DISSERTATION

FULL WAVEFORM INVERSION FOR ULTRASOUND COMPUTED TOMOGRAPHY IN THE DETERMINISTIC AND BAYESIAN SETTINGS

Submitted by

Scott Ziegler

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2022

Doctoral Committee:

Advisor: Jennifer Mueller

Wolfgang Bangerth
Margaret Cheney
Marlis Rezende

ABSTRACT

FULL WAVEFORM INVERSION FOR ULTRASOUND COMPUTED TOMOGRAPHY IN THE DETERMINISTIC AND BAYESIAN SETTINGS

Ultrasound computed tomography (USCT) is a noninvasive imaging technique in which acoustic waves are sent through a region and measured after transmission and reflection in order to provide information concerning that region. There are many reconstruction techniques for USCT which rely on linearization of the total pressure field, but this simplifying assumption often causes a loss of resolution and poor results in highly reflective media. Full waveform inversion (FWI) is a method popularized by the geophysical community which makes use of entire time-dependent pressure measurements and repeated solutions of the nonlinear wave equation. Due to this lack of linearization, FWI is able to produce high-fidelity sound speed reconstructions, albeit at a steep computational cost.

In this dissertation, we explore the use of the FWI techniques in both the deterministic and Bayesian settings. For the deterministic case, an algorithm for FWI is derived which makes use of the adjoint method for the computation of functional derivatives and the software package k-Wave for the solution of the nonlinear wave equation. This algorithm is tested on numerical breast and lung phantoms for a variety of regularization functionals and parameters, where it displays an excellent ability to reconstruct the size and shape of inhomogeneities. For the lung phantom, a novel application of a structural similarity index regularization term is used with an Electrical Impedance Tomography prior to speed convergence and improve organ boundary delineation. In the Bayesian setting, a Metropolis-adjusted Langevin FWI algorithm is proposed and tested on a simplified breast phantom, with an emphasis on reducing computational expense. Preliminary results from this test show promise for future research on FWI in the Bayesian framework.

ACKNOWLEDGEMENTS

There is hardly space here to properly acknowledge all of the people who contributed to this work over the last six years. Without question this dissertation is a result of input from many different people, and I am eternally grateful for their help. I would like to thank and acknowledge Dr. Mueller for all of her patience, support, and knowledge. She has been a wonderful advisor and I hope to continue working and learning from her in the future. My committee has played a large role in the completion of this work as well. I have taken four different classes from Dr. Bangerth, and he has been an invaluable mentor and terrific educator throughout my time at Colorado State. Dr. Cheney's input has been illuminating and her very large part in my job search will never be forgotten. Dr. Rezende has provided significant input from a different perspective and has been a very positive influence on this work. I thank my committee for their time, wisdom, and encouragement.

I would like to acknowledge my supervisor and colleagues at the Naval Research Laboratory for their patience and support as I have finished my dissertation over the past year. I look forward to putting the knowledge obtained from this work to use in the future.

My family has been incredibly supportive of me through the highs and lows of this work. Their love and assistance are always appreciated. A big thank you to Charlotte as well, who has been with me first hand to experience my joy and struggle throughout this process. I love you all.

Finally, I must thank my friends for their input both intellectually and emotionally. The past six years would have been far less enjoyable without Adam, Alex, Brittany, Dean, Dustin, Elliot, John, Joshua, Juanita, Levi, and Tom. Thank you for everything.

DEDICATION

To the lifelong friendships that were made concurrently with this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

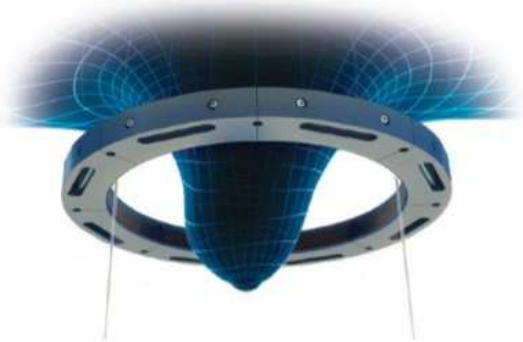
Chapter 1

Introduction

Ultrasound Computed Tomography (USCT) is a noninvasive and non-ionizing imaging technique in which high frequency sound waves are emitted toward an object of unknown sound speed and measured after transmission and reflection in order to provide information about the boundary or acoustical properties of that object. The most common acoustical properties of interest are refraction and sound speed [5]. By discovering these attributes we are able to form a qualitative image of the object as well as obtain quantitative information which can be compared against materials with similar refractions and/or sound speeds. Thus we can learn about not only the shape of the object(s) but also its material properties. USCT has applications in several fields including material sciences [6], geophysics [7], and medical imaging [8]. Our focus is on the application of USCT to the medical field, where it has grown in popularity as a method for imaging the breast [9, 10], brain [11], and lungs [12–14]. USCT provides many intriguing possibilities with regards to the diagnosis of certain diseases and maladies, including the diagnosis of cancer [15]. In addition, there is hope in the medical community that USCT can eventually replace x-ray mammography as the primary method of imaging the breast or at the very least supplement traditional x-ray images. X-ray mammography introduces ionizing radiation to the body and is also very painful. USCT alternatives for x-ray mammography are already being explored and in some cases are already in use. These alternatives involve the patient resting their breast in a water bath for imaging instead of undergoing painful compression. A ring transducer within the bath then images a two dimensional (2D) “slice" of the breast and is free to move up and down in order to produce a three dimensional image. An example of the SoftVue machine from Delphinus medical technologies [16] for imaging the breast is shown in figure 1.1a.

While USCT has been researched as a viable method for imaging the breast for years, it has more recently become a method of interest for imaging the brain. Since the skull is a strong acoustic scatterer, most reconstruction techniques for imaging the brain using ultrasound are done in

(a) The SoftVue ultrasound machine from Delphinus technologies, figure reproduced from [16].



(b) A ring array of transducers surrounds the breast, figure reproduced from [16].

