in the tensor $Y^{(1)}$. The images in Figure 4.7 represent the contour lines of the travel time of the first and the 46th dimension, i.e. angular position, of this output tensor.




(a) (b)

Figure 4.7: Contour plot of two angular positions of the output training data $Y^{(1)}$

Before we start the learning process, we have to think about the structure of the network and the hyperparameters that occur. As mentioned earlier, we prefer a CNN. There are only weak guidelines about the depth and width of the network. The network used was found by trial and error and is presented in the Figure 4.8. The number of layers and units is limited by the computing power of the used hardware. The input of the network passes through three convolutional layers and two fully connected layers. The output is a vector that can be reshaped to the output $\hat{Y} \in \mathbb{R}^{57\times57\times180}$. The ReLU function is used as the activation function throughout the complete network. Each convolutional layer is followed by a max-pooling operator to reduce the amount of parameters.
As one can see, there are many ways to change the hyperparameters of the network: Regarding the convolutional layers the amount of layers, the size of the kernels, the number of the resulting feature maps, the stride and the padding for each convolution can be adjusted. Concerning the fully connected layers again the number of layers can be changed as well as the amount of units within a layer. Further, one can substitute the max-pooling operator, e.g. by an average operator and one can use another activation function.

The network and the optimization process of the weights are implemented via **Python** using the library **torch**. Several optimization algorithms are available for neural networks. The most famous among them are, for example the **Stochastic Gradient Descent (SGD)** and the **Adaptive Moment Estimation (Adam)**. The latter is used in our case. The choice of the appropriate optimization algorithm depends on several factors, including the nature of the problem and the architecture of the neural network. The Adam algorithm uses momentum and an adaptive learning rate. The following algorithm is pre-implemented in torch based on the publication of Kingma and Ba [53]. Note that we want to find the parameters $\theta^*$ that minimize the cost function $J(\theta, \mathbf{X}_T, \mathbf{Y}_T)$. In the following algorithm, we understand $\theta$ as a

vector of all weights of the network.

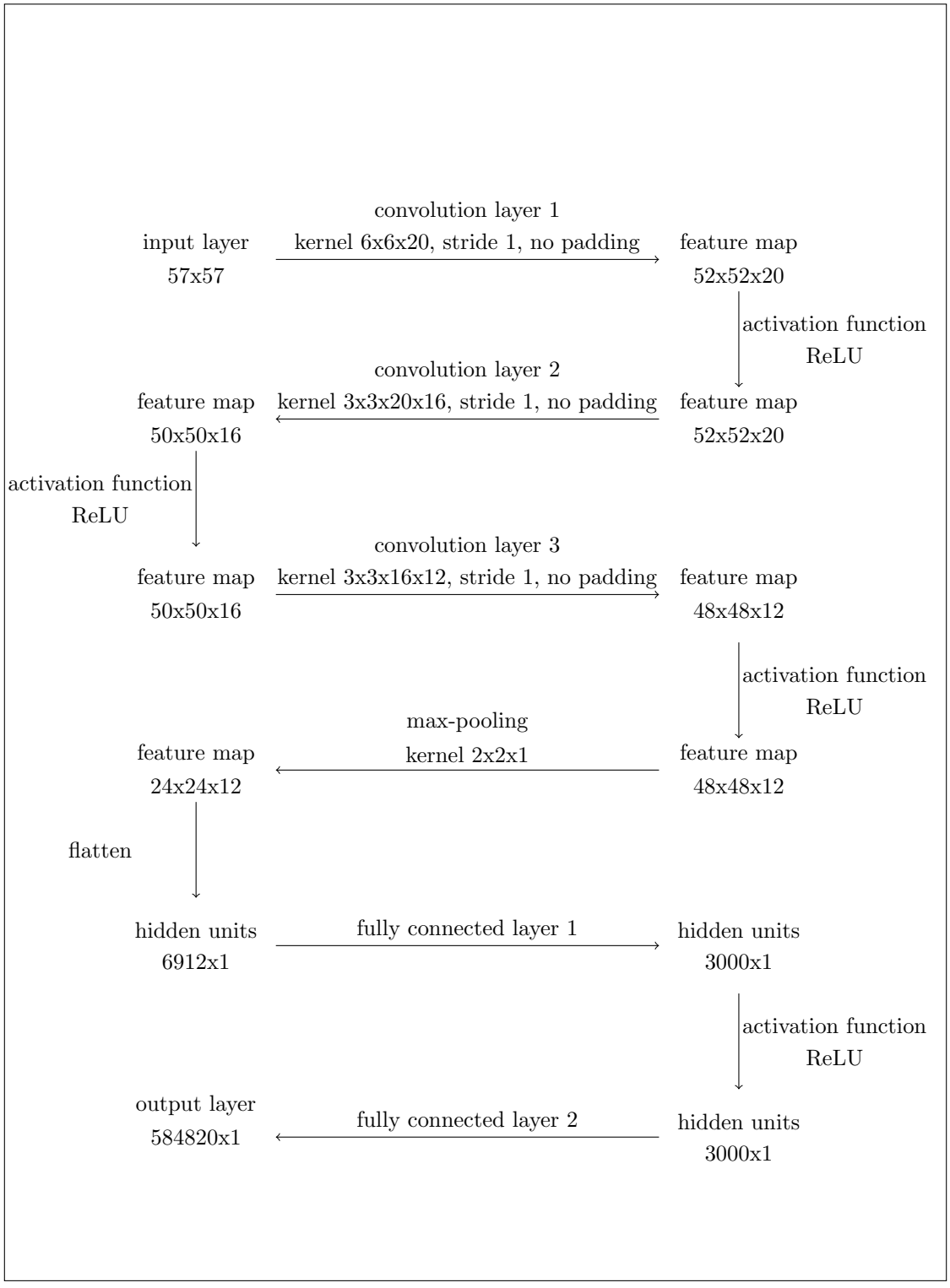



Figure 4.8: Structure of the implemented CNN

**Algorithm 4.14** (**Adam algorithm**).

**INPUT:**

*Initialize the first momentum* $m_0$ *and the the second momentum* $r_0$*:*

$$m_0 = r_0 = 0.$$

*Set* $t = 0$ *and choose* $\beta_1, \beta_2 \in [0, 1)$*,* $\epsilon = 10^{-8}$ *and a small parameter* $a$*.*

**STEP 1:**

*Set* $t = t + 1$ *and calculate*

$$\begin{aligned} g_t &= \nabla_\theta J(\theta, \mathbf{X}_T, \mathbf{Y}_T)|_{\theta=\theta_{t-1}}, \\ m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\ r_t &= \beta_2 r_{t-1} + (1 - \beta_2) g_t^2, \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \quad \hat{r}_t = \frac{r_t}{1 - \beta_2^t}. \end{aligned}$$

**STEP 2:**

*Update* $\theta_t = \theta_{t-1} - a\frac{\hat{m}_t}{\sqrt{\hat{r}_t} + \epsilon}$.

**STEP 3:**

*Repeat Step 1 and Step 2 until convergence.*

To control the learning process, we split our synthetic data set 4.13 randomly into a training data set

$$\mathbf{X}_T \times \mathbf{Y}_T := \{(\tilde{X}^{(1)}, \tilde{Y}^{(1)}), ..., (\tilde{X}^{(N_T)}, \tilde{Y}^{(N_T)})\} \subseteq \mathbf{X} \times \mathbf{Y}$$

consisting of 90 % of the data, i.e. $N_T = 1620$, and a validation set $\mathbf{X}_V \times \mathbf{Y}_V$ including the remaining 10 %, such that

$$\mathbf{X} \times \mathbf{Y} = (\mathbf{X}_T \times \mathbf{Y}_T) \cup (\mathbf{X}_V \times \mathbf{Y}_V).$$

Further, we set $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and choose a small parameter $a = 10^{-5}$. We go through 4000 **epochs** with a **batch size** of all training data.

**Remark 4.15.**

*a) Normally, the cost functional* $J(\theta, \mathbf{X}_T, \mathbf{Y}_T)$ *is not differentiable because of the activation functions used in the network. In our case, we use the ReLU function that is not differentiable at zero. The pre-implemented optimization toolbox in torch handles this problem by including subgradients. For more information about subgradient methods, we refer to [42].*

*b) To accelerate the training process of the network, it is possible to subdivide the training data into batches. For the first optimization step the first batch is used, then the second and so on. An epoch is completed when all batches have been run through once. In our case, we can use a batch size of* 1620 *because of having a small training set.*

The cost function is given by using the mean squared error as the loss function. We obtain for the optimization process

$$\begin{aligned} J(\theta, \mathbf{X}_T, \mathbf{Y}_T) &= \frac{1}{N_T} \sum_{i=1}^{N_T} l\left(\Phi_\theta(\tilde{X}^{(i)}), \tilde{Y}^{(i)}\right) \\ &= \frac{1}{N_T} \sum_{i=1}^{N_T} \frac{1}{57 \cdot 57 \cdot 180} ||\Phi_\theta(\tilde{X}^{(i)}) - \tilde{Y}^{(i)}||_2^2 \\ &= \frac{1}{1620 \cdot 584820} \sum_{i=1}^{1620} \sum_{j=1}^{584820} \left(\Phi_\theta(\tilde{X}^{(i)})_j - \tilde{Y}_j^{(i)}\right)^2. \end{aligned} \tag{4.14}$$

The values of the cost function after some epochs for the training and the validation set are given in Table 4.1. Note, that we use

$$J(\theta, \mathbf{X}_V, \mathbf{Y}_V) = \frac{1}{\tilde{N} - N_T} \sum_{i=N_T+1}^{\tilde{N}} l\left(\Phi_\theta(\tilde{X}^{(i)}), \tilde{Y}^{(i)}\right) \tag{4.15}$$

for the evaluation of the validation set. We notice a small deviation between the values of the two cost functions which can be reduced by further increasing the amount and the diversity of the training data.

Table 4.1: Value of the cost function for the training and validation set

| Amount of epochs | 3000 | 4000 | 6000 |
|---|---|---|---|
| $J(\theta, \mathbf{X}_T, \mathbf{Y}_T)$ | 0.0054 | 0.0045 | 0.0035 |
| $J(\theta, \mathbf{X}_V, \mathbf{Y}_V)$ | 0.0064 | 0.0055 | 0.0045 |

Up to this point, we have generated the training and validation data set, defined the structure of the network and determined its parameters via an optimization process of the cost function. In the following, we investigate how well the network maps unknown inputs and how large the time saving is. We start with a randomly chosen input $X_{Test_1}$. Its external structure resembles the external structure of the training data, see Figure 4.9 (a). Note that here the material-dependent velocity is depicted and not the refractive index. We compute the forward operator once via the FMM and once via the deep neural network $\Phi_{\theta^*}$. The overlapping contour lines for one angular position are plotted in Figure 4.9 (b). For this example, we can calculate the relativ error $e_r$ as

$$e_r = \frac{||\Phi_{\theta^*}(X_{Test_1}) - FMM(X_{Test_1})||}{||FMM(X_{Test_1})||} = 0.0124.$$

The results for unchanging external structures are very promising. The error increases by translating to new structures, see Figure 4.9 (c) and (d). However, with a relative error of 3.65 %, we are still within a low error range considering that no round external structures were integrated into the learning process.

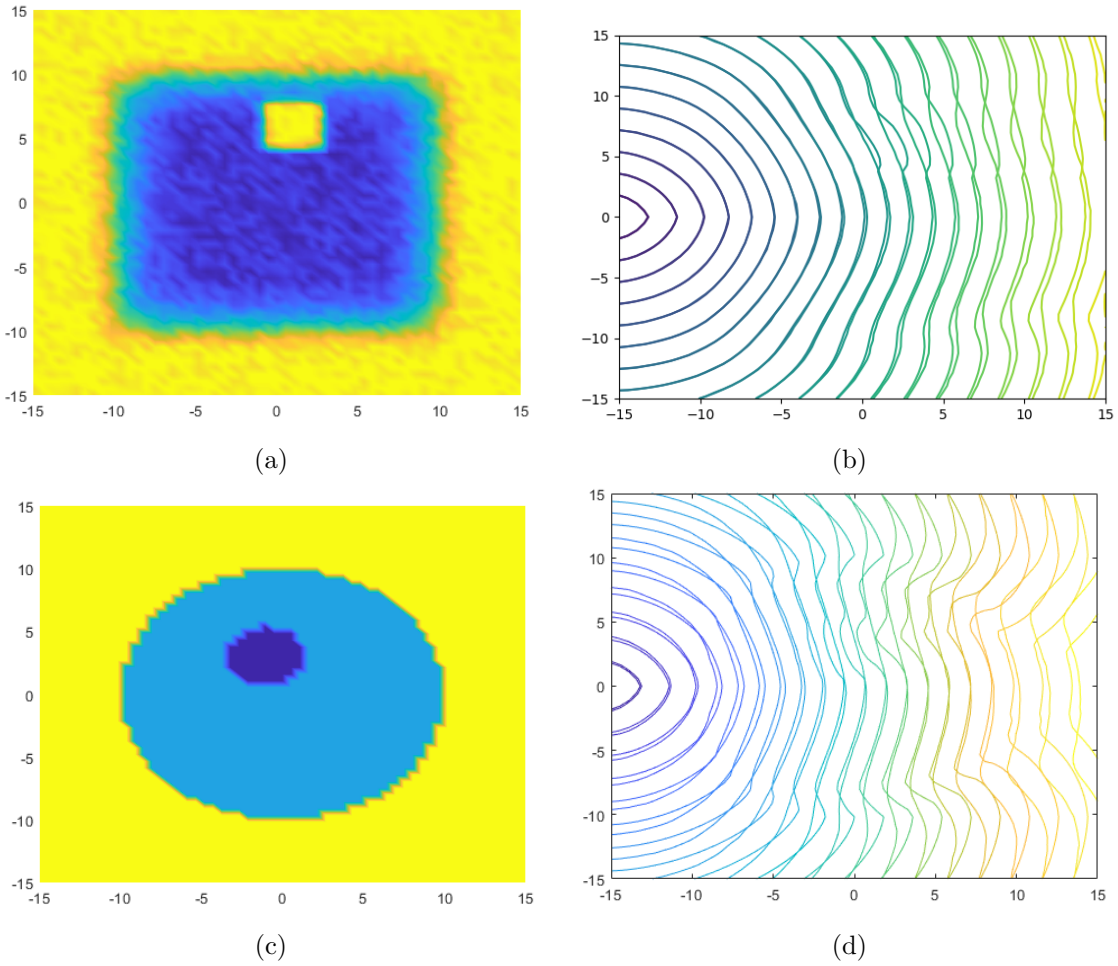



Figure 4.9: Comparison of the performance of the FMM and the CNN for two test data $X_{Test_1}$ (a) and $X_{Test_2}$ (c) displaying the overlapping contour lines of the solution of the forward operator given by the FMM and the network for one angular position (b), (d)

While the influence of the relative error on the reconstruction will be investigated in the next subsection, we would like to conclude this subsection by comparing the computation time between the two forward operators. The aim was to speed up the Landweber iteration. So, if we accelerate the evaluation of the forward operator, we can automatically achieve our goal. The implementation was done on an Intel Core i9-11900K processor containing eight kernels and a RAM of 64 GB. For the evaluation of the FMM, we use all eight kernels, such that the differential equation is solved parallel for some initial conditions. For the forward pass through the CNN, just one kernel is necessary. If we look at our test file $X_{Test_1}$ again, we can see an average speedup by a factor of 8.8. Nearly the same holds for the second test data $X_{Test2}$ with a speedup by the factor 9.3 because the evaluation time is independent of the value of the refractive index. We can thus record the first result for Chapter 4:

The substitution of the forward operator solved by the FMM with a CNN including an a prior learning process results in a significant acceleration of one iteration step in the Landweber iteration.