Figure 1.1: An example of a machine currently in use to image the breast using USCT.

three dimensions [11]. As computing power has increased, these three-dimensonal reconstructions have become plausible. An exciting and relatively new possibility is using USCT to image the lungs, which until recently was thought to be impossible. However, it was recently discovered [12] that there is a low frequency band in the ultrasonic range (between 1kHz and 1MHz) for which the lungs become transparent to acoustic waves. Many lung diseases require constant monitoring, and since USCT involves no ionizing radiation, it is an intriguing option for these situations. A typical setup for imaging the lungs would likely look similar to the ring array transducer in figure 1.1b, except that the transducers would be placed directly on the thorax. The reconstruction techniques for imaging the breast and lungs would not need to differ greatly as they are both primarily done in 2D space, as opposed to techniques for the brain, which are performed in three dimensions. Thus the method that we present in this thesis can be used for imaging either the breast or lungs with little modification needed.

With many other imaging modalities already in use and well-established, it is worth considering why we would want to develop more and/or better methods for USCT. While these other medical imaging modalities are often good enough for many situations, USCT has some advantages which make it worth exploring. Using transmitted and reflected acoustic waves (as is done in USCT) provides more information concerning the object in question and can thus lead to higher resolution images as compared with B-mode (brightness mode) ultrasound, which measures only reflected

waves. In B-mode ultrasound [17], a linear array of transducers is used to focus high frequency (1-20MHz) sound waves at a point of interest in the body. These sound waves are reflected at points of changing acoustic attenuation and measured again by the linear array of transducers. Then, using information concerning both the time delay of the signal and the attenuation in the signal, the acoustic attenuation of the organs and tissues can be reconstructed. The method for this reconstruction is similar to the techniques presented in section 2.1. B-mode ultrasound is a useful technique in many diagnostic circumstances, the most well-known of which is viewing a fetus *in vitro*. While B-mode ultrasound is sufficient for these scenarios, USCT has the advantage of greater resolution and the ability to provide information concerning more acoustical properties than just attenuation.

Computerized tomography (CT, or x-ray tomography) has long been in use and involves sending x-rays through the human body and measuring the attenuation in the signal at a receiver location. Reconstruction algorithms for this modality are well established [18] and can provide sharp images, but CT introduces ionizing radiation to the body. This means there is danger in an individual being imaged too often with this method and precludes it from being used on a patient who requires frequent scans. Another common imaging modality is Magnetic Resonance Imaging (MRI), which does not use ionizing radiation but instead uses magnetic fields to produce an image of the human organs. MRI provides good results, but as shown in [19], USCT is able to image breast tissues in a comparable manner to MRI at a fraction of the cost. This bodes well for the use of USCT in the detection of breast cancer. It is unlikely that USCT will replace these other imaging modalities entirely, but it could serve as an intriguing supplement which proves to be the correct choice for certain medical situations. An example of one of these situations would be in imaging women with dense breasts. X-ray mammography performs poorly in this case, but in [20] the authors show that USCT has the ability to accurately diagnose tumors even in women with dense breasts.

Figure 1.2 shows a slightly more detailed 2D setup for USCT for the purpose of explaining the data collection process. An ultrasonic sound wave (or pressure wave) is sent through one

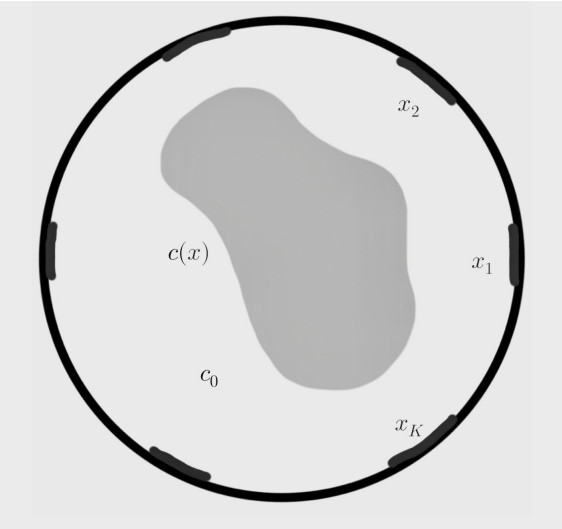


Figure 1.2: A sample setup for imaging an object with sound speed $c(\boldsymbol{x})$ using USCT.

transducer at location $\boldsymbol{x}_k$, and a full ring of transducers surrounds the object to be imaged. The object has sound speed $c(\boldsymbol{x})$ with the assumption that the background sound speed is homogeneous and given by the value c_0. This pressure wave then interacts with the medium by either passing through the medium with some change in wave speed, reflecting off of the object, or a combination of both. Each transducer (including the transducer acting as a source) then measures the returning sound wave to give a collection of K time dependent signals where K is the total number of transducers. This process is repeated for the remaining transducers in order to produce a $K \times K$ set of measured pressure waves. Typically every transducer will act as a source at some point in an effort to "illuminate" the object with sound waves from as many angles as possible. We note that figure 1.2 only shows 6 transducers for simplicity, but in a real-world situation the number of transducers should be much higher. The time dependent pressure waves measured on each transducer comprise the data for the reconstruction problem in USCT. Using this data to reconstruct the structure and acoustical properties of the domain is the primary goal of USCT and indeed will be our objective throughout this dissertation.

There are three primary ways in which this work extends on or adds novelty to the current literature. The first is in the use of full waveform inversion (FWI) techniques to image a numerical lung phantom in the low-frequency ultrasonic region. FWI has been used to produce reconstruc-

tions of both the breast and brain, but to our knowledge it has not been tested on a simulation of a cross section of the human thorax before now. In addition, while lower frequency ultrasonic ranges are used for FWI in the geophysics community [21], those same ranges are seldom found in the medical imaging regime. Thus our use of frequencies in the kHz range for FWI simulations is a useful examination for the medical ultrasound community.

The second originality is the use of a structural similarity index (SSIM) regularization term in USCT. In [22], the authors use an SSIM structure term to regularize computed tomography reconstructions at different energy levels. We use that same regularization term with an Electrical Impedance Tomography (EIT) conductivity reconstruction, which is the first use of that regularization term across imaging modalities. This represents a novel use of both the SSIM regularization term and an EIT prior within USCT reconstructions. Our investigation of three different regularization parameters (Tikhonov, total variation, and SSIM-EIT) for a numerical lung phantom represents an excellent starting point for future work on FWI lung imaging.

The final extension within this work is an investigation of USCT in the Bayesian framework. While the vast majority of work on USCT takes place in the deterministic setting, there have been authors who have explored the problem in the Bayesian setting [23, 24]. In [23], the authors employ the Bayesian approximation error approach and utilize an algorithm similar to the one used in this thesis, but their algorithm does not form a Markov chain and is not intended to produce samples from a posterior distribution. Instead, the authors use the Bayesian approximation error approach to test different approximate forward solvers. This is a different goal altogether from ours. The authors of [24] consider the Bayesian approach for B-mode ultrasound, not tomographic ultrasound. There has also been work investigating FWI from a Bayesian perspective, but that work has been almost entirely in the geophysics community [25, 26]. Our investigation of the Bayesian approach to FWI is focused on medical applications, which has yet to be thoroughly explored. In addition, we propose using a Metroplis-Adjusted Langevin algorithm to obtain FWI results, which is a novel application of that algorithm.

The remainder of the thesis is structured as follows: chapter 2 gives an overview of several of the most popular reconstruction techniques currently in use for USCT as well as provides some background on inverse acoustic scattering problems in general. This will naturally lead to a discussion of FWI, which we describe and expand on in chapter 3. Results of FWI on a numerical breast phantom with different regularization parameters follows. Chapter 3 also features a novel technique for using an electrical impedance tomography (EIT) prior in FWI and displays numerical results for this approach on a lung phantom. In chapter 4 we present a Bayesian approach to FWI and describe a novel algorithm for USCT in the Bayesian framework. Preliminary numerical results for this Bayesian algorithm are shown for a test phantom which approximates the simplified cross section of a human breast. Finally, chapter 5 concludes with a summary of our findings and possibilities for future research.

Chapter 2

Survey of Current Literature

The following sections in chapter 2 give a sample of popular approaches to solving the inverse problem in USCT. This chapter is by no means meant to be comprehensive, and inquisitive readers should make use of the references in each section if they would like to learn more.

2.1 Ray-based Methods

Some of the earliest and still most popular methods for determining the sound speed of an inhomogeneous medium using ultrasonic waves can be broadly grouped under the name “ray-based methods". These methods treat the sound waves as 1D rays passing directly from an emitter to a receiver and then use either amplitude or time-of-flight (TOF) techniques to produce a reconstruction. TOF methods were originally popularized by researchers working on CT. Representing x-rays as 1D objects which pass in a straight line from emitter to receiver is an accurate model, so in CT scans TOF reconstructions are very accurate. Representing acoustic waves in this manner is not as accurate, and as a result ray-based methods for USCT produce reconstructions with many artifacts and low resolution [5]. However, TOF methods for USCT are computationally cheap to implement and can still produce an image which captures the prominent features of the inhomogeneous sound speed map c. For this reason, ray-based methods are common for imaging regions which do not require high resolution, or as a starting point for algorithms which can attain higher resolution but are otherwise very time laborious.

Consider the measurement setup portrayed in figure 2.1. A pressure wave is sent through the object we wish to image (the target) and we measure either the resulting pulse on only the transmitting linear array (reflection USCT) or measure on both the transmitting array and the opposite array (transmission USCT). The quantity being measured on the array differs depending on whether we are using an amplitude ray-based method or a TOF ray-based method. In the amplitude method, the measurements are the ratios of the amplitudes of the pulses when they leave the transmitter to

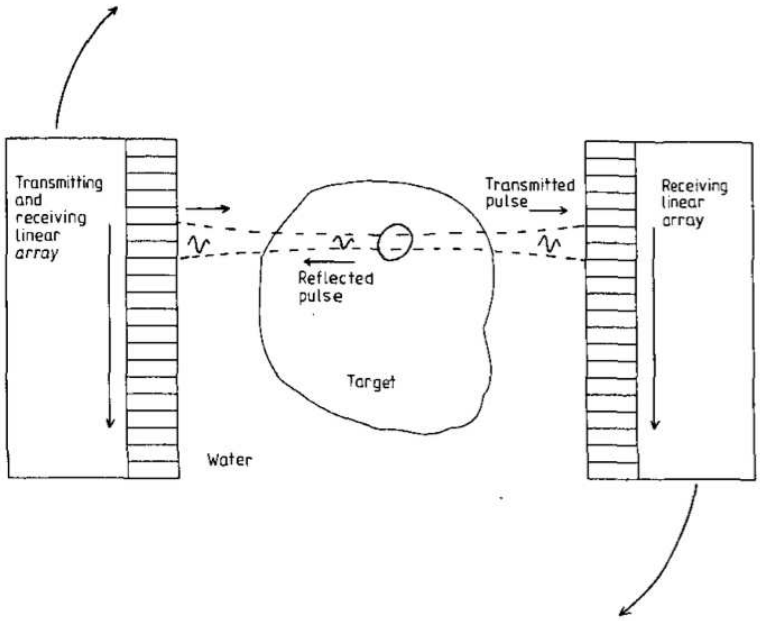


Figure 2.1: A sample measurement setup for ray-based methods, figure reproduced from [1].

their amplitudes when they are received. These measurements can then be combined to reconstruct the attenuation coefficient [27]. The other approach is to measure the time of flight taken for each pulse to travel between the emitter and the receiver. These measurements are then used to reconstruct the sound speed distribution of the target. For reasons presented in [1], reconstructing the sound speed distribution using TOF measurements is preferable to attenuation reconstructions and has thus been more widely applied in USCT research.

We now provide a brief overview of TOF reconstructions. If we assume that we can discretize our target into an $N \times N$ grid, then we may assign the sound speed for each pixel c_j as the average sound speed throughout that pixel. We then assume the area outside of the target but within the transmitting and receiving array has constant sound speed c_w. This is shown in figure 2.2.