## 4.3 Numerical evaluation of the learned Landweber iteration

In the last subsection, we developed and tested a way to replace the forward operator with a CNN. The learning process resulted in a significant improvement of the computation time concerning the evaluation of the forward operator together with a low relative error in the solution. Now, we investigate how the new forward operator affects the reconstruction process using the Landweber method. From a mathematical point of view, we aim to reconstruct with the learned Landweber iteration

$$v_{i+1}^{\delta} = v_i^{\delta} - \omega F'(v_i^{\delta})^*(\Phi_{\theta^*}(v_i^{\delta}) - y^{\delta}), \quad i = 0, ..., k^* - 1, \tag{4.16}$$

where $v_{k^*}^{\delta}$ denotes the approximated solution after $k^* - 1$ iteration steps. We test different phantoms that are unknown to our learned network and calculate the total reconstruction error. By comparing the normal Landweber iteration with the learned one, we observe that we include an error $\epsilon(v_i^{\delta})$ in every iteration step having

$$F(v_i^{\delta}) = \Phi_{\theta^*}(v_i^{\delta}) + \epsilon(v_i^{\delta}), \tag{4.17}$$

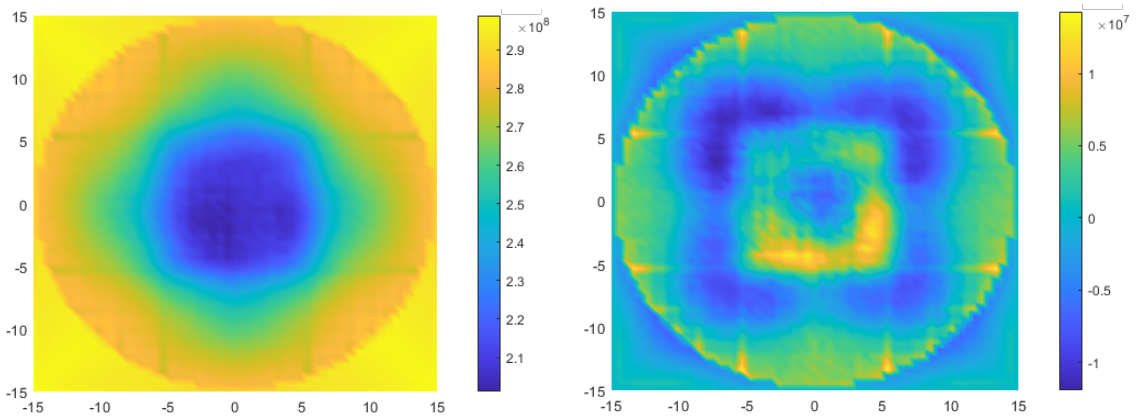
such that the normal iteration step is corrupted by the term $\omega F'(v_i^{\delta})^*\epsilon(v_i^{\delta})$. We obtain

$$\begin{aligned} v_{i+1}^{\delta} &= v_i^{\delta} - \omega F'(v_i^{\delta})^*(F(v_i^{\delta}) - \epsilon(v_i^{\delta}) - y^{\delta}) \\ &= v_i^{\delta} - \omega F'(v_i^{\delta})^*(F(v_i^{\delta}) - y^{\delta}) + \omega F'(v_i^{\delta})^*\epsilon(v_i^{\delta}). \end{aligned}$$

We start with example $v_1$, the Gaussian function of equation (3.36). The stopping index $k^*$ is chosen as the first iterate where the norm of the residual stops decreasing. It holds

$$\begin{aligned} &||y^{\delta} - \Phi_{\theta}^*(v_i^{\delta})|| - ||y^{\delta} - \Phi_{\theta}^*(v_{i+1}^{\delta})|| \geq 0 \text{ for } i = 0, ..., k^* - 1, \\ &||y^{\delta} - \Phi_{\theta}^*(v_i^{\delta})|| - ||y^{\delta} - \Phi_{\theta}^*(v_{i+1}^{\delta})|| < 0 \text{ for } i = k^*. \end{aligned}$$

We compare the relative error of the reconstruction via the learned Landweber method using Network 4.8 of the previous subsection with the results of Chapter 3. First of all, we notice that the reconstruction with the relaxation parameter $\omega = 0.1$ already stops after 152 steps. It is obvious, that the reason could be the error term in the learned Landweber iteration. Nevertheless, we result in a low relative error of 1.69 % that, however, is about three times higher than the relative error of the normal Landweber iteration. So, the quality of the reconstruction becomes worse, but the reconstruction time is significantly reduced by accelerating the evaluation of the forward operator. The latter is an enormous advantage concerning inline monitoring. Figure 4.10 displays the reconstruction of $v_1$ and the absolute error $v_1 - v_{1,k^*}^{\delta}$ using a noise of 2 % for the travel time data. In Figure 4.11, we compare the development of the relative error of the normal and the learned Landweber iteration. We see that we obtain similar results in the first iterations, but the learned Landweber method reaches its lower bound faster. This is due to the error term in the forward operator, leading to modelling inaccuracy.




(a) (b)

Figure 4.10: Reconstruction $v^{\delta}_{1,k^*}$ (a) of the material-dependent velocity $v_1$ with the learned Landweber iteration showing the error plot $v_1 - v^{\delta}_{1,k^*}$ (b)

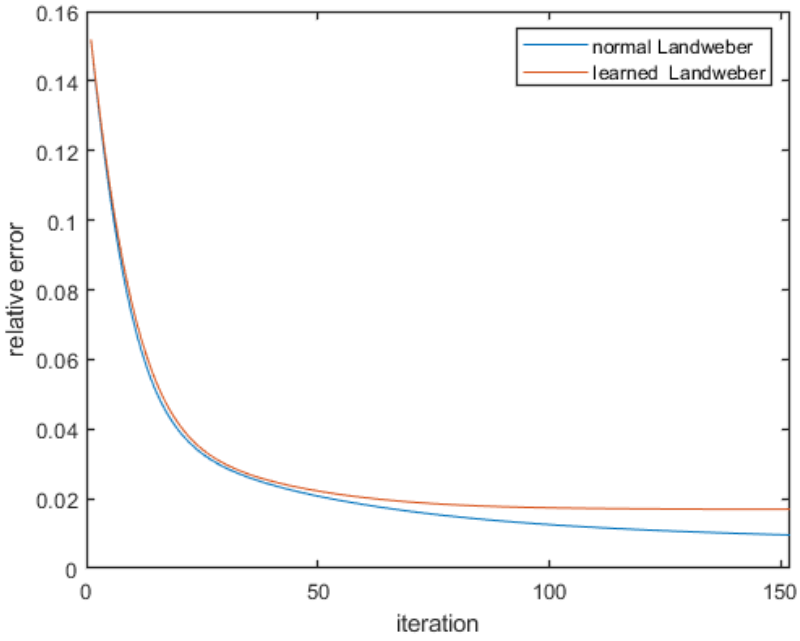



Figure 4.11: Development of the relative error for the normal and the learned Landweber iteration for the reconstruction of phantom $v_1$

In the second step, we investigate more complex structures. Again, we include a defect in the plastic block. This time, we observe a central air inclusion by using the function

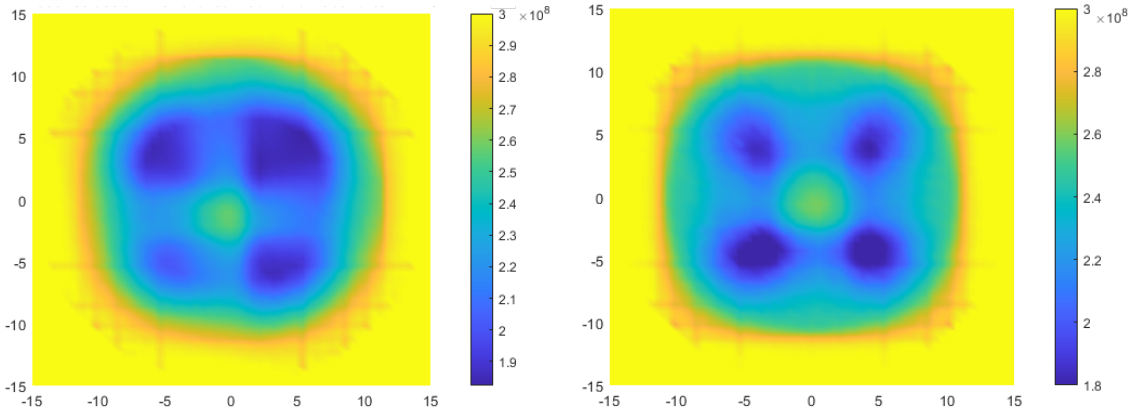
$$v_5 : \Omega \to \mathbb{R}^+, \; v_5(x,y) = (3 - \mathrm{e}^{\frac{-x^8-y^8}{10^8}} + \mathrm{e}^{-\frac{x^8+y^8}{300}}) \cdot 10^8.$$

Further, we use function $v_3$, see equation (3.37), to compare again our results with the normal Landweber iteration. For both reconstructions, we obtain qualitative information about the position of the unknown defect and a relative error between four and five percent. While we obtain quantitative inferences applying the normal Landweber procedure, we obtain only the external structure with the learned Landweber procedure. We can only recognize the approximate structure and position,

see Figure 4.12. At this point, it is important to mention that the blue artifacts of the reconstructions are normal in a small iteration range and are normally smoothed by more iterations. Here, the stopping index is significantly lower and the artifacts are present. All in all, we use a relatively small training set. It is left to amplify the training data to obtain more detailed results. For this, better hardware is necessary or we have to transfer the network to a server. In Table 4.2, we summarize the reconstruction parameters and results for the three investigated functions:

Table 4.2: Reconstruction parameters and results for the functions $v_1$, $v_3$ and $v_5$ with the learned Landweber iteration

| | $v_1$ | $v_3$ | $v_5$ |
|---|---|---|---|
| Amount of iterations $k^*$ | 152 | 11 | 29 |
| Relative error $e_r$ | 1.69 % | 4.93 % | 4.36 % |
| Residual $\|\|R^\delta_{k^*}\|\|$ | 0.5703 | 0.8856 | 1.1644 |
| Relaxation parameter $\omega$ | 0.1 | 2.5 | 1.5 |



(a) (b)

Figure 4.12: Reconstruction $v^\delta_{3,k^*}$ (a) and $v^\delta_{5,k^*}$ of the material-dependent velocity $v_3$ and $v_5$ with the learned Landweber iteration

Note that we are mainly interested in the execution time of the reconstruction process and the quantitative structure of the object under investigation. For further qualitative information one can use, e.g. the modified ART developed by Jens Tepe et al. [88]. We see that both goals have been achieved. On the one hand, we were able to accelerate the reconstruction process and, on the other hand, we were able to localize possible defects.

Finally, it is important to mention that we can improve the reconstruction process using some preliminary information. Suppose we include the information that we search for a defect in the middle. In that case, we can improve the reconstruction of the learned Landweber iteration with respect to phantom $v_5$, see Figure 4.13. Here, we use the same structure for the network but an adapted training data set including more centered defects. However, if we reconstruct phantom $v_1$ with this

training data, we will fail due to an overfitting process. In Figure 4.14, we present the reconstruction of phantom $v_5$ after the same number of steps using the normal Landweber iteration and we obtain a worse reconstruction which also contains the blue artifacts. The reconstruction with the normal Landweber method is of course significantly better if the number of iterations is increased, see Figure 4.15.

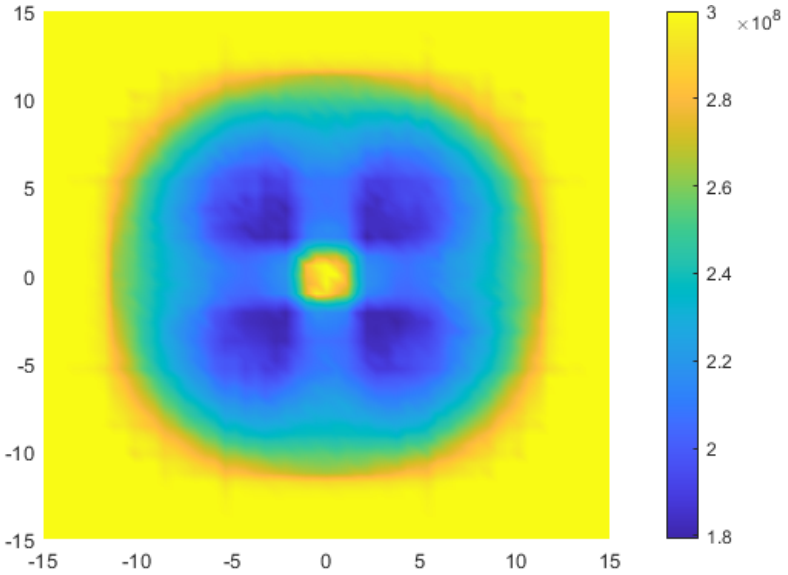



Figure 4.13: Reconstruction $v^{\delta}_{5,k^*}$ of the velocity $v_5$ after $k^* = 25$ iterations of the learned Landweber method using an adapted training data set

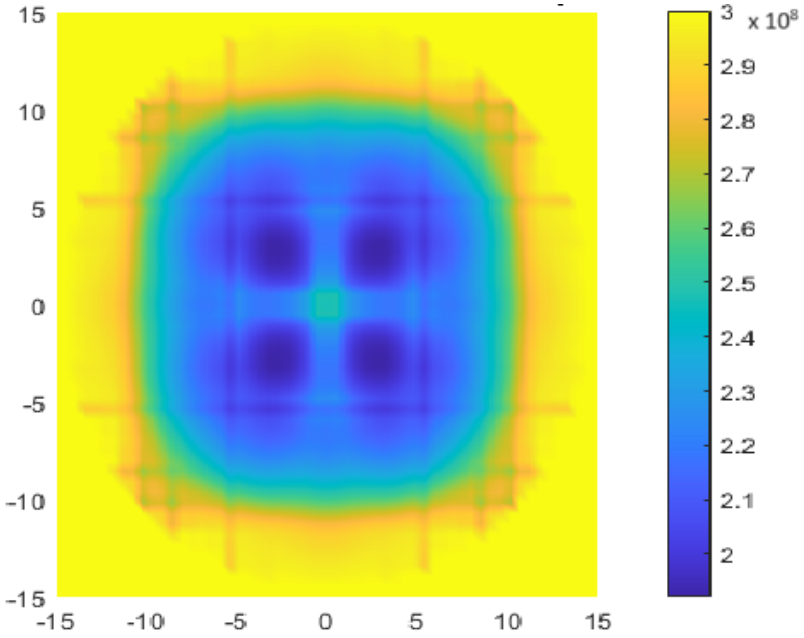



Figure 4.14: Reconstruction $v^{\delta}_{5,k^*}$ of the velocity $v_5$ after $k^* = 25$ iterations of the normal Landweber method

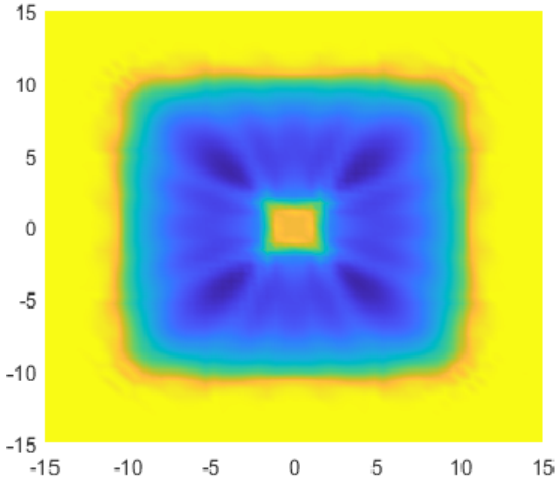



Figure 4.15: Reconstruction $v^{\delta}_{5,k^*}$ of the velocity $v_5$ after $k^* = 13000$ iterations of the normal Landweber method

## 4.4 Sparsity-based learning of the eikonal equation

The presented learned Landweber method is the result of an excessive energy consumption. To obtain the findings of the last subsection several cycles consisting of the generation and selection of the training data, the determination of the network structure and an evaluation process with the final reconstruction via the learned Landweber method were performed. Especially, the optimization process of the network parameters and the evaluation process is costly and leads together with the high amount of free parameters to enormous computational costs. We end up optimizing large models with billions of parameters. The enhancement in the performance of deep learning models is largely driven by the expansion of network weights and the subsequent increase in computational requirements for training the network [95]. Consequently, the computational costs are exponentially increasing and the carbon footprint associated with training expansive neural networks has become important for many companies [16]. One way out is the inclusion of mathematical concepts that result in sparse weights of the network with which we deal in the following subchapter. We generate sparse parameters $\tilde{\theta}$ meaning that some weights are set to zero. In doing this, we neglect small parameters or those with minimal impact on the network's output. In our research, we mainly focus on using a regularization term within the optimization process, see equation (4.9), and transfer the theory to the given example of learning the solution of the eikonal equation. We investigate whether the SNN performs as well as the dense one. The investigations done in this subsection are inspired by the project **DELETO - Deep Learning in Tomography** founded by the **Bundesministerium für Bildung und Forschung (BMBF)** where sparse neural networks represent a central content.