Now if l_j is the length of the ultrasonic ray as it traverses the jth pixel, then the total time t_t for the pulse to travel from A to B is

$$t_t = \sum_{j=1}^{N^2} \left(\frac{l_j}{c_j} \right) + \frac{d - \sum_{j=1}^{N^2} l_j}{c_w}.$$

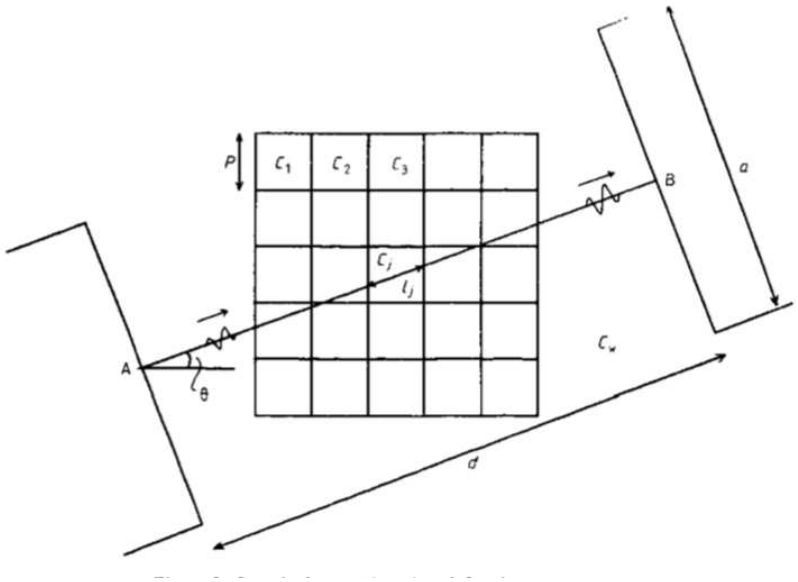


Figure 2.2: Setup for a TOF reconstruction, figure reproduced from [1].

If we also have the time t_w it takes for an ultrasonic ray to travel from A to B along the same path with no target present (no inhomogeneity), then we can subtract this time from t_t to get

$$\begin{aligned} t_t - t_w &= \sum_{j=1}^{N^2} l_j \left(\frac{1}{c_j} - \frac{1}{c_w} \right) \\ &= \sum_{j=1}^{N^2} l_j \eta_j, \end{aligned}$$

where η_j is known as the refractive index of the jth pixel. Note that if we know c_w, reconstructing η_j is the same as reconstructing c_j. We thus proceed by measuring $t_t - t_w$ for a number of different parallel rays across the object to form a TOF projection image.

Given the TOF projection setup as described above, there are a number of methods for reconstructing an image of the target object. By far the most popular method is filtered backprojection [28], in which the projection is convolved with an appropriate filter and then retransmitted along the same direction in which it was obtained. There has been extensive work on the mathematics behind filtered backprojection, and in the interest of brevity we leave those descriptions to the following sources: [29–32]. As stated earlier, TOF reconstructions in USCT are appealing due

to their simplicity and computational efficiency. However, when an acoustic wave interacts with a strongly inhomogeneous medium there is likely to be an interplay between the pressure wave and the medium including transmission, reflection, as well as constructive and destructive interference within the pressure wave itself. For this reason, alternatives to ray-based methods have been and continue to be explored.

2.2 Diffraction Tomography

Another popular approach for reconstructing the sound speed of an object is to use a technique called diffraction tomography [2,33,34]. The setup for a single diffraction tomography experiment is shown in figure 2.3. A plane wave is emitted toward an unknown object at angle θ with wave number $k(\boldsymbol{x}) = \omega/c(\boldsymbol{x})$, where ω is the angular frequency of the incident time harmonic wave and $c(\boldsymbol{x})$ is the spatially varying sound speed. The resulting acoustic field $p(\boldsymbol{x})$ satisfies the equation

$$\rho(\boldsymbol{x})\nabla \cdot \left(\rho^{-1}(\boldsymbol{x})\nabla p(\boldsymbol{x})\right) + k^2(\boldsymbol{x})p(\boldsymbol{x}) = -\phi^{\text{inc}}(\boldsymbol{x}) \tag{2.1}$$

where $\rho(\boldsymbol{x})$ is the density and $\phi^{\text{inc}}(\boldsymbol{x})$ is the incident field. If we assume constant density, equation (2.1) reduces to the Helmholtz equation:

$$\Delta p(\boldsymbol{x}) + k^2(\boldsymbol{x})p(\boldsymbol{x}) = -\phi^{\text{inc}}(\boldsymbol{x}). \tag{2.2}$$

Let us now assume that we know a reference medium with wave number $k_0(\boldsymbol{x})$. Its acoustic field is modeled by

$$\Delta p_0(\boldsymbol{x}) + k_0^2(\boldsymbol{x})p_0(\boldsymbol{x}) = -\phi^{\text{inc}}(\boldsymbol{x}). \tag{2.3}$$

Subtracting (2.3) from (2.2) gives

$$\Delta(p(\boldsymbol{x}) - p_0(\boldsymbol{x})) + k_0^2(\boldsymbol{x})(p(\boldsymbol{x}) - p_0(\boldsymbol{x})) = -q(\boldsymbol{x})p(\boldsymbol{x}) \tag{2.4}$$

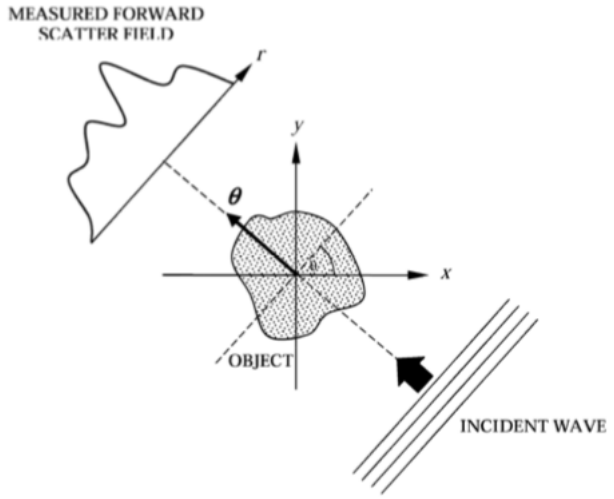


Figure 2.3: Setup for a diffraction experiment, figure reproduced from [2].

where $q(\boldsymbol{x}) = k^2(\boldsymbol{x}) - k_0^2(\boldsymbol{x})$.

Diffraction techniques proceed at this point by replacing $p(\boldsymbol{x})$ in the right hand side of (2.4) by $p_0(\boldsymbol{x})$. That is, we replace the total acoustic field $p(\boldsymbol{x})$ by the initial acoustic field $p_0(\boldsymbol{x})$ This is known as the Born Approximation [35]. With the Born Approximation in place, we can express the solution to (2.2) as

$$p(\boldsymbol{x}) = p_0(\boldsymbol{x}) + \int_\Omega G_0(\boldsymbol{x} - \boldsymbol{y}) q(\boldsymbol{y}) p_0(\boldsymbol{y}) dy, \tag{2.5}$$

where $\Omega \subset \mathbb{R}^2$ is the relevant domain and $G_0(\cdot)$ is the Green's function for the Helmholtz equation with domain Ω and reference medium $k_0(\boldsymbol{x})$. Equation (2.5) is called the Lippmann Schwinger equation [36]. The inverse problem is now stated as: given the initial pressure $p_0(\boldsymbol{x})$ and measurements of the total pressure field $p(\boldsymbol{x})$, determine $q(\boldsymbol{x})$. Note that determining $q(\boldsymbol{x})$ is equivalent to determining the sound speed $c(\boldsymbol{x})$.

With the setup for the inverse problem established, there are a few different methods for reconstructing $q(\boldsymbol{x})$. Two common methods are interpolation in the frequency domain and interpolation in the space domain, which is typically called backpropagation [37]. For acoustic problems with a large amount of scattering, backpropagation is computationally expensive and therefore recon-

structions in the frequency domain are more common. For reconstructing in the frequency domain with the Born approximation in place, a common approach is to use the gridding algorithm [38,39]. Another solution technique with the Born approximation in place is the distorted Born iterative method (DBIM) [40, 41]. In DBIM, we formulate the inverse problem as a minimization problem over the parameter $q(\boldsymbol{x})$ and find a solution iteratively. At each iteration, the reference medium is updated to $k_n(\boldsymbol{x})$ and the Green's function is recomputed at $k_n(\boldsymbol{x})$ as opposed to $k_0(\boldsymbol{x})$. Call this recomputed Green's function G_n. Then the cost functional we want to minimize is

$$F(q) = p_n(\boldsymbol{x}) + G_n \circledast (q_n(\boldsymbol{x})p_n(\boldsymbol{x})),$$

where $p_n(\boldsymbol{x})$ is the pressure at the current iteration, $q_n(\boldsymbol{x}) = k(\boldsymbol{x}) - k_n(\boldsymbol{x})$, and $\circledast$ represents convolution. Now the solution to the inverse problem is given as the solution to

$$\text{argmin}_q \left\{ \|F(q) - p_{\text{meas}}\|_2^2 + \alpha \mathcal{R}(q) \right\}, \tag{2.6}$$

where p_{meas} is the measured pressure on the transducers or other measurement device, $\mathcal{R}(q)$ is a regularization functional and α is a regularization parameter. There are many methods for solving the nonlinear optimization problem in (2.6), and we leave a description of these methods to [42].

Diffraction methods have the benefit of being relatively easy to implement while making fewer simplifying assumptions as compared to TOF methods. However, by replacing the total pressure wave with the incident wave in the Born approximation we are taking a linearizing step which omits information concerning the scattered wave. With that being said, diffraction methods are able to achieve very good reconstructions without being computationally burdensome.

2.3 Algebraic Reconstruction Technique

The next stop in our survey of the current literature in USCT is the algebraic reconstruction technique (ART). Much of the beauty and usefulness of ART comes from its simplicity, as there are few overly complex steps required to implement the algorithm. We begin by superimposing a

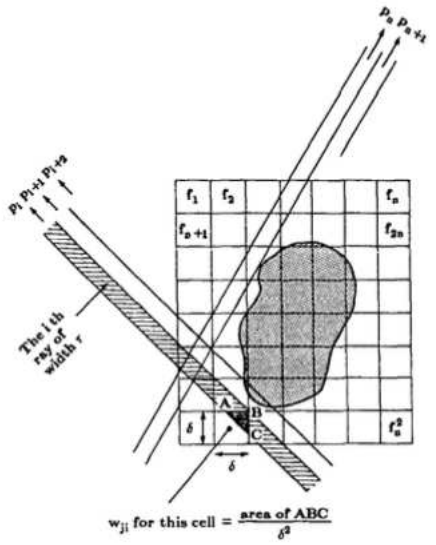


Figure 2.4: Setup for ART, figure reproduced from [3].

square grid over the image $f(x, y)$ that we want to reconstruct, as in figure 2.4. Let f_j denote the value of $f(x, y)$ in cell j and N^2 be the total number of cells. The shaded portion in figure 2.4 represents the emitted sound wave that we are using to probe the object of interest. In ART this sound wave is represented as a ray with width τ rather than a true hyperplane. If we then let p_i represent the pressure measurements corresponding to the propagation of the ith ray, then we can express the relationship between these measurements and the object $f(x, y)$ as

$$\sum_{j=1}^{N^2} w_{ij} f_j = p_i, \quad i = 1, 2, ..., M, \tag{2.7}$$

where M is the total number of rays and w_{ij} is the weighting factor representing the contribution of the jth cell to the ith ray, as represented in figure 2.4.