We start the investigations of a simpler problem, a kind of toy problem, to gain a first impression about the application of sparsity-generating processes. For this purpose, we use the so-called **MNIST database**, a large data set of handwritten digits that includes 60000 training and 10000 validation items [58]. We model a CNN consisting of two convolutional layers and two fully connected layers with a total of

21330 weights. The network aims to solve the classification task of recognizing the given number. The complete structure of the network is provided in Appendix C.1. For this toy problem, we compare the performance of the dense network with the sparse ones where the sparsity is generated once without regularization just using a threshold parameter (**pruned network**) and once by installing an $\ell^1$-regularization in the optimization process together with a threshold parameter (**regularized network**). Furthermore, we introduce freezing in the learning process of the parameters (**frozen network**) where some weights are randomly skipped in the optimization process. In Table 4.3, we show the results of these four optimization processes: We see that the dense network performs best, closely followed by the regularized network. The pruned network is better than the randomly freezing process but worse than the $\ell^1$-regularization. All in all, we generate a sparsity of 45.74 % with the $\ell^1$-regularization term which is remarkable seeing that the accuracy only is 0.007 % worse. By the term **sparsity** in this context we mean the percentage of weights equal to zero among all weights of the network.

Table 4.3: Performance of a dense network, a regularized network, a pruned network and a frozen network applied to the MNIST database

| network | dense | regularized | frozen | pruned |
|---|---|---|---|---|
| $J(\theta, \mathbf{X}_T, \mathbf{Y}_T)$ | 0.1481 | 0.1725 | 0.3215 | 0.2092 |
| $J(\theta, \mathbf{X}_V, \mathbf{Y}_V)$ | 0.1333 | 0.1615 | 0.2994 | 0.191 |
| accuracy | 0.96 | 0.953 | 0.915 | 0.946 |

After this initial experiment, we want to transfer the idea of sparsity to a more complex task: the learning process of the solution of the eikonal equation. Due to the results of the toy problem, we only compare the dense network with the regularized one. In a first setting, we use a small data set consisting of 340 data that are subdivided into 306 training and 34 validation data. We compare two dense networks, one with 18 million parameters (**dense network 1**) and another with 1370 million parameters (**dense network 2**). For the larger network, we also generate a SNN using an $\ell^1$-regularization combined with a threshold parameter. We apply these networks to the regression problem of solving the eikonal equation. The results are very promising, as shown in Table 4.4. The larger network achieves a sparsity of 50.8 % resulting in minimal performance reduction. Additionally, we observe that the network with fewer parameters performs significantly worse than the network with more weights which is consistent with existing literature. The findings are visualized for an input from the validation set in Figure 4.16. We see the overlapping contour lines of the travel time $T(\mathbf{x})$ calculated once using the FMM and once using the neural network, for all three settings.

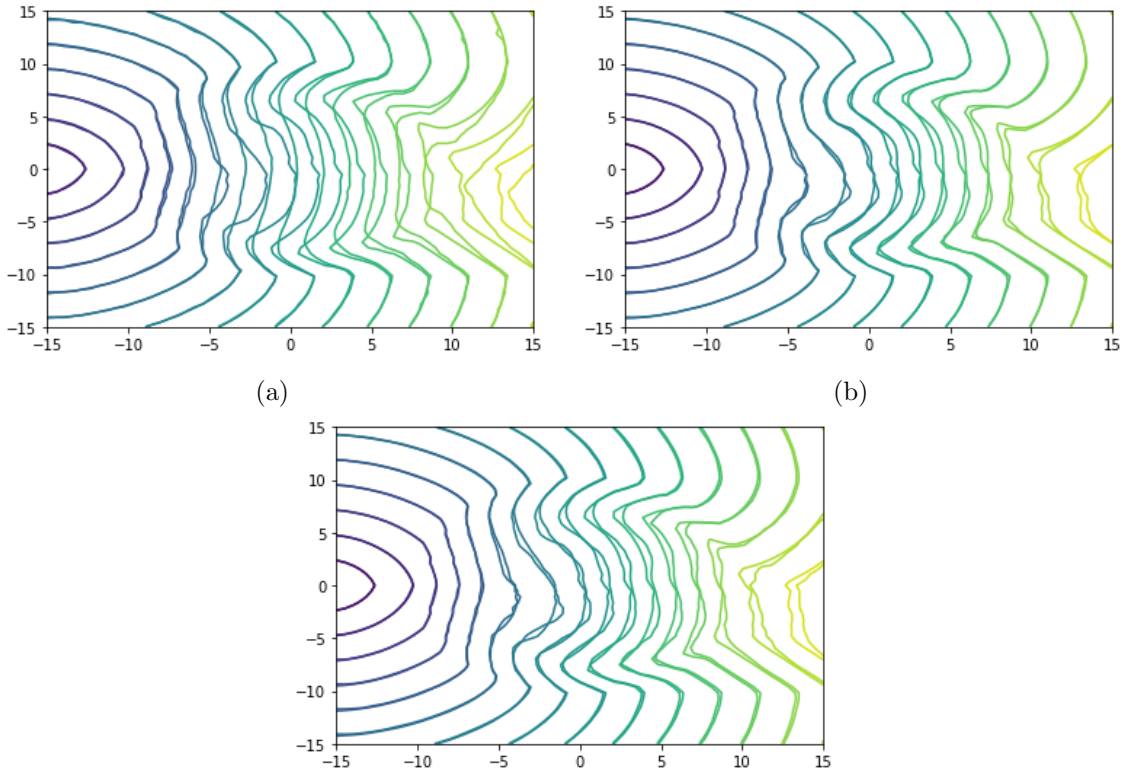



Figure 4.16: Visualization of the network performance for the dense network 1 (a), dense network 2 (b) and the SNN (c) in comparison to the ground truth contour lines of the travel time $T(\mathbf{x})$

Table 4.4: Performance of two dense networks and a sparse one applied to the eikonal equation using a small data set

| network | dense 1 | dense 2 | sparse |
|---|---|---|---|
| $J(\theta, \mathbf{X}_T, \mathbf{Y}_T)$ | 0.0315 | 0.0015 | 0.0017 |
| $J(\theta, \mathbf{X}_V, \mathbf{Y}_V)$ | 0.0433 | 0.0032 | 0.0035 |

Based on the good results of the preliminary examinations, we now deal with the actual problem: We use the network of Figure 4.8 and add some sparsity promoting regularization term to the cost functional. The generated sparsity can be controlled by two parameters, the regularization parameter $\alpha$ as well as the threshold parameter $\zeta$. We use an $\ell^1$-regularization, i.e. we solve

$$\tilde{\theta}^* = \arg\min_{\theta \in \Theta} \Big( J(\theta, X, Y) + \alpha R_1(\theta) \Big). \tag{4.18}$$

We take again the Adam algortihm 4.14 where $g_t$ is given corresponding to equation (4.18) as

$$g_t = \nabla_\theta \Big( J(\theta, X, Y) + \alpha R_1(\theta) \Big) \Big|_{\theta = \theta_{t-1}}$$

We perform a thresholding after fifty iterations and set all weights less than $\zeta$ to zero. We only update the moments and parameters according to the non-zero values. The Adam algorithm 4.14 is transformed to the sparsity-generating Adam algorithm as follows:

**Algorithm 4.16.** *(**Sparsity generating Adam algorithm**)*

**INPUT:**

*Initialize the first momentum $m_0$ and the the second momentum $r_0$:*

$$m_0 = r_0 = 0.$$

*Set $t = 0$ and choose $\beta_1, \beta_2 \in [0, 1)$, $\epsilon = 10^{-8}$ and a small parameter $a$. Choose a threshold parameter $0 < \zeta << 1$ and a regularization parameter $\alpha$.*

**STEP 1:**

*Set $t = t + 1$ and calculate*

$$g_t = \nabla_\theta \Big( J(\theta, X, Y) + \alpha R_1(\theta) \Big) \Big|_{\theta=\theta_{t-1}},$$
$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \;\; r_t = \beta_2 r_{t-1} + (1 - \beta_2) g_t^2,$$
$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \;\; \hat{r}_t = \frac{r_t}{1 - \beta_2^t}.$$

**STEP 2:**

*Update $\theta_t = \theta_{t-1} - a \frac{\hat{m}_t}{\sqrt{\hat{r}_t} + \epsilon}$.*

**STEP 3:**

*For $t$ mod 50 = 0 set all weights less than $\zeta$ equal to zero.*

**STEP 4:**

*Repeat Step 1, 2 and 3 until convergence. Skip all weights equal to zero.*

**Remark 4.17.**
*The fourth step of Algorithm 4.16 can be modified. Instead of skipping the weights equal to zero, we can use subgradients for these weights. In the implementation, we use for every layer a mask with entries equal to zero to guarantee the sparsity and to skip the parameters equal to zero.*

We choose different values for the threshold parameter $\zeta$ and for the regularization parameter $\alpha$ and measure the generated sparsity, the value of the cost function for the training data and the value of the cost function for the validation set. Note that we use the cost functions (4.14) and (4.15) to compare the values with the dense network of Subsection 4.2. If we select the parameters $\alpha$ and $\zeta$ to large, the promoted sparsity will tend to be 100 % which results in unusable networks for the reconstruction. For the whole investigation, we fall back on the same training and validation data as in Section 4.2. The results are illustrated in Table 4.5. We depict

the values of the cost functions and the generated sparsity for tree combinations of threshold parameter and regularization parameter. We see that the choice of smaller parameters leads to a lower sparsity of the weight $\tilde{\theta}_i^*$, $i = 1, 2, 3$, but to a lower value of the cost functions. In total, we carry out 4000 epochs, again, with a batch size corresponding to the amount of training data. We notice that the sparse networks perform slightly worse than the dense network where the values of the cost functions are given in Table 4.1. It is noteworthy that with the choice of $\alpha = \zeta = 10^{-8}$ we obtain nearly the same result though we save about 12 % of the parameters.

Table 4.5: Performance of the sparse network applied to the eikonal equation using the data set of Section 4.2 for different threshold parameters $\zeta$ and regularization parameters $\alpha$

| Threshold parameter $\zeta$ | $1 \cdot 10^{-7}$ | $5 \cdot 10^{-8}$ | $1 \cdot 10^{-8}$ |
|---|---|---|---|
| regularization parameter $\alpha$ | $1 \cdot 10^{-7}$ | $5 \cdot 10^{-8}$ | $1 \cdot 10^{-8}$ |
| $J(\theta, \mathbf{X}_T, \mathbf{Y}_T)$ | 0.007 | 0.0066 | 0.0056 |
| $J(\theta, \mathbf{X}_V, \mathbf{Y}_V)$ | 0.0079 | 0.0056 | 0.0046 |
| Sparsity of $\tilde{\theta}_i^*$ | 59.11% | 48.31% | 11.9% |

To conclude this subsection, we illustrate the distribution of the sparsity in Figure 4.17 where we see the percentage of weights equal to zero per layer for the three settings of $\alpha$ and $\zeta$. We notice that the highest percentage of zero elements occur in the fully connected layers, therefore, in the last two layers of the network. At this point, it would be interesting to investigate whether the percentages will change significantly if the layers are arranged differently. Does the sparsity depend on the type of layer or the position of the layer in the network? A suitable explanation is provided by vanishing gradients: The weight updates are very small in the front layers near the input, and so, these weights tend slower to zero [34].

## 4.5 Numerical evaluation of the sparsity-based learned Landweber iteration

Analogous to Subsection 4.3, we investigate how the SNN affects the reconstruction process and compare our results, especially with the learned Landweber iteration. We transform equation (4.16) into a sparsity-based Landweber iteration by substituting the optimal dense weights $\theta^*$ by the optimal sparse weights $\tilde{\theta}^*$. We obtain a third version of Landweber's method

$$v_{i+1}^\delta = v_i^\delta - \omega F'(v_i^\delta)^*(\Phi_{\tilde{\theta}^*}(v_i^\delta) - y^\delta), \quad i = 0, ..., k^* - 1, \tag{4.19}$$

where again $v_{k^*}^\delta$ denotes the approximated solution after $k^* - 1$ iterations steps. We focus on phantom $v_1$ and $v_5$ and use the optimal weights $\tilde{\theta}_1^*$, $\tilde{\theta}_2^*$, $\tilde{\theta}_3^*$ within the neural network to reconstruct. The stopping index $k^*$ is chosen as the first index for which the norm of the residual no longer decreases. It holds

$$||y^\delta - \Phi_{\tilde{\theta}^*}(v_i^\delta)|| - ||y^\delta - \Phi_{\tilde{\theta}^*}(v_{i+1}^\delta)|| \geq 0 \text{ for } i = 0, ..., k^* - 1,$$
$$||y^\delta - \Phi_{\tilde{\theta}^*}(v_i^\delta)|| - ||y^\delta - \Phi_{\tilde{\theta}^*}(v_{i+1}^\delta)|| < 0 \text{ for } i = k^*.$$

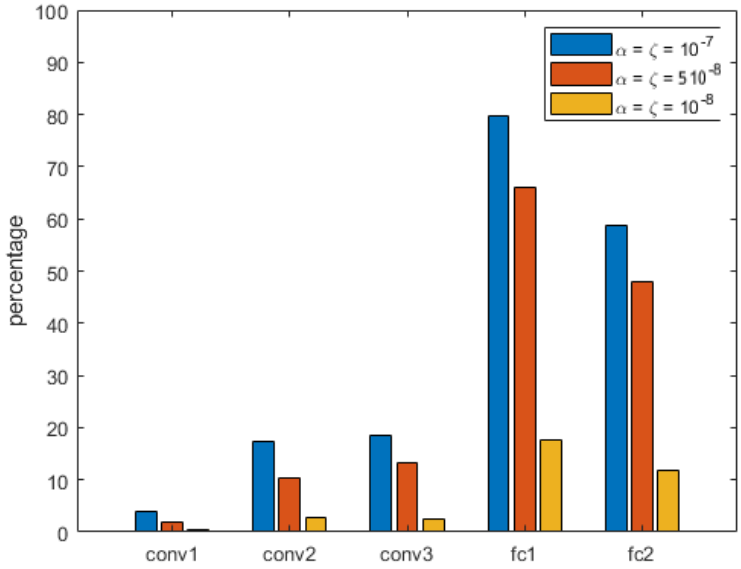



Figure 4.17: Percentage of the weights equal to zero per layer (convolutional layer (conv), fully connected layer (fc))

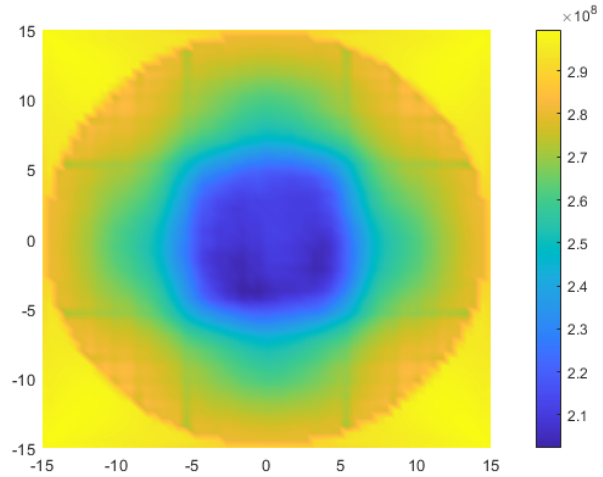
We note that the reconstruction results for the two phantoms under consideration are only minimally worse than the reconstruction results of Subchapter 4.3, see Table 4.6, comparing the relative error and using the same relaxation parameter $\omega$ and noise level $\delta$. Applying the sparse weight $\tilde{\theta}_3^*$, the relative error increases by 0.22 % for the reconstruction of $v_1$ and just by 0.01 % for the reconstruction of $v_5$. This is remarkable as approximately 12 % of the weights were set to zero. As the values of the cost function of Table 4.5 suggest the reconstruction is worse for $\tilde{\theta}_1^*$ and $\tilde{\theta}_2^*$ but still remarkably close to the results of the learned Landweber method, considering that approximately 50 % of the weights were eliminated.

Table 4.6: Reconstruction results for the functions $v_1$ and $v_5$ with the sparsity-based learned Landweber iteration using $\tilde{\theta}_1^*$, $\tilde{\theta}_2^*$, $\tilde{\theta}_3^*$

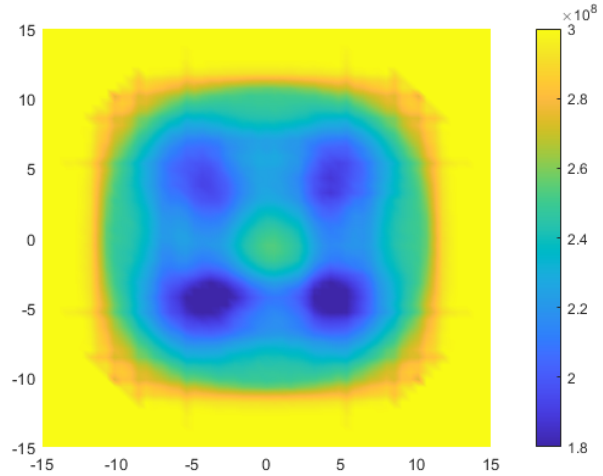
| | $v_1$ | | | $v_5$ | | |
|---|---|---|---|---|---|---|
| Network weights | $\tilde{\theta}_1^*$ | $\tilde{\theta}_2^*$ | $\tilde{\theta}_3^*$ | $\tilde{\theta}_1{}^*$ | $\tilde{\theta}_2{}^*$ | $\tilde{\theta}_3{}^*$ |
| Amount of iterations $n^*$ | 71 | 50 | 130 | 26 | 28 | 29 |
| Relative error $e_r$ | 2.08 % | 2.78 % | 1.91 % | 4.58 % | 4.76 % | 4.37 % |
| Residual $\|\|R_{k^*}^{\delta}\|\|$ | 1.992 | 0.9447 | 0.6095 | 1.4604 | 1.64 | 1.1511 |
| Parameter $\omega$ | 0.1 | 0.1 | 0.1 | 1.5 | 1.5 | 1.5 |

If we look at the reconstruction images, see Figure 4.18, we can also make quantitative statements as in Subsection 4.3. We recognize the defect in the center of the object, see Figure 4.18 (b). Of course, just like the learned Landweber method, the sparsity-based one does not provide exact reconstructions. The iteration of the two

methods stops earlier than the normal Landweber method due to the inaccuracy in the forward operator. By expanding the training data and increasing the size of the network, the learning process associated with the reconstruction should be improved.




(a)




(b)

Figure 4.18: Reconstruction $v^{\delta}_{1,k^*}$ (a) and $v^{\delta}_{5,k^*}$ (b) of the material-dependent velocity $v_1$ and $v_5$ with the sparsity-based learned Landweber method using $\tilde{\theta_3}^*$

According to our research, a SNN performs almost as well as a dense network. To generate good reconstruction results, the modeling of the network and the training data play the most important roles. Once these have been found, the sparsity approach used here can achieve significant energy savings without a deterioration of the reconstruction. Regarding the $\ell^1$-regularization term, it is important to fine-tune the parameters $\alpha$ and $\zeta$ to control the sparsity.

# 5 Learned anomaly detection with terahertz radiation in inline monitoring

The following chapter is based on the AiF project **Terahertz-Prozesstomografie bei der Extrusion von Kunststoffhalbzeugen** in collaboration with the SKZ in Würzburg. The content is published in [64, 66, 67]. We deal with process monitoring in the extrusion of plastics using THz radiation. Detecting internal defects in a material is one of the main challenges in inline monitoring of plastics. We aim to reduce product discard by an efficient inline control that intervenes directly during the manufacturing process. In the project, we concentrate on whether a defect is present and not on the exact reconstruction of the object. The latter is investigated in the previous chapters. The content of the project marks the starting point of the whole investigations presented in this thesis. In addition to the detection of defects, the exploration aims to assess the appropriateness of the eikonal equation as a suitable physical model for simulating the high-frequency propagation of electromagnetic waves.

In the first study, we use supervised training data from a measuring system that approximates an inline process. The measuring system and the real-time measurements were generated by the SKZ in Würzburg. We use THz radiation from 120 GHz to 170 GHz. The system consists of one emitter and three receivers that rotate around the object of interest. We measure data encoding intensity, refraction, reflection and temporal information. With these data, we develop a technique based on a learned probability density function to automatically detect outliers in an inline process. A threshold parameter is determined. If a measurement falls below this threshold parameter, there is a high probability that a defect is present. In a further study, we restrict ourselves to a single feature, the temporal information, and include simulated data in the learning process. For this purpose, we model the propagation of the THz radiation by the eikonal equation to enlarge the set of data for this feature. We simulate a diversity of defects without the need for further time-consuming measurements. We test our investigations on an unknown object and compare the anomaly detection trained on the hybrid data with the anomaly detection based on non-simulated ones.

In the following, we present the mathematical basis of anomaly detection and the resulting algorithm. Then, we describe the measurement system and the data set before illustrating and discussing the numerical results.

## 5.1 Classification algorithm

We evaluate measurements of THz radiation from an inline process with a machine learning technique called **anomaly detection** to detect defects in products. We define an **anomaly** as a significant variation from typical values [65]. The idea of anomaly detection to alarm if a measurement is inconsistent with the expected behavior is a typical machine learning task and is applied in many different fields, e.g. in fraud detection, health care, insurance or cyber security [12]. In our research, it is used to detect defects in the interior of an object during an inline process. The starting point of the algorithm are measurements from intact, i.e. defect-free objects to generate a set of training data $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(m)}\} \subset \mathbb{R}^d$, $m \in \mathbb{N}$. Each data point has the same dimension $d \geq 1$ that describes the number of attributes called features. The attributes are represented by real numbers. We focus on random variables and a probability density function. For a profound theory of stochastic concepts, we recommend the work of Klenke and Shiryaev [55, 85]. For our algorithm, we assume that the data $\mathbf{x}^{(i)}$ are realizations of a real-valued random variable $X$. Motivated by the shape of the distribution of the data, see Figure 5.4, and the fact that it is appropriate to model the scattering of measurements as normal distributed, see [18], we suppose our data and its attributes to follow a Gaussian distribution. To obtain our probability density function

$$p : \mathbb{R}^d \to \mathbb{R}$$

we first determine the parameters of the Gaussian distribution for a given training data set. Then, we learn a threshold parameter $\epsilon^*$ and decide depending on this parameter and the probability density function, whether a new data point $\mathbf{x}_{\text{test}}$ is an outlier, i.e. an anomaly, or not. To learn the parameter $\epsilon^*$ we use a validation set and a decision function. Typical data of intact objects are characterized by a relatively large value of $p(\cdot)$, whereas anomalous data are identified by small values of $p(\cdot)$. The algorithm, which is presented in detail below, is based on [61] and [89].

**Notation 5.1.**
*We use the notation* $\mathbf{x}$ *if* $\mathbf{x} \in \mathbb{R}^d$ *with* $d > 1$ *throughout the entire chapter. If* $d = 1$*, we write* $x$*.*

Let $x \in \mathbb{R}$ be a realization of an $\mathcal{N}(\mu, \sigma^2)$-distributed random variable $X$ with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}$. We define the probability density function of the univariate Gaussian distribution by

$$p(x; \mu, \sigma^2) := \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{5.1}$$

and compute the values of $\mu$ and $\sigma^2$ with the help of the training data and the formulas

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \quad \text{and} \quad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)^2.$$

Accordingly, let $\mathbf{x}^{(i)} \in \mathbb{R}^d$ with $d \geq 2$ be a $\mathcal{N}(\mu, \Sigma)$-distributed random vector X, then we estimate the parameter $\mu \in \mathbb{R}^d$ and the covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ by

$$\mu = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}^{(i)} \quad \text{and} \quad \Sigma = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}^{(i)} - \mu)(\mathbf{x}^{(i)} - \mu)^\intercal \tag{5.2}$$

and characterize the probability density function of the multivariate Gaussian distribution by

$$p(\mathbf{x}; \mu, \Sigma) := \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^\intercal \Sigma^{-1}(\mathbf{x}-\mu)}. \tag{5.3}$$

We denote the determinant of $\Sigma$ by $|\Sigma|$. In the multivariate setting it is also possible to calculate the probability density function using the assumption that the features $x_1, x_2, ..., x_d$ are independent and identically distributed. Then, we use the functions $p(x_i; \mu_i, \sigma_i^2)$, $i = 1, ..., d$, and learn its parameters as in the one-dimensional case. Under this assumption the probability density function is calculated by

$$p(\mathbf{x}; \mu, \sigma^2) = p(x_1; \mu_1, \sigma_1^2) \cdot p(x_2; \mu_2, \sigma_2^2) \cdot ... \cdot p(x_d; \mu_d, \sigma_d^2).$$

After determining the probability density function in a first step, we learn in a second step a threshold parameter $\epsilon^*$ to identify whether a measurement is inconsistent or not. For this, we need a further supervised data set, the validation set. We label each measurement $\mathbf{x}_V^{(i)}$ with a value $y_V^{(i)} \in \{0, 1\}$, where $y_V^{(i)} = 1$ indicates an anomalous measurement and $y_V^{(i)} = 0$ a defect-free one. The validation set is given by

$$\{(\mathbf{x}_V^{(1)}, y_V^{(1)}), (\mathbf{x}_V^{(2)}, y_V^{(2)}), ..., (\mathbf{x}_V^{(l)}, y_V^{(l)}\}) \subset \mathbb{R}^d \times \{0, 1\}, \quad l \in \mathbb{N}.$$

For a probability density function $p$, any $\epsilon > 0$ and $\mathbf{x} \in \mathbb{R}^d$, we define the **decision function** $f$ by

$$f(\mathbf{x}, \epsilon) := \begin{cases} 1, & p(\mathbf{x}) < \epsilon \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

Using the validation set and a given value $\epsilon$, we compare the values of the decision function $f(\mathbf{x}_V^{(i)}, \epsilon)$ and the labels of the data $y_V^{(i)}$ for $i = 1, 2, ..., l$ to compute the **confusion matrix**

$$\mathbf{C} := \begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix} \in \mathbb{N}^{2\times 2}. \tag{5.5}$$

The matrix $\mathbf{C}$ characterizes the quality of the classification for a fixed $\epsilon$ and consists of four entries: the entry **true positive** (TP) counts the number of data points with

$$f(\mathbf{x}_V^{(i)}, \epsilon) = y_V^{(i)} = 1,$$

**true negative** (TN) represents the amount of data points with

$$f(\mathbf{x}_V^{(i)}, \epsilon) = y_V^{(i)} = 0,$$

**false positive** (FP) gives us the number of measurements with

$$f(\mathbf{x}_V^{(i)}, \epsilon) = 1 \neq y_V^{(i)} = 0,$$

and **false negative** (FN) sums up the measurements with

$$f(\mathbf{x}_V^{(i)}, \epsilon) = 0 \neq y_V^{(i)} = 1.$$

If the confusion matrix resembles a diagonal matrix, i.e. $FN \approx 0$ and $FP \approx 0$, we will know that the classification works well.

Via the entries of the confusion matrix, we define the two values **precision** (prec) and **recall** (rec) that depend for a fixed probability density function and an unchanged validation set on the threshold parameter $\epsilon$. It holds

$$prec(\epsilon) = \frac{TP}{TP + FP} \in [0, 1], \tag{5.6}$$

$$rec(\epsilon) = \frac{TP}{TP + FN} \in [0, 1]. \tag{5.7}$$

If the confusion matrix is diagonal, the two values take the right interval boundary, i.e. $prec(\epsilon) = rec(\epsilon) = 1$. Our classifier performs poorly if both values are close to zero. We use the $F_1$**-score**

$$F_1(\epsilon) = 2\frac{prec(\epsilon) \cdot rec(\epsilon)}{prec(\epsilon) + rec(\epsilon)} \tag{5.8}$$

to calculate the threshold parameter $\epsilon^*$. The $F_1$-score combines the precision and recall value by the harmonic mean. Finally, the threshold parameter $\epsilon^*$ is given as the value that maximizes the $F_1$-score. We set

$$\epsilon^* = \underset{\epsilon \in [0, p_{max}]}{\arg\max}\, F_1(\epsilon), \tag{5.9}$$

where $p_{max}$ represents the maximal value of the probability density function $p$.

In the last step, we access our test set

$$\{(\mathbf{x}_T^{(1)}, y_T^{(1)}), (\mathbf{x}_T^{(2)}, y_T^{(2)}), ..., (\mathbf{x}_T^{(l)}, y_T^{(l)})\} \subseteq \mathbb{R}^d \times \{0, 1\}, \quad l \in \mathbb{N}$$

to evaluate the anomaly detection algorithm. The test data set has the same composition as the validation data set and consists of supervised measured data. We calculate the confusion matrix and the $F_1$-score for the test set for the learned probability density function of the multivariate Gaussian distribution and the learned threshold parameter $\epsilon^*$. If the classifier fails, i.e. that the $F_1$-score is far away from the value 1, we must augment our data sets. We have to increase the number of measurements or include simulated data. If the $F_1$-score is close to one the classifier can be used to indicate whether an unknown data set includes irregularities. In this case, we identify our measurement $\mathbf{x}_{new}$ as an outlier, and consequently, intervene the inline process, if the value of the probability density function $p(\mathbf{x}_{new})$ is smaller than the threshold parameter $\epsilon^*$. The following Algorithm 5.2 summarizes the classification algorithm.