With the linear system defined in (2.7) we are able to solve for f_j to get the value of $f(x, y)$ in each grid square and thus a reconstruction of the object represented by $f(x, y)$. However, even taking into account the fact that the values w_{ij} are mostly zero, the system in (2.7) is often far too large to do any type of traditional matrix inversion. Instead, most ART reconstructions make use of the Kaczmarz method [43, 44] to more efficiently solve the linear system and produce an

image of the object $f(x, y)$. Since the remainder of this technique deals simply with solving a large linear system effectively, we will leave a thorough discussion of the remainder of the method to the previous sources. There are few steps to implement in ART, but much of that simplicity comes again at the cost of modeling sound waves as rays. As stated in section 2.1, this assumption leads to more information in the form of artifacts than in the object itself. In order to avoid these issues we must reduce the amount of simplifying assumptions in our model, as is done in the next section.

2.4 Full Waveform Inversion

Full Waveform Inversion (FWI) is an imaging technique that has been popular in the geophysics community for decades [21] and is currently making its way into the medical imaging field as well [5, 11, 45–48]. The mathematics of FWI have been well developed by the petroleum industry to image hydrocarbon reserves within the earth [49, 50]. FWI methods are characterized by their use of the full time dependent signal on each transducer and are also unique in that they directly solve the wave equation without any linearization. Thus compared to the TOF and ART techniques, FWI more accurately models the physics of wave propagation. This leads to superior results with regards to resolution and artifact control, at the cost of being quite computationally burdensome. The computational intensity comes from the requirement of solving a fully nonlinear and ill-posed optimization problem. Mathematically, the optimization problem associated with FWI is written concisely as

$$c = \text{argmin}_{\tilde{c}} \left\{ \|F(\tilde{c}) - p_{\text{meas}}(\boldsymbol{x}_k, t)\|_2^2 + \alpha \mathcal{R}(\tilde{c}) \right\}, \tag{2.8}$$

where $F(\tilde{c})$ is the solution to the wave equation at the sound speed map $\tilde{c}$ and at the measurement locations, $p_{\text{meas}}(\boldsymbol{x}_k, t)$ is the measured pressure, $\boldsymbol{x}_k$ are the measurement locations, α is a regularization parameter and $\mathcal{R}(\tilde{c})$ is a regularization functional. We will define the operator $F(\cdot)$, the regularization functional $\mathcal{R}$, and introduce more detailed notation for equation (2.8) in chapter 3. After discretization, solving equation (2.8) becomes an optimization problem over a high dimen-

sional Euclidean space, and those familiar with these types of problems understand the difficulty in finding efficient algorithms for their solution. However, due to improvements in computer hardware and more sophisticated algorithms, FWI is currently a popular method for producing images in USCT. There is a real need for accurate sound speed reconstructions diagnostically in order to, for example, determine whether a particular inhomogeneity is fatty tissue, a benign tumor, or a malignant tumor. Thus the improvement in resolution and accuracy is often significant enough to justify the added computational burden associated with FWI.

The term "full waveform inversion" really involves an array of different reconstructing techniques that share the common theme of full aperture, time dependent data and a lack of linearization in the solution to the forward problem. In [51] and [52], the authors move the problem to the frequency domain and solve the nonlinear Helmholtz equation in place of the wave equation. Reconstructions at lower frequencies are used as starting points in reconstructions at higher frequencies, and close to the full spectrum is used in order to extract as much information as possible from the data. Depending on the number of frequencies used, this can accelerate the computation time as compared to solving the problem in the time domain. The work in [51, 52] is meritable, but we follow [5] by keeping the data firmly in the time domain and solving the nonlinear wave equation instead of the Helmholtz equation.

In the next two chapters we formulate the forward and inverse problems in USCT using a FWI perspective. In chapter 3, we analyze the deterministic approach to FWI and see that adding a regularization term encoding knowledge of the structure of the domain can improve the quality of the results. In chapter 4, we take a different approach and view the problem of USCT through a Bayesian lens while maintaining our devotion to FWI.

Chapter 3

Deterministic Full Waveform Inversion

Our analysis of FWI for USCT begins in the deterministic setting. We first review the mathematical setup for FWI and consider how three different regularization terms affect the final reconstruction. The last of these regularization terms is a novel Structural Similarity regularization function which makes use of an image obtained using Electrical Impedance Tomography (EIT). The description of this technique requires background on both structural similarity and EIT, which we provide. Next, we highlight the solution of the forward problem in USCT using the software package k-Wave and the adjoint method for calculating necessary functional derivatives. Finally, we demonstrate how the algorithm performs on synthetic data generated from both a numerical breast phantom and lung phantom.

3.1 The Minimization Problem

There are many methods for solving an inverse problem in the deterministic setting. Some of these methods are direct, meaning a technique was established which can solve the problem in a non-iterative process. However, when these direct methods are not available or do not perform well, we often use an iterative technique for solving the inverse problem which involves minimizing a given functional. FWI involves finding the sound speed map c which best matches the sound speed that generated pressure data $p_{\text{meas}}(\boldsymbol{x}, t)$. With only this goal in mind, we say the solution to the inverse problem in USCT is the sound speed map c which satisfies

$$\begin{aligned} c &= \text{argmin}_{\tilde{c}} \left\| F(\tilde{c}) - p_{\text{meas,l}}(\boldsymbol{x}_k, t) \right\| \\ &= \text{argmin}_{\tilde{c}} \left\{ \frac{1}{2} \sum_{l=1}^{K} \sum_{k=1}^{K} \left\| F_{l,k}(\tilde{c}) - p_{\text{meas,l}}(\boldsymbol{x}_k, t) \right\|^2_{L^2([0,T])} \right\}, \end{aligned} \tag{3.1}$$

where $F_{l,k} \colon L^2(\Omega) \to L^2([0,T])$ is the operator which maps the sound speed c to the time dependent pressure signal of duration T on the transducer at location $\boldsymbol{x}_k$, when the transducer at location

$\boldsymbol{x}_l$ acts as an acoustic source. $p_{\text{meas,l}}(\boldsymbol{x}_k, t)$ is the time dependent pressure recorded at transducer $\boldsymbol{x}_k$ when transducer $\boldsymbol{x}_l$ acts as an acoustic source. Explicitly, $F_{l,k}(c)$ is the solution to the 2D wave equation over a domain of sound speed c at the transducer $\boldsymbol{x}_k$ with only transducer $\boldsymbol{x}_l$ acting as a source. Note that we are suppressing the dependence of c on time, and assuming that the sound speed profile is static during the imaging process. If the inverse problem were well posed in the sense of Hadamard [53] (that is, if $F_{l,k}(c)$ were a compact operator with a continuous inverse for all l, k), then the inverse problem in (3.1) would have a unique solution which could be found using traditional nonlinear optimization techniques such as steepest descent, conjugate gradient, or Gauss-Newton [42]. However, the problem is ill-posed in that the solution does not depend continuously on the data. Thus if we tried to solve the inverse problem posed in equation (3.1) using real data or even simulated data in the presence of noise, we would get a poor and possibly nonsensical reconstruction. The solution to this problem of ill-posedness is to use a regularization term. We now describe three different regularization functionals/functions and explain their meaning within the context of FWI.