**Algorithm 5.2.** ***(Density-based anomaly detection)***

**INPUT:**

- *Training set* $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(m)}\} \subseteq \mathbb{R}^d$
- *Validation set* $\{(\mathbf{x}_V^{(1)}, y_V^{(1)}), (\mathbf{x}_V^{(2)}, y_V^{(2)}), ..., (\mathbf{x}_V^{(l)}, y_V^{(l)})\} \subseteq \mathbb{R}^d \times \{0, 1\}$
- *Test set* $\{(\mathbf{x}_\mathrm{T}^{(1)}, y_\mathrm{T}^{(1)}), (\mathbf{x}_\mathrm{T}^{(2)}, y_\mathrm{T}^{(2)}), ..., (\mathbf{x}_\mathrm{T}^{(l)}, y_\mathrm{T}^{(l)})\} \subseteq \mathbb{R}^d \times \{0, 1\}$

- *Measured data* $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(J)}\} \subseteq \mathbb{R}^d$

**STEP 1:**

*Consider the training set as realizations of an* $\mathcal{N}(\mu, \Sigma)$*-distributed random variable,* $\mu \in \mathbb{R}^d$*,* $\Sigma \in \mathbb{R}^{d \times d}$*, and estimate the probability density function (5.3) by (5.2).*

**STEP 2:**

*Use the validation set and the decision function (5.4) to construct the confusion matrix* $\mathbf{C}$ *(5.5) and to find an optimal threshold parameter* $\epsilon^* \geq 0$ *by finding the maximizer (5.9) of the* $F_1$*-Score (5.8).*

**STEP 3:**

*Evaluate the algorithm by using the test set. If the evaluation fails, enhance the training data set.*

**CLASSIFICATION:**

*for* $j = 1 : J$

*if* $p\big(\mathbf{x}^{(j)}; \mu, \Sigma\big) < \epsilon^*$ *then set* $y^{(j)} = 1$ *(an outlier/defect has been detected)*

*else* $y^{(j)} = 0$ *(no outlier/defect detected).*

## 5.2 Terahertz measuring system and data set

The original THz tomography system displayed in Figure 5.1 was constructed at the Plastic Center in Würzburg. The system aims to simulate the procedure of an inline monitoring process. All measured objects and the measured data used for the anomaly detection algorithm were generated and recorded by employees of the SKZ. Figure 5.2 shows the setup of the THz measuring system from Figure 5.1 as a cross-section. The figures illustrate that the system consists of an emitter that also acts as a receiver and two further receivers. The transmitter $E_1$ emits electromagnetic radiation of a frequency between 0.12 and 0.17 THz. At the same time, it measures reflection data, see receiver $R_1$ in Figure 5.2. A second receiver $R_2$ is located opposite the emitter to collect information about the transmission process. The third receiver is placed near the second one to register signals caused by the refraction of the radiation. The measuring equipment is fixed on a turntable that rotates around the object of interest. In Figure 5.1, one can also notice two lenses causing the Gaussian beam of the electromagnetic radiation and the observed object in the center of the system. If we compare this setting with Figure 1.2, we can see that the number of receivers is significantly reduced. Only limited financial resources were available for the AiF project, which prevented an increase in the number of receivers. Further, we neglected the modeling of the lenses in our previous investigations of the reduced inverse problem of THz tomography in Chapter 2 to 4.

While the investigated object is fixed, it is possible to shift the turntable vertically with a step size of 1 mm. For every vertical position $k$ a complete 360° rotation

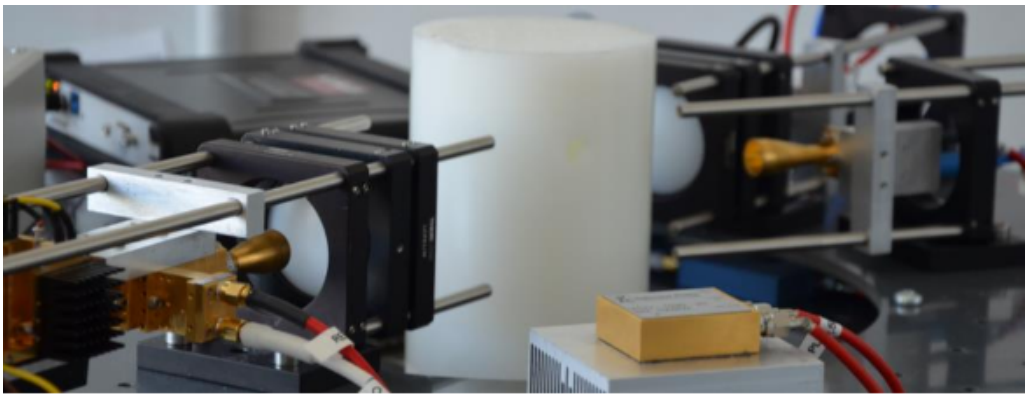


Figure 5.1: THz tomography system at Plastic Center (SKZ), Würzburg

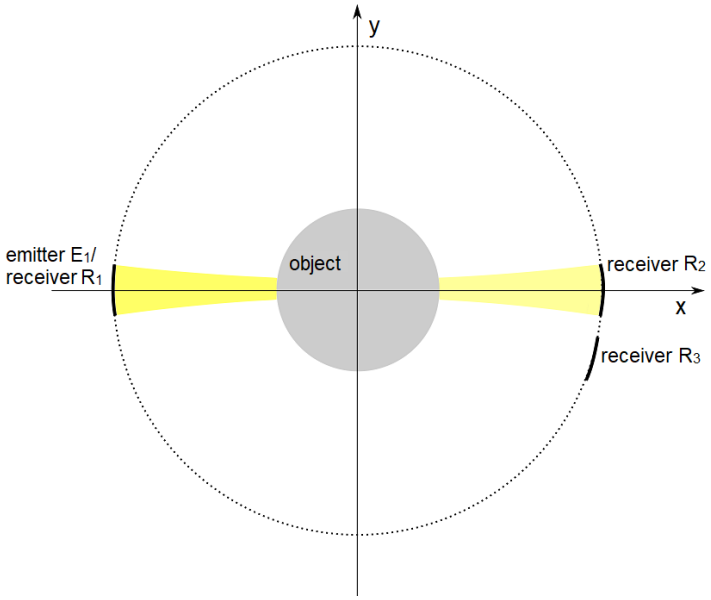
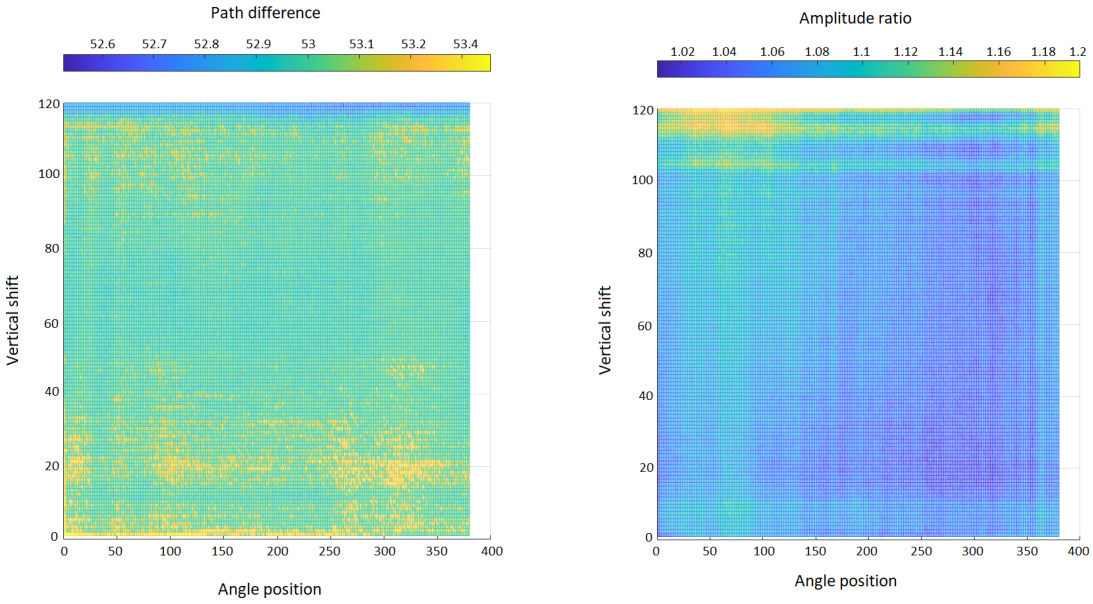



Figure 5.2: Schematic THZ tomograph

is realized in $Z$ steps. Consequently, we scan two-dimensional slices of the entire three-dimensional object by shifting and rotating the turntable. We obtain a d-dimensional matrix $\mathbf{X} = (\mathbf{x}_{kz})_{k=1,\ldots,K,z=1,2,\ldots,Z}$ for every measured object. Each single d-dimensional entry $\mathbf{x}_{kz}$ of the matrix is composed of five features: Receiver $R_1$ informs about phase and absorption information of the reflection process. The second receiver $R_2$ registers the same information about the transmission process. Figure 1.3 illustrates a measured horizontal shift,i.e. the phase shift, and a vertical shift, i.e. the absorption loss, of the amplitude in red compared to the reference signal in blue. The fifth feature belongs to receiver $R_3$. Since no reference signal is given for this receiver, we only detect whether a signal is present or not and how intense it is. Due to the calibration measurement where no refraction occurs, receiver $R_3$ only provides information about the amplitude of the signal.

To generate the training, validation and test data used for our anomaly detection, solid pipes made of polyethylene were constructed. The pipes have a diameter of 10 cm and are characterized by its refractive index of about $n = 1.53$ and its absorption coefficient $\alpha = 0.06$ cm$^{-1}$. We first scanned solid pipes without defects. The

$K \times Z$-matrices for the features path difference and amplitude ratio in transmission are represented in Figure 5.3. We see the angular position on the $x$-axis and the vertical shift on the $y$-axis. One rotation consists of $Z = 380$ angular positions and the turntable is shifted in $Z = 120$ steps.



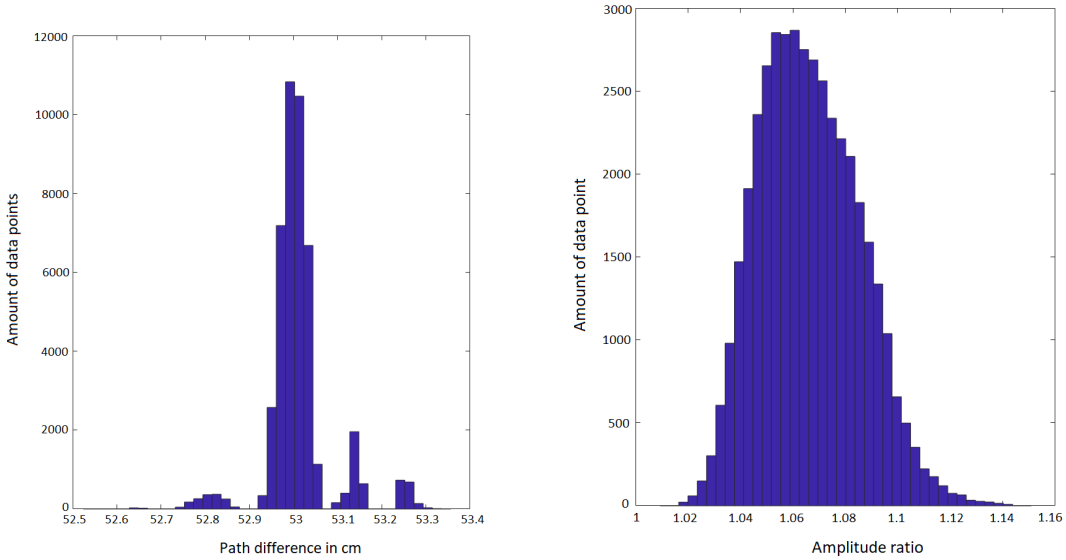

(a) (b)

Figure 5.3: Data points for the path difference (a) and the amplitude ratio (b) of a solid pipe without defects in transmission

In a second step, we manufactured vertical and horizontal holes in the pipes. We filled some of them with materials like oil, metal or polyethylene of a different refractive index. We thus create synthetic defects. All in all, we end up with a data set containing 220400 five-dimensional data points of measurements from intact solid pipes and 105965 data points from defect samples. To generate our training set $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(m)}\} \subseteq \mathbb{R}^5$ we use 60% of the intact measurements. The remaining intact data points are distributed evenly among the test and validation data so that the data sets are each composed of 20% of the typical data and 50% of the anomalous measurements.

In Figure 5.4, we present an example of the distribution of the data points that originate from measurements with defect-free solid pipes. We see the registered values at receiver $R_2$ opposite the transmitter. Indeed, the distribution resembles a Gaussian distribution concerning both the path difference on the left side as well as the amplitude ratio on the right side. Similar results can be found for receivers $R_1$ and $R_3$. The latter only provides measurements of the amplitude as mentioned before.

Comparing our THz measuring system and our data set to a real inline process, we have to specify one difference: The material moves continuously instead of being fixed. In our investigation, we simulate the movement of the material by the step-wise shift of the measuring system. In reality, for example in an extrusion process, the emitter and the receivers describe a different trajectory: Using a horizontally and




(a) (b)

Figure 5.4: Distribution of the data points registered at receiver $R_2$

vertically fixed measuring system that rotates around an object that continuously moves through the measuring system, we obtain a helical trajectory relative to the material. So, instead of having two-dimensional data for $K$ slices, we deal in a real inline process with three-dimensional data since the object is not shifted but moving continuously.

## 5.3 Numerical results for measured data sets

In our first investigation, we use the data set described in section 5.2 to evaluate our classification algorithm. We set $d = 5$ by dealing with the five features measured by our three receivers, such that we utilize information about the refraction, reflection and transmission of the radiation. After learning the parameters of the algorithm and evaluating them with the test set, we apply the classification to investigate an unknown pipe. We show that our application is suitable for an inline process monitoring based on THz radiation.

According to Algorithm 5.2, we first use the training set to compute the parameters of the multivariate Gaussian distribution (5.3) with $d = 5$. We obtain

$$\mu = \begin{pmatrix} 53.601002 \\ 1.015691 \\ 0.139608 \\ 0.010417 \\ 83.159275 \end{pmatrix}$$

and

$$\Sigma = \begin{pmatrix} 0.258068 & -0.012679 & 0.003472 & 0.000218 & -0.355831 \\ -0.012679 & 0.001625 & -0.000057 & -0.000001 & -0.012545 \\ 0.003472 & -0.000057 & 0.002129 & -0.000013 & -0.034956 \\ 0.000218 & -0.000001 & -0.000013 & 0.000016 & -0.000867 \\ -0.355831 & -0.012545 & -0.034956 & -0.000867 & 3.299866 \end{pmatrix}.$$

Afterwards, we estimate the threshold parameter $\epsilon^* = 2.260130$ via optimizing the $F_1$-score for the validation set. In the third step of Algorithm 5.2, we calculate the confusion matrix

$$\mathbf{C} = \begin{pmatrix} 52982 & 434 \\ 0 & 43266 \end{pmatrix}.$$

for the given test set and the corresponding $F_1$-score $F_1(\epsilon^*) = 0.995921$. We evaluate that each outlier is found and that only 434 data points are incorrectly detected as an anomaly though they do not represent a defect. Consequently, our $F_1$-score tends to one, which is the desired result. The learning process is complete.

In the next step, we scan an unknown solid pipe that may contain defects. We measure the five features mentioned above and apply the classification part of the anomaly detection algorithm. For this purpose, we calculate the probability density function for our measured data $\{\mathbf{x}_M^{(1)}, \mathbf{x}_M^{(2)}, ..., \mathbf{x}_M^{(J)}\} \subseteq \mathbb{R}^5$ with the learned expected values and covariance matrix. We have $J = K \cdot Z$, where again $K = 315$ indicates the number of vertical shifts and $Z = 380$ the number of angular positions per slice. Figure 5.5 visualize the results $\mathbf{Y} = (y_{kz})_{k=1,...,K,z=1,...,Z}$ of the classification: A yellow pixel corresponds to an outlier, i.e. $y_{kz} = 1$, while a blue pixel shows no irregularity, i.e. $y_{kz} = 0$. We note that the classification algorithm detects two defects that both appear twice in Figure 5.5 because of doing a full 360° rotation. Consequently, the defects are scanned in 180° intervals. Reading the plot from top to bottom, we see that the first horizontal lines are marked completely yellow. The transition between air and material is mapped in these lines. They are followed by an area of 40 cm representing an anomaly. After some blue horizontal lines, a second yellow area appears that indicates a second defect. For the lower 20 cm of the pipe no deviation from the standard is measured.

By comparing our measurements with the exact dimension of the object, we notice that the pipe indeed contains two defects: a vertical damage of 4 cm from above and a second one, a lateral hole of 8 mm in the lower half. Our investigation aimed to find anomalous areas in the material, not to characterize the defect or to determine the exact dimension. Considering the results, we can state that our goals were achieved. Based on the presented results, a full reconstruction of the interior of the object can be investigated, see Chapter 3. By reconstructing only critical areas, it is again possible to save a lot of time which is an important factor in inline monitoring.

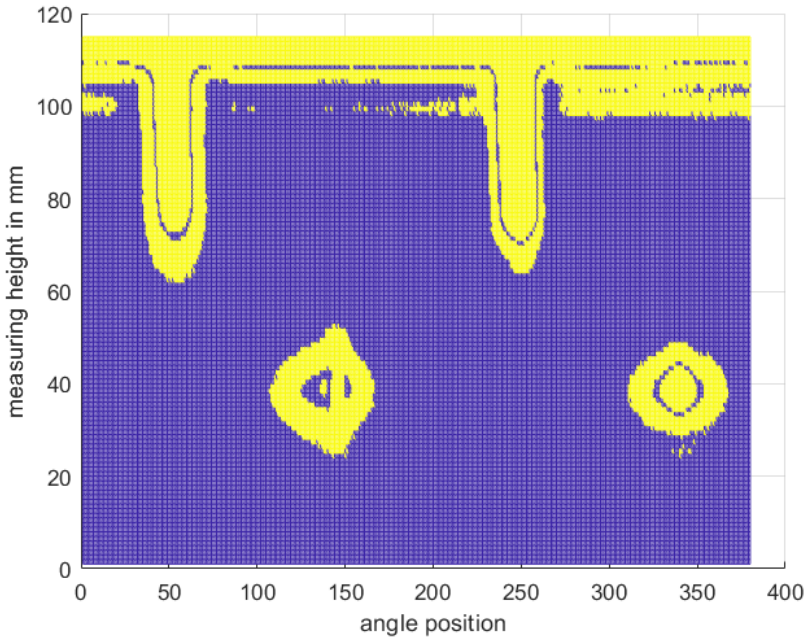



Figure 5.5: Investigation of an unknown pipe using the classification algorithm

Regarding the computational time of Algorithm 5.2, it is obvious that the total time depends on the amount of data used for the learning process. In our case, we need about three seconds (Intel Core i7-8565U processor). The most time-consuming part is the solution of the optimization problem in step two depending on the step size used for $\epsilon$. Furthermore, the amount of correct measurements used for the learning process influences the performance significantly. We use the maximum number of measurements generated at the SKZ.

## 5.4 Numerical results based on partly simulated data

In the following section, we refer to the physical model from Chapter 2. We reduce our previously used five-dimensional data set to one feature, the temporal information of receiver $R_2$, and involve simulated data instead. We validate our simulation by comparing synthetic with real measured data. Afterwards, we enlarge our data set for the one-dimensional case. Using the eikonal equation, we generate time-of-flight measurements of intact and damaged objects without further time-consuming real-time measurements. Moreover, no further objects with defects need to be generated. To demonstrate an added value of the synthetic data, we compare the anomaly detection algorithm using the hybrid data set with the anomaly detection algorithm only applied to the measured data for the one-dimensional case.

As derived in the previous chapters the travel time $T$ of electromagnetic radiation can be described by the eikonal equation (2.19) for high frequencies from a theoretical point of view. Because of our measuring system, we focus again on a domain $\Omega \subseteq \mathbb{R}^2$. Additionally, we consider a suitable constraint $T(x_0, y_0) = 0$ for an initial value

$(x_0, y_0)^\intercal \in \partial\Omega$ on the boundary $\partial\Omega$ of the domain $\Omega$, such that we result in the partial differential equation (3.1) with the initial condition (3.7).

Figure 5.6 represents a simulation of the travel time of the THz radiation emitted at position $(x_0, y_0)^\intercal = (-15, 0)^\intercal$. The partial differential equation is solved by the FMM, see Chapter 3. We take into account the usual geometry of the THz beam generated by the lenses in our measuring system that focus the electromagnetic radiation. Therefore, we deal with a Gaussian intensity profile that can be characterized by its Rayleigh zone $z_0$ and its waist size $w_0$, see [88, 93]. When solving the eikonal equation for the simulated data we restrict our domain to the Gaussian beam, whose radius $w(z)$ is given by

$$w(z) = w_0 \sqrt{1 + \left(\frac{z}{z_0}\right)^2}$$

where $z$ describes the distance from the waist of the beam. A transformation for $z \neq 0$ leads to

$$w(z) = w_0 z \sqrt{\frac{1}{z^2} + \frac{1}{z_0^2}}.$$

For $z \gg z_0$ we see that $w(z)$ approaches a straight line which was assumed in our simulation in Figure 5.6 to approximate the Gaussian profile.

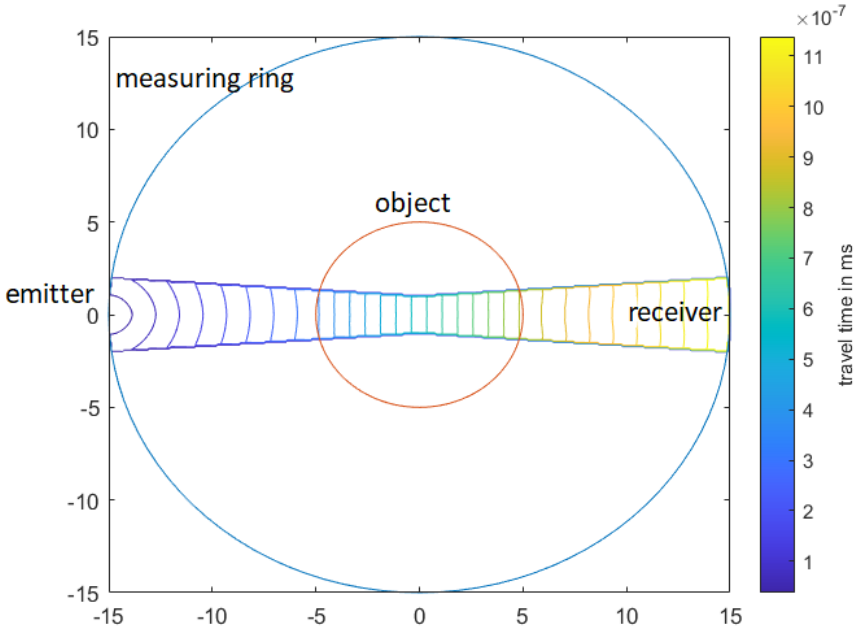



Figure 5.6: Simulation of the travel time of the THz radiation by solving the eikonal equation and approaching the Gaussian profile

In the next step, we validate the physical model and the resulting simulated data by comparing it with real measurements of our inline measuring system. We scan a polyethylene solid pipe with a diameter of 10 cm and a homogeneous refractive index of $n = 1.53$, and simultaneously, simulate the process via the FMM implemented in MATLAB. Figure 5.7 visualizes the result: We see the path difference of the THz

radiation on the $y$-axis as a function of the angular position for simulated data in blue and measured data in red. We add a uniform noise of 3 % to our simulated data to account for measurement and observation errors. Indeed, the two data series yield comparable results by comparing a 360° measurement of one slice. The mean value is a good benchmark to compare the simulated and measured data. We obtain a mean value of $s_{sim} = 53.119081$ mm for the simulated data and a mean value of $s_{real} = 53.602705$ mm for the measured ones. It results a relative error $\Delta_{rel}$ of

$$\Delta_{rel} = \frac{s_{real} - s_{sim}}{s_{real}} \cdot 100\ \% = 0.9022\ \%,$$

indicating a good consistency of simulation and measurement.

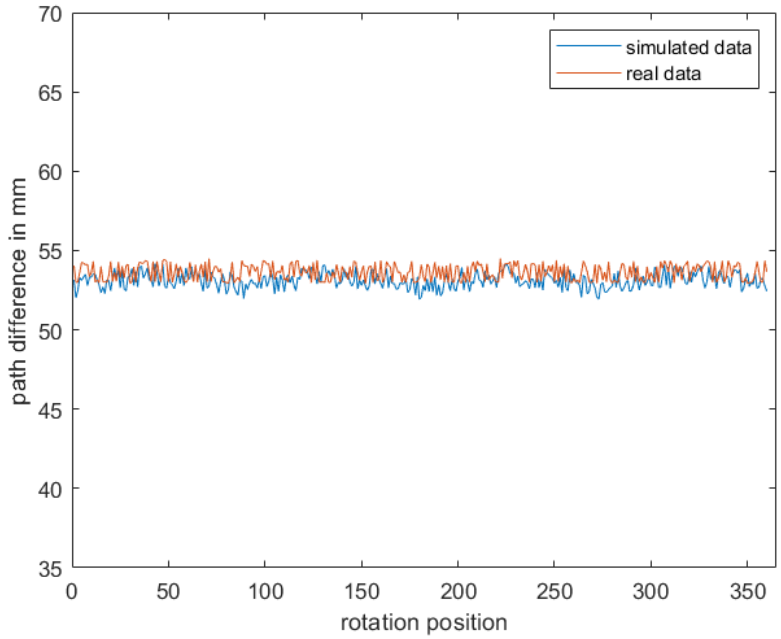



Figure 5.7: Comparison of the simulated and the measured data for a solid pipe with a diameter of 10 cm and a refractive index of $n = 1.53$

The simulation of the data provides a basis for a more economic application of our anomaly detection algorithm. By the generation of synthetic data the manufacturing process of the material, in our case the solid pipes, can be limited. It is possible to simulate defects instead of constructing them after the extrusion process. Especially for complex inline products, such as window frames for example, a virtual object design is a great advantage. We can increase our data set without time-consuming measurements. We focus on a one-dimensional setting because the solution of the eikonal equation only simulates one of the five features. So, we augment our data set described in section 5.2 for the feature travel time measured at receiver $R_2$. Analogous to the previous subsection, we perform a one-dimensional anomaly detection concerning the feature travel time of receiver $R_2$. We subdivide our hybrid data as in Section 5.2. After learning the parameters of the univariate Gaussian distribution $p(x; \mu, \sigma^2)$ and the threshold parameter $\epsilon^*$, we investigate the unknown pipe. The parameters of equation (5.1) are given by

$$\mu = 53.546907 \text{ and } \sigma^2 = 0.282028.$$

The optimization problem of Step 2 in Algorithm 5.2 yields the optimal threshold parameter $\epsilon^*$ that leads to a maximal $F_1$-score of 0.489209. The $F_1$-score for the hybrid data set is significantly lower than the $F_1$-score from Section 5.3 because of the reduced number of features. Figure 5.8 depicts the matrix $\mathbf{Y}$ of the anomaly detection of the unknown pipe using the learned parameters of the one-dimensional Gaussian distribution resulting from the hybrid data set. As in the plot of Figure 5.5, we indicate defects by yellow pixels and normal measurements by blue pixels.

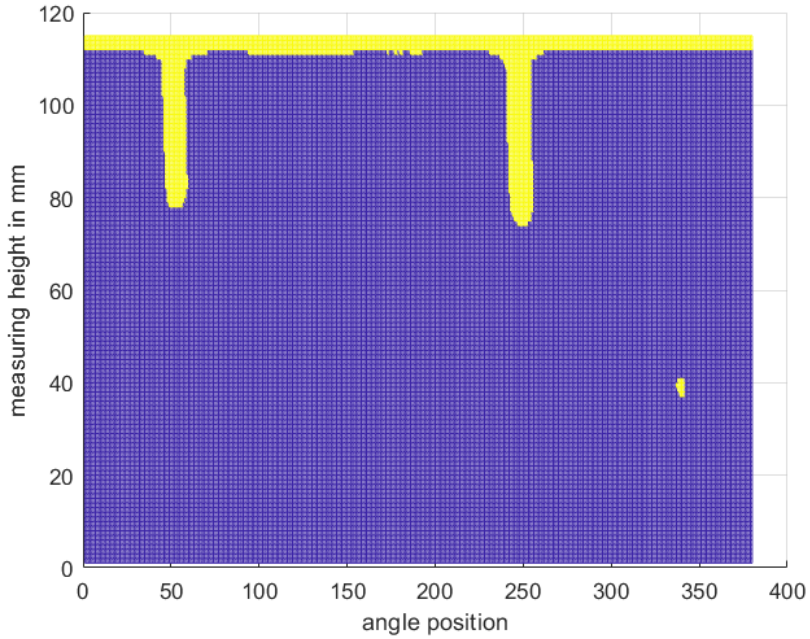



Figure 5.8: One-dimensional anomaly detection for the unknown pipe using a hybrid data set

To validate the one-dimensional anomaly detection with a hybrid data set and to demonstrate an added value, we also learn the parameters only by the real measured data of receiver $R_2$, i.e. we neglect the simulation. Figure 5.9 presents the result: It is obvious that in the hybrid setting the vertical damage is detected more reliably than in the setting without simulations. The second defect area in the lower part of the pipe is weakly detected in both cases because we consider only one feature.

Comparing the results of this section to the results of Section 5.3, we conclude that the best classification was archived by the multidimensional setting. However, a tendency to over-sensitivity is evident that would appear with a further increase in the number of features. The investigation from this section serves as a proof of concept to show the potential of synthetic data in machine learning applications. A more accurate anomaly detection is already possible for one feature using a hybrid data set instead of only measured data. For a multivariate approach with simulated data, further physical models have to be investigated to simulate the remaining four features. One possibility to simulate the intensity of the THz beam measured by receiver $R_2$ is given by Tepe et al. [88], where a modified ART was developed to reconstruct the absorption coefficient and the refractive index. Another challenge will

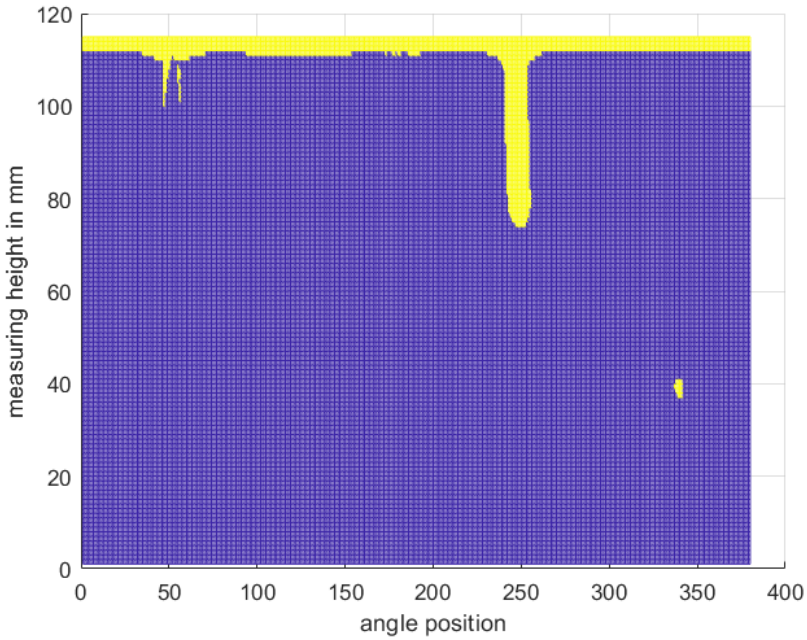



Figure 5.9: One-dimensional anomaly detection set for the unknown pipe using only measured data

be to classify the defect with regard to the material properties, shape and size. To this end, it is conceivable to amplify the anomaly detection algorithm towards a deep learning approach using deep neural networks. For the generation of simulated data, it will be vital to investigate the nature of typical defects in practical applications.