3.1.1 Tikhonov Regularization

The first regularization functional we consider is Tikhonov regularization [54], and is a straightforward but effective way to circumvent the ill-posedness of an inverse problem. The Tikhonov regularization term penalizes sound speed maps c which are too large with respect to the L^2 norm. The Tikhonov functional is typically written

$$\mathcal{R}_{\text{Tikh}} \coloneqq \frac{\alpha}{2} \|c\|^2_{L^2(\Omega)}, \tag{3.2}$$

where α is a regularization parameter which determines how strongly we enforce the regularization term. The expression in (3.2) penalizes sound speed maps c that are large in norm, which may not be optimal in every situation. For medical applications, we may assume the sound speed of our domain will not differ greatly from the sound speed of a homogeneous background, likely water.

Thus a more practical Tikhonov regularization term for us is

$$\mathcal{R}_{\text{Tikh},c_0} \coloneqq \frac{\alpha}{2} \left\| c - c_0 \right\|^2_{L^2(\Omega)}, \tag{3.3}$$

where c_0 is a homogeneous background sound speed that is assumed *a priori*. The Tikhonov regularization term in (3.3) penalizes sound speeds which differ greatly from the background sound speed. This avoids large spikes in sound speed in the reconstruction and typically creates an overall smoother reconstruction as compared to other regularization methods. In practice, Tikhonov regularization often works quite well and is a good starting point for testing an algorithm due to its simplicity. With the functional (3.3) in hand, we write the Tikhonov regularized inverse problem for FWI as

$$c = \text{argmin}_{\tilde{c})} \left\{ \frac{1}{2} \sum_{l=1}^{K} \sum_{k=1}^{K} \left\| F_{l,k}(\tilde{c}) - p_{\text{meas,l}}(\boldsymbol{x}_k, t) \right\|^2_{L^2([0,T])} + \frac{\alpha}{2} \left\| \tilde{c} - c_0 \right\|^2_{L^2(\Omega)} \right\}. \tag{3.4}$$

Tikhonov regularization can be understood as a balance between two requirements: $F_{l,k}(c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)$ should be small in norm, meaning the sound speed map c produces pressure measurements similar to the collected data, and $c - c_0$ should be small with respect to the L^2 norm.

3.1.2 Total Variation Regularization

While Tikhonov regularization often results in smooth reconstructions, total variation (TV) regularization [30, 55, 56] excels at preserving edges and delivering sharpness to an image. This is because TV uses the L^1 norm of the derivative of the sound speed map c, allowing for piecewise regular solutions to the inverse problem and therefore better edge preservation. Mathematically, the TV regularization term is written

$$\mathcal{R}_{\text{TV}} = \alpha \int_{\Omega} |\nabla c| d\boldsymbol{x}, \tag{3.5}$$

where α is again a regularization parameter and ∇c is the gradient of the function c. While the TV regularization term given in (3.5) is the traditional way of writing the functional and allows for straightforward evaluation, the use of the L^1 norm (or l^1 norm after discretization) makes the functional in (3.5) non-differentiable. This is problematic, as many minimization techniques require us to evaluate the derivative of the loss term in (3.1) as well as the regularization terms. To account for this, we alter the TV regularization term to be

$$\begin{aligned}\mathcal{R}_{\text{TV}} &= \alpha \int_{\Omega} \left(\nabla c \cdot \nabla c + \epsilon^2\right)^{1/2} d\boldsymbol{x} \\ &= \alpha \int_{\Omega} \left(|\nabla c|^2 + \epsilon^2\right)^{1/2} d\boldsymbol{x}, \end{aligned} \tag{3.6}$$

where ϵ is a small, positive number. This choice of TV regularization term ensures that the derivative of $\mathcal{R}_{\text{TV}}$ is well defined everywhere and will allow us to use gradient-based descent algorithms to solve the minimization problem.

With the TV regularization functional defined in (3.6), we write the TV regularized inverse problem in USCT as

$$c = \text{argmin}_{\tilde{c}} \left\{ \frac{1}{2} \sum_{l=1}^{K} \sum_{k=1}^{K} \|F_{l,k}(\tilde{c}) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\|^2_{L^2([0,T])} + \alpha \int_{\Omega} \left(|\nabla \tilde{c}|^2 + \epsilon^2\right)^{1/2} d\boldsymbol{x} \right\}. \tag{3.7}$$

TV regularization can be understood as a balance between two requirements: the sum of each $F_{l,k}(c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)$ should be small in norm and ∇c should be small with respect to the L^1 norm. A discussion of the discretization of the total variation regularization term and its derivative can be found in [57].

3.1.3 Structural Similarity Regularization with EIT

The final regularization method we consider is a novel approach to using the structural similarity index (SSIM) as a regularizer. In [22], an SSIM map is used in a regularization term for CT images. The authors compare the structure of images produced using x-ray reconstructions at different energy levels. We shall consider an SSIM structural regularization term which compares

the images produced using USCT with those obtained from a different imaging modality, electrical impedance tomography (EIT). To provide a full description of this approach, we first give an overview of EIT and SSIM. This section concludes with a presentation of the SSIM regularization term we employ in our reconstructions of a lung phantom.