# 6 Conclusion and outlook

Motivated by an AiF project in cooperation with the SKZ in Würzburg, the thesis dealt with the examination of the inverse problem of THz tomography including data-driven enhanced methods. We investigated a completely new physical approach in the context of THz tomography, the eikonal equation, to simulate the propagation of electromagnetic radiation in the THz range. The initial project is discussed in the last chapter of the thesis, however, represents the starting point of our investigations. Building on existing research, especially the one of Wald et al. [93], we derived a high-frequency approximation of the wave equation resulting in the eikonal equation. During project work, we developed an anomaly detection in inline monitoring of plastics and ceramics. We used simulated as well as real measured data provided by the SKZ. The main results of the project are firstly, that the eikonal equation is a practicable physical model to simulate the propagation of electromagnetic waves for high frequencies, and secondly, that we could detect defects in an extrusion process of plastics. Following these findings, we continued the study of the inverse problem of THz tomography using the eikonal equation in this thesis. While in the project the focus was on detecting defects, we expanded the research in our thesis to reconstruct the interior of an object for the two-dimensional case. We take into account the aspect of time efficiency and energy saving that are the most important factors in industry in addition to the accuracy of the reconstruction.

So, in the first part of the thesis, we introduced the topics of THz tomography and machine learning with regard to historical developments and existing research and presented some basis. We established the eikonal equation as the physical model and the reduced inverse problem of reconstructing the refractive index $n$ from THz tomographic measurements.

In the next part, we discussed the mathematical investigations of the inverse problem. We examined the non-linear forward operator including the parameter-to-solution operator, the trace operator and a linear observation operator. We derived an expression for the Fréchet derivative and its adjoint, enabling us to compute the gradient of the least squares functional. This gradient is crucial in the reconstruction techniques we utilized resulting in the Landweber method. The analysis then focused on the numerical solution of the forward operator and the adjoint of the Fréchet operator. We introduced the FMM to solve our forward problem and a method based on the difference quotient to calculate the adjoint of the linearization. Finally, we implemented the Landweber iteration to reconstruct the refractive index of tested objects with noisy synthetic data. The results are suitable for a qualitative and quantitative evaluation. So, we solved the reduced inverse problem of the two-dimensional THz tomography.

In the following section of the thesis, we presented two enhanced methods, a partially learned Landweber and a sparse learned Landweber method. Both methods are data-driven. The main focus here was time efficiency and energy saving. The forward operator derived in the first part, especially the solution of it using numerical methods, is time-consuming. By substituting the original forward operator with a CNN, we achieved the second objective of the thesis: We could accelerate the reconstruction process accepting an increase in the relative error of the reconstruction and focusing on quantitative results. Further, we could save weights and consequently energy in the learning process of the CNN by using a regularization technique in the optimization process of the parameters resulting in a sparse network. The sparse network used for the solution of the forward operator performed almost as well as the dense network.

In conclusion, all the objectives outlined in the motivation of the thesis have been successfully achieved. The research combines mathematical theory with real-time problems. The influence of industrial factors is noticeable throughout the entire investigation, especially in the last two chapters. This thesis represents one of the first scientific approaches in applying data-driven methods to THz tomography. The results obtained can be transferred to other physical models or inverse problems. Future research endeavors will focus on adapting, refining and optimizing the approach pioneered in this thesis. Finally, the next few lines will provide an overview of potential extensions and improvements for the thesis.

First of all, one can switch the physical model. The eikonal equation was chosen since the Helmholtz equation causes problems in particular for higher frequencies. Models using the wave equation fully capture the inverse problem in spatial and temporal dimensions but are numerically complex to implement. So, using the eikonal equation, we reduce the complexity of the underlying problem and only focus on the reconstruction of the refractive index neglecting the absorption coefficient. It remains the task of current research to investigate the problem using the wave equation. Here, the use of artificial intelligence could improve efficiency: It is imaginable that the solution of the wave equation is learned via a neural network, according to the solution of the eikonal equation in our thesis, using both real measurement data and simulated data. Again, the numerical effort would be reduced in the reconstruction process identically to the investigations in the underlying thesis.

To improve the results of Chapter 4, it is necessary to amplify the data set, ideally with real measurement data. In contrast to the AiF project presented in Chapter 5, it is possible and affordable nowadays to increase the number of receivers on the entire measurement ring and thus generate real measurements that match our reconstruction algorithm. A wide range of varied data can reduce overfitting. The network could react more flexibly to complex data. To handle an increasing data set, the hardware has to be adapted. More computer power is unavoidable and the switch to a server has to be made. The CNN can then be expanded in width and depth. It would be interesting to investigate whether more computer power automatically results in a better reconstruction. Note, that the approximation theorems postulate a more precise learning process of the forward operator by amplifying the amount of parameters within the network. Regarding the sparsity ansatz, it would be interesting

to compare further regularization terms with the $\ell^1$-regularization, for example the group lasso presented in equation (4.12). Furthermore, in our investigations, we pursued a dense-to-sparse approach. It is also possible to start with a sparse net, resulting in a sparse-to-sparse approach.

Staying in the setting of the thesis, it is left to substitute the evaluation of the adjoint of the linearized forward operator $F'(n)^*$ by a second network. This would lead to a further acceleration of the reconstruction process accompanied by a further increase in the reconstruction error. Two approaches are conceivable at this point: On the one hand, the numerical solution of the partial differential equation given in Theorem 3.21 can be learned according to the learning process already completed for the forward operator. Thus, in Algorithm 3.27 a second network $\Psi_{\theta^{**}}$ can be placed with the optimal parameters $\theta^{**}$. We obtain for the reconstruction

$$v_{i+1}^{\delta} = v_i^{\delta} - \omega \Psi_{\theta^{**}}(\Phi_{\theta^*}(v_i^{\delta}) - y^{\delta}), \quad i = 0, ..., k^* - 1.$$

On the other hand, it is possible to use the existing network $\Phi_{\theta^*}(\cdot)$, respectively $\Phi_{\tilde{\theta}^*}(\cdot)$, and calculate its gradient for the implementation of the learned Landweber method. We obtain

$$v_{i+1}^{\delta} = v_i^{\delta} - \omega \Phi'_{\theta^*}(v_i^{\delta})^*(\Phi_{\theta^*}(v_i^{\delta}) - y^{\delta}), \quad i = 0, ..., k^* - 1.$$

The difference between both approaches is, that once a second learning process is required generating a new approximation error and once the existing error is transferred to the gradient.

Furthermore, the error analysis and the convergence of the learned Landweber iteration is an open question. Up to now, the network has been selected via a trial and error process and the learned Landweber method is stopped depending on the relative error. A time-consuming process is performed, consisting of generating the network, learning the parameters and finally testing the network within a Landweber iteration. If the reconstruction fails, a new learning process will be started or the training set will be changed. In general, there are plenty of mathematical investigations left regarding neural networks. In the underlying case, it would be desirable to estimate the error

$$||\epsilon(v)|| = ||\Phi_{\theta^*}(v) - F(v)||$$

of the learned network and its influence of the convergence of the reconstruction. From a mathematical point of view, it would be an enormous step to obtain a convergence statement similar to the discrepancy principle and Remark 3.23.

Finally, there is the possibility of connecting the learned Landweber method with a post-processing to improve the reconstruction. For this, it is necessary to include a new network mapping the reconstruction to a reworked version to better identify the defects and material parameters. For example, one could use an variational autoencoder network [54]. Additionally, further information about the object to be reconstructed could be included, e.g. the probability of some defects.

To conclude the thesis, it can be generally stated that the use of data-driven enhanced methods in THz tomography is possible and can be applied in the field of

non-destructive testing for dielectric materials. The ideas and results presented in this thesis can be applied to many other areas whenever a partial differential equation is solved in a time-consuming manner. Overall, the potential of THz tomography is far from exhausted and there are many starting points for further development to make it suitable for effective industrial use.

# A Notations

| | |
|---|---|
| $F$ | forward operator |
| $\mathcal{D}(F)$ | domain of $F$ |
| $\mathbf{x}$ | an element of $\mathbb{R}^n$ |
| $\|\| \cdot \|\|_Y$ | norm of the vector space $Y$ |
| $\delta$ | noise |
| $y^\delta$ | noisy data |
| $n(\mathbf{x})$ | refractive index at position $\mathbf{x}$ |
| $v(\mathbf{x})$ | material dependent velocity of the radiation at position $\mathbf{x}$ |
| $c_0$ | speed of light |
| $N$ | number of receivers |
| $T(\mathbf{x})$ | travel time at position $\mathbf{x}$ |
| $S$ | parameter-to-solution operator |
| $\gamma$ | trace operator |
| $\mathcal{Q}$ | observation operator |
| $E_\nu$ | surface of receiver $\nu$ |
| $e_\nu$ | senor characteristics |
| $F'(v)^*$ | linearized adjoint operator |
| $M^2$ | number of pixels |
| $J$ | number of angular positions |
| $k^*$ | stopping index |
| $R^\delta$ | residual |
| $\epsilon_v$ | relative error |
| $\omega$ | relaxation parameter for Landweber |

| | |
|---|---|
| $\mathcal{X}$ | domain set of a network |
| $\mathcal{Y}$ | label set of a network |
| $\hat{\mathbf{y}}$ | output vector of a network |
| $W_i$ | weight matrix connecting the $i$-1-th layer with the $i$-th layer |
| $b_i$ | bias in the $i$-th layer |
| $\tau(\cdot)$ | activation function |
| $L$ | number of hidden layers |
| $N_T$ | number of training data |
| $\mathbf{X}_T$ | input training data |
| $\mathbf{Y}_T$ | output training data |
| $\mathbf{X}_V$ | input validation data |
| $\mathbf{Y}_V$ | output validation data |
| $\theta$ | weights in a network |
| $\tilde{\theta}$ | weights in a sparse network |
| $\Theta$ | parameter space of all weights in the network |
| $J(\cdot,\cdot,\cdot)$ | cost functional |
| $l(\cdot,\cdot)$ | loss function |
| $R(\cdot)$ | regularization functional of the cost function |
| $\theta^*$ | optimal weights in a network |
| $\Phi_{\theta^*}(\cdot)$ | learned forward operator |
| $\alpha$ | regularization parameter for the cost functional |
| $\zeta$ | threshold parameter for the sparsity |
| $\Phi_{\tilde{\theta}^*}(\cdot)$ | learned sparse forward operator |
| $\tilde{\theta}^*$ | optimal weights in a sparse network |
| $p(\cdot;\mu,\sigma^2)$ | probability density function of the univariate Gaussian distribution |
| $p(\cdot;\mu,\Sigma)$ | probability density function of the multivariate Gaussian distribution |
| $\epsilon^*$ | optimal threshold parameter of the anomaly detection |
| $f(\cdot,\epsilon)$ | decision function with threshold parameter $\epsilon$ |
| $C$ | confusion matrix |

# B Some supplementary mathematical theory

**Definition B.1** (**Sigmoidal**)**.**
*A function $\tau^* : \mathbb{R} \to \mathbb{R}$ is called **sigmoidal** if the following conditions hold:*

*a)* $\lim_{z\to\infty} \tau^*(z) = 1.$

*b)* $\lim_{z\to-\infty} \tau^*(z) = 0.$

**Definition B.2** (**Discriminatory**)**.**
*Let $I = [0,1]^m$, $\mu$ a Borel measure on $I$. A function $\tau^* : \mathbb{R} \to \mathbb{R}$ is called **discriminatory** if*

$$\int_I \tau^*(w^T x + b)\, d\mu(x) = 0 \text{ for all } w \in \mathbb{R}^m, b \in \mathbb{R}$$

*then $\mu = 0$.*

**Lemma B.3.**
*Any continuous sigmoidal function $\tau^* : \mathbb{R} \to \mathbb{R}$ is discriminatory.*

**Theorem B.4** (**Universal approximation theorem**, [14])**.**
*Let $I = [0,1]^m$ and $\tau : \mathbb{R}^n \to \mathbb{R}^n$ component-wise defined by a continuous discriminatory function $\tau^* : \mathbb{R} \to \mathbb{R}$. Then*

$$\{\varphi_n : \mathbb{R}^m \to \mathbb{R} : \varphi_n(\mathbf{x}) = w^T \tau(W\mathbf{x} + b),\ W \in \mathbb{R}^{n\times m},\ b, w \in \mathbb{R}^n\}$$

*is dense in $C(I)$.*

**Lemma B.5.** *Let $I \subset \mathbb{R}^m$, $\tau^* : \mathbb{R} \to \mathbb{R}$ with*

$$\tau^*(z) = ReLU(z).$$

*Then $\tau^*$ is discriminatory, i.e. we obtain the universal approximation property of ReLU networks.*

*Proof.* Let $\mu$ be a Borel measure and for all $w \in \mathbb{R}$ and $b \in \mathbb{R}$ it holds

$$\int_I ReLU(wx + b)\, d\mu(x) = 0.$$

We construct a sigmoid bounded, continuous (Borel measurable) function by subtracting two ReLU-functions with different parameters.

Consider

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x \in [0,1] \\ 1 & x > 1 \end{cases}$$

Any function $\tilde{f}(x) := f(wx + b)$ can be described by

$$\tilde{f}(x) = ReLU(wx + b) - ReLU(wx + b - 1)$$

For any $w \in \mathbb{R}$, $b \in \mathbb{R}$ we have

$$\int_I f(wx + b)\, d\mu(x) = \int_I ReLU(wx + b)\, d\mu(x) - \int_I ReLU(wx + b - 1)\, d\mu(x) = 0.$$

We obtain $\mu = 0$ because $f$ is discriminatory.

We have $\Sigma_n(f) = span\{f(wx + b) : w \in \mathbb{R}^n, b \in \mathbb{R}\}$and $\Sigma_1(f)$ is dense in $C([0,1])$ by the proof. It follows that $\Sigma_n(f)$ is dense in $C([0,1])^n$
(See '*An overview of artificial neural networks for mathematicians*' Guilhoto, 2018)
□

**Lemma B.6** (**Product rule for divergence**).
*For a scalar-valued function $\varphi : \mathbb{R}^n \to \mathbb{R}$ and a vector-valued function $v : \mathbb{R}^n \to \mathbb{R}^n$, we have the following identity:*

$$\operatorname{div}(\varphi v) = \varphi \operatorname{div} v + v \nabla \varphi.$$

*Proof.* Let $\varphi : \mathbb{R}^n \to \mathbb{R}$ and $v : \mathbb{R}^n \to \mathbb{R}^n$

$$\begin{aligned} \operatorname{div}(\varphi v) &= \sum_{l=1}^{n} \frac{\partial(\varphi v_k)}{\partial x_k} \\ &= \sum_{l=1}^{n} \Big( \varphi \frac{\partial v_k}{\partial x_k} + \frac{\partial \varphi}{\partial x_k} v_k \Big) \\ &= \varphi \operatorname{div} v + v \nabla \varphi \end{aligned}$$

□

**Theorem B.7.** *The parameter-to-solution operator $S$ is non-linear.*

*Proof.* Let us assume that $S$ is linear. Then it holds

$$S(\lambda v) = \lambda S(v).$$

for a scalar $\lambda \in \mathbb{R}$. So, we have that $S(\lambda v) = T_{\lambda v}$ solves

$$\begin{aligned} |\nabla T_{\lambda v}(x,y)|^2 &= \frac{1}{\lambda^2 v^2(x,y)} \\ \Leftrightarrow \lambda^2 |\nabla T_{\lambda v}(x,y)|^2 &= \frac{1}{v^2(x,y)} \\ \Leftrightarrow |\nabla T_{\lambda^2 v}(x,y)|^2 &= \frac{1}{v^2(x,y)} \end{aligned}$$

for the given starting condition. We obtain

$$S(v) = T_{\lambda^2 v}$$

which is a contradiction to $S(v) = T_v$ and the uniqueness of the operator. □

**Theorem B.8.**
*The linear filter given by Definition 4.7 represents a discrete convolution with the kernel $H^*$ creating from vertical and horizontal mirroring of $H$.*

*Proof.*

$$\begin{aligned}(I * H^*)(u,v) &= \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} I(u-i, v-j)\cdot H^*(i,j)\\ &= \sum_{(i,j)\in R_H} I(u-i, v-j)\cdot H^*(i,j)\\ &= \sum_{(i,j)\in R_H} I(u-i, v-j)\cdot H(-i,-j)\\ &= \sum_{(i,j)\in R_H} I(u+i, v+j)\cdot H(i,j)\end{aligned}$$

□

**Theorem B.9** (**Properties of the convolution**, [23], p. 107)**.**
*For $f, g, h, g_1, g_2 : \mathbb{R}^n \to \mathbb{C}$, $a \in \mathbb{C}$, we have the following properties:*

*a) Commutativity:*

$$f * g = g * f.$$

*b) Linearity:*

$$a(f * g) = (af) * g = f * (ag).$$
$$f * (g_1 + g_2) = f * g_1 + f * g_2.$$

*c) Associativity:*

$$(f * g) * h = f * (g * h).$$

# C Source code

**Code C.1.**

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 10, kernel_size = 5)
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.fc1 = nn.Linear(320,50)
        self.fc2 = nn.Linear(50,10)

    def forward(self, x):
        x = F.relu(F.max_pool2d(self.conv1(x),2))
        x = F.relu(F.max_pool2d(self.conv2(x),2))
        x = x.view(-1,320)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return F.log_softmax(x, dim = -1)
```

**Remark C.2.**
*The code for Chapter 3, 4 and 5 is uploaded on github, see* ***https://github.com/tschus71/THZ****.*

# Bibliography


[1] J Adler and O Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[2] C C Aggarwal. Neural networks and deep learning. *Springer*, 10(978):3, 2018.

[3] M Z Alom, T M Taha, C Yakopcic, S Westberg, P Sidike, M S Nasrin, M Hasan, B C Van Essen, A A Awwal, and V K Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3):292, 2019.

[4] I Asimov. I, robot. *A collection of short stories*, 1940.

[5] M Benning and M Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, 2018.

[6] S S Biswas. Potential use of chat gpt in global warming. *Annals of biomedical engineering*, 51(6):1126–1127, 2023.

[7] B Blaschke, A Neubauer, and O Scherzer. On convergence rates for the iteratively regularized Gauss-Newton method. *IMA Journal of Numerical Analysis*, 17(3): 421–436, 1997.

[8] M Born and E Wolf. *Priciples of Optics*. Cambridge University Press, 2013.

[9] W Burger and M J Burge. *Digitale Bildverarbeitung: Eine Algorithmische Einführung Mit Java*. Springer-Verlag, 2009.

[10] T I Burgess, K Howard, E Steel, and E L Barbour. To prune or not to prune; pruning induced decay in tropical sandalwood. *Forest Ecology and Management*, 430:204–218, 2018.

[11] A Chacon and A Vladimirsky. Fast two-scale methods for eikonal equations. *SIAM Journal on Scientific Computing*, 34(2):A547–A578, 2012.

[12] V Chandola, A Banerjee, and V Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[13] M G Crandall and P-L Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

[14] G Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[15] W Demtröder. *Experimentalphysik 2*. Springer Spektrum, 2013.

[16] P Dhar. The carbon impact of artificial intelligence. *Nat. Mach. Intell.*, 2(8): 423–425, 2020.

[17] A C Doyle. *A Study in Scarlet*. Penguin Group, 2011.

[18] K Eden and H Gebhard. *Dokumentation in der Mess-und Prüftechnik*. Springer, 2014.

[19] H W Engl, M Hanke, and A Neubauer. *Regularization of inverse problems*. Springer Science & Business Media, 1996.

[20] L C Evans. *Measure theory and fine properties of functions*. Routledge, 2018.

[21] L C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

[22] J Frankle and M Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[23] R C. Gonzalez and R E. Woods. *Digital image processing*. Addison-Wesley, Reading, Mass. [u.a.], reprinted with corrections edition, 1993. URL `http://www.gbv.de/dms/ilmenau/toc/164078770.PDF`.

[24] R G González-Acuña and H A Chaparro-Romo. *Stigmatic Optics*. IOP Publishing, 2020. URL `https://dx.doi.org/10.1088/978-0-7503-3463-1`.

[25] I Goodfellow, Y Bengio, and A Courville. *Deep learning*. MIT press, 2016.

[26] W H Guss and R Salakhutdinov. On universal approximation by neural networks with uniform guarantees on approximation of infinite dimensional maps. *arXiv preprint arXiv:1910.01545*, 2019.

[27] J Hadamard. *Lectures on Cauchy's problem in linear partial differential equations*, volume 15. Yale university press, 1923.

[28] M Haenlein and A Kaplan. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California management review*, 61 (4):5–14, 2019.

[29] M Haltmeier, R Kowar, and M Tiefentaler. Data-driven Morozov regularization of inverse problems. *arXiv preprint arXiv:2310.14290*, 2023.

[30] M Hanke. A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse problems*, 13(1):79, 1997.

[31] M Hanke, A Neubauer, and O Scherzer. A convergence analysis of the landweber iteration for nonlinear ill-posed problems. *Numerische Mathematik*, 72(1):21–37, 1995.

[32] T Hartwick, D Hodges, D Barker, and F Foote. Far infrared imagery. *Applied Optics*, 15(8):1919–1922, 1976.

[33] J Hauck, D Stich, P Heidemeyer, M Bastian, and T Hochrein. Untersuchung der Bauteileigenschaften von Kunststoffprodukten mittels zeitaufgelöster Terahertz-Spektroskopie. *DGZfPJahrestagung, Dresden*, 2013.

[34] J Hirschle. *Deep Natural Language Processing: Einstieg in Word Embedding, Sequence-to-Sequence-Modelle und Transformer mit Python*. Carl Hanser Verlag GmbH Co KG, 2022.

[35] B Hofmann. *Mathematik inverser Probleme*. Teubner Stuttgart, 1999.

[36] K Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[37] B B Hu and M C Nuss. Imaging with terahertz waves. *Optics letters*, 20(16): 1716–1718, 1995.

[38] D H Hubel and T N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574, 1959.

[39] D H Hubel and T N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[40] S Hubmer and R Ramlau. Convergence analysis of a two-point gradient method for nonlinear ill-posed problems. *Inverse Problems*, 33(9):095004, 2017.

[41] S Hubmer, A Ploier, R Ramlau, P Fosodeder, and S van Frank. A mathematical approach towards THz tomography for non-destructive imaging. *arXiv preprint arXiv:2010.14938*, 2020.

[42] F Jarre and J Stoer. *Optimierung*. Springer, 2004.

[43] B Jin and P Maass. Sparsity regularization for parameter identification problems. *Inverse Problems*, 28(12):123001, 2012.

[44] Q Jin. On a regularized Levenberg-Marquardt method for solving nonlinear inverse problems. *Numerische Mathematik*, 115(2):229–259, 2010.

[45] Q Jin and U Tautenhahn. On the discrepancy principle for some Newton type methods for solving nonlinear inverse problems. *Numerische Mathematik*, 111 (4):509–558, 2009.

[46] Harald K. *Grundlagen der elektromagnetische Feldtheorie*. Springer Spektrum, 2018.

[47] B Kaltenbacher and T T N Nguyen. Discretization of parameter identification in PDEs using neural networks. *Inverse Problems*, 38(12):124007, 2022.

[48] B Kaltenbacher, A Neubauer, and O Scherzer. *Iterative regularization methods for nonlinear ill-posed problems*. de Gruyter, 2008.

[49] B Kaltenbacher, F Schöpfer, and T Schuster. Iterative methods for nonlinear ill-posed problems in banach spaces: convergence and applications to parameter identification problems. *Inverse Problems*, 25(6):065003, 2009.

[50] B Kaltenbacher, T Schuster, and A Wald. *Time-dependent Problems in Imaging and Parameter Identification*. Springer, 2021.

[51] C Y Kao, S Osher, and J Qian. Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations. *Journal of Computational physics*, 196(1):367–391, 2004.

[52] R Kimmel and J A Sethian. Computing geodesic paths on manifolds. *Proceedings of the national academy of Sciences*, 95(15):8431–8435, 1998.

[53] D P Kingma and J Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[54] D P Kingma and M Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[55] A Klenke. *Wahrscheinlichkeitstheorie*, volume 1. Springer, 2006.

[56] A Krogh and J Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[57] S Kružkov. Generalized solutions of the Hamilton-Jacobi equations of eikonal type. i. formulation of the problems; existence, uniqueness and stability theorems; some properties of the solutions. *Mathematics of the USSR-Sbornik*, 27(3):406, 1975.

[58] Y LeCun, C Cortes, C Burges, et al. MNIST handwritten digit database, 2010.

[59] M Leshno, V Y Lin, A Pinkus, and S Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[60] S Leung and J Qian. An adjoint state method for three-dimensional transmission traveltime tomography using first-arrivals. *Journal of Computational physics*, 4 (1):249–266, 2006.

[61] K Limthong. Real-time computer network anomaly detection using machine learning techniques. *Journal of Advances in Computer Networks*, 1(1):126–133, 2013.

[62] B Littau, J Tepe, G Schober, S Kremling, T Hochrein, and P Heidemeyer. Entwicklung und evaluierung der potenziale von Terahertz-Tomografie-Systemen, 2016.

[63] P Maass. Deep learning for trivial inverse problems. In *Compressed Sensing and Its Applications*, pages 195–209. Springer, 2019.

[64] M Mayr, C Meiser, T Hochrein, M Bastian, T Schuster, and G Schober. *Terahertz-Tomografie in der Extrusion von Kunststoffhalbzeugen*. Shaker Verlag, 2021.

[65] K G Mehrotra, C K Mohan, and H Huang. *Anomaly detection principles and algorithms*, volume 1. Springer, 2017.

[66] C Meiser, T Schuster, and A Wald. A classification algorithm for anomaly detection in terahertz tomography. In *International Conference on Large-Scale Scientific Computing*, pages 393–401. Springer, 2021.

[67] C Meiser, A Wald, and T Schuster. Learned anomaly detection with terahertz radiation in inline process monitoring. *Sensing and Imaging*, 23(1):30, 2022.

[68] F Minolts, S Kremling, D Stich, P Heidemeyer, M Bastian, and T Hochrein. 3D-Prüfung von Faserverbundwerkstoffen mittels vollelektronischer Terahertz-Systeme, 2014.

[69] D M Mittleman. Twenty years of terahertz imaging. *Optics express*, 26(8): 9417–9431, 2018.

[70] Y E Nesterov. A method of solving a convex programming problem with convergence rate $o(k^{-2})$. In *Doklady Akademii Nauk*, volume 269, pages 543–547. Russian Academy of Sciences, 1983.

[71] A Neubauer. Tikhonov regularization of nonlinear Ill-posed problems in Hilbert scales. *Applicable analysis*, 46(1-2):59–72, 1992.

[72] A Neubauer. On Landweber iteration for nonlinear ill-posed problems in Hilbert scales. *Numerische Mathematik*, 85(2):309–328, 2000.

[73] M A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015.

[74] D Nüßler and J Jonuscheit. Terahertz based non-destructive testing (ndt). *Technisches Messen*, 2020.

[75] D Nüßler and J Jonuscheit. Terahertz based non-destructive testing (ndt). *tm-Technisches Messen*, 88(4):199–210, 2021.

[76] S Otmar. A convergence analysis of a method of steepest descent and a two–step algorothm for nonlinear ill–posed problems: A two–step algorothm for nonlinear ill–posed problems. *Numerical functional analysis and optimization*, 17(1-2): 197–214, 1996.

[77] S Rawas. Chatgpt: Empowering lifelong learning in the digital age of higher education. *Education and Information Technologies*, pages 1–14, 2023.

[78] A Rieder. *Keine Probleme mit inversen Problemen: eine Einführung in ihre stabile Lösung*. Springer-Verlag, 2003.

[79] D Rothermel and T Schuster. Solving an inverse heat convection problem with an implicit forward operator by using a projected quasi-Newton method. *Inverse Problems*, 37(4):045014, 2021.

[80] E Rouy and A Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992.

[81] S Scardapane, D Comminiello, A Hussain, and A Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[82] J A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[83] J A Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999.

[84] J A Sethian and A Vladimirsky. Fast methods for the eikonal and related Hamilton-Jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences*, 97(11):5699–5703, 2000.

[85] A N Shiryaev. *Probability-1*, volume 95. Springer, 2016.

[86] J Stoer and R Bulirsch. *Introduction to Numerical Analysis*. Springer, 2002.

[87] E Strubell, A Ganesh, and A McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[88] J Tepe, T Schuster, and B Littau. A modified algebraic reconstruction technique taking refraction into account with an application in terahertz tomography. *Inverse Problems in Science and Engineering*, 25(10):1448–1473, 2017.

[89] A Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 2020.

[90] R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[91] G Tzydynzhapov, P Gusikhin, V Muravev, A Dremin, Y Nefyodov, and I Kukushkin. New real-time sub-terahertz security body scanner. *Journal of Infrared, Millimeter, and Terahertz Waves*, 41(6):632–641, 2020.

[92] L Vierus and T Schuster. Well-defined forward operators in dynamic diffractive tensor tomography using viscosity solutions of transport equations. *arXiv preprint arXiv:2111.05722*, 2021.

[93] A Wald and T Schuster. Tomographic terahertz imaging using sequential subspace optimization. In *New Trends in Parameter Identification for Mathematical Models*, pages 261–290. Springer, 2018.

[94] D Werner. *Funktionalanalysis*. Springer-Verlag, 2007.

[95] P Wimmer and A P Mehnert, Jand Condurache. Dimensionality reduced training by pruning and freezing parts of a deep neural network: a survey. *Artificial Intelligence Review*, pages 1–39, 2023.

[96] M Yuan and Y Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.

[97] S Zhong. Progress in terahertz nondestructive testing: A review. *Frontiers of Mechanical Engineering*, 14(3):273–281, 2019.

[98] D-X Zhou. Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, 48(2):787–794, 2020.

[99] W Zouaghi, MD Thomson, K Rabia, R Hahn, V Blank, and H Roskos. Broadband terahertz spectroscopy: principles, fundamental research and potential for industrial applications. *European Journal of Physics*, 34(6):S179, 2013.