EIT

Electrical impedance tomography (EIT) is a noninvasive imaging technique which produces information concerning the electrical properties of the object being imaged. "Electrical properties" is typically understood to be either the electrical conductivity or permittivity. EIT has been implemented widely in the medical community and continues to be explored for imaging in areas such as fluid flow, geography, and material sciences [58]. Like USCT, EIT involves no ionizing radiation and is thus an appealing alternative to x-ray tomography. EIT also features relatively cheap hardware, so it is an attractive option for a medical situation involving continuous monitoring [59]. In EIT, a voltage is applied to the boundary $\partial\Omega$ of the region Ω of interest and current is measured at the boundary in order to reconstruct the aforementioned electrical properties. We note that while we formulate our model with the assumption of transmitting voltage and measuring current, it is always preferable to transmit current and measure voltage. In that case, the Dirichlet-to-Neumann map mentioned below is replaced by the Neumann-to-Dirichlet map, but the mathematical literature is formulated in terms of the Dirichlet-to-Neumann map.

It is outside the scope of this work to include a complete mathematical description of EIT, but we proceed by giving an overview of the forward and inverse problems in EIT as well as providing references for the inversion technique used in this work. We leave a more thorough investigation of EIT to the following sources: [60, 61].

The electric potential $u(\boldsymbol{x})$ in a domain Ω is modeled by the equation

$$\nabla \cdot \gamma(\boldsymbol{x}, \omega) \nabla u(\boldsymbol{x}) = 0, \tag{3.8}$$

where $\boldsymbol{x} \in \Omega$, $\gamma(\boldsymbol{x}, \omega) = \sigma(\boldsymbol{x}, \omega) + i\epsilon(\boldsymbol{x}, \omega)$ is the admittivity, $\sigma(\boldsymbol{x}, \omega)$ is the conductivity, $\epsilon(\boldsymbol{x}, \omega)$ is the permittivity, and ω is the angular frequency of the applied current. If we apply a known voltage $f(\boldsymbol{x})$ to the boundary of Ω, we have the homogeneous Dirichlet boundary condition

$$u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega. \tag{3.9}$$

We can then measure the resulting current density $j(\boldsymbol{x})$ on the boundary, which corresponds to knowing the Neumann boundary condition

$$\gamma(\boldsymbol{x}, \omega)\frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{n}} = j(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega, \tag{3.10}$$

where $\boldsymbol{n}$ is the outward unit normal vector on $\partial\Omega$. The term $\gamma(\boldsymbol{x}, \omega)\frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{n}}$ maps the voltage on the boundary to the current density on the boundary. For this reason, $\gamma(\boldsymbol{x}, \omega)\frac{\partial u(\boldsymbol{x})}{\partial \boldsymbol{n}}$ is called the Dirichlet-to-Neumann (DN) map, and is denoted Λ_γ. We follow [62] and define the weak form of the DN map $\Lambda_\gamma : H^{\frac{1}{2}}(\partial\Omega) \to H^{-\frac{1}{2}}(\partial\Omega)$ as

$$\langle \Lambda_\gamma f, g \rangle = \int_\Omega \gamma \nabla u \nabla v, \tag{3.11}$$

where $v \in H^1(\Omega)$ has trace g on $\partial\Omega$ and u is the unique solution to the Dirichlet problem defined by equations (3.8), (3.9). Definitions of $H^1(\Omega)$ and $H^{\frac{1}{2}}(\partial\Omega)$ can be found in [63], and $H^{-\frac{1}{2}}(\partial\Omega)$ is the dual space of $H^{\frac{1}{2}}(\partial\Omega)$, as usual. A more in-depth analysis of the DN map can be found in [64].

The data in EIT is the DN map Λ_γ, or more specifically a discrete approximation of Λ_γ. The forward problem in EIT is thus to determine a function $u(\boldsymbol{x})$ satisfying equations (3.8) an (3.9) given knowledge of the admittivity $\gamma(\boldsymbol{x}, \omega)$. The inverse problem in EIT is to determine $\gamma(\boldsymbol{x}, \omega)$ given knowledge of the DN map Λ_γ, given knowledge of j in equation (3.10).

There are many techniques for carrying out the inversion in EIT. An overview of several popular methods can be found in [65]. For this work, a Newton-Raphson method was used to produce an

image of the (real-valued) conductivity $\sigma(\boldsymbol{x}, \omega)$ of a lung phantom. This reconstruction is shown in section 3.6.2. An overview of the Gauss-Newton method and its use in EIT is given in [66].

SSIM

There is not a function which maps $\sigma(\boldsymbol{x}) \rightarrow c(\boldsymbol{x})$ directly, meaning there is no canonical way to obtain the sound speed of a domain Ω given the conductivity over that region. However, a conductivity map $\sigma(\boldsymbol{x})$ provides useful information related to the location, shape, and size of inhomogeneities in the region Ω. We thus need to leverage this information provided in the EIT reconstruction for use in the FWI algorithm.

We will use the information contained in the EIT conductivity reconstruction within an SSIM-structure regularization term. For this section, we assume that our sound speed map c has been discretized on a grid of size $N \times N$. The remainder of the analysis in this section supposes we are working with this discretized sound speed map $\boldsymbol{c}_n$ and the discretized conductivity map $\boldsymbol{\sigma}$. The notation $(\boldsymbol{c}_n)_i$ refers to the ith element of $\boldsymbol{c}_n$, where in this case we take $\boldsymbol{c}_n$ to be the vector in $\mathbb{R}^{N^2}$ obtained by vectorizing the discretized sound speed map over Ω. Note that we are using the notation $\boldsymbol{c}_n$ to suggest a sound speed map at step n of an iterative algorithm.

SSIM is a method for determining the similarity of two or more images by comparing the images in terms of their luminance, contrast, and structure [67]. The luminance of an image is the mean of the total quantity being measured (in our case, the mean of the sound speed). Thus an image with a high sound speed in many pixels will have a high luminance. The contrast of an image is the standard deviation of all of the pixel values. Therefore an image that is nearly uniform in sound speed will have a low contrast as opposed to an image with large changes in the sound speed, which will of course have high contrast. While one can analyze the structure of an image itself, we are more interested in comparing the structure of one image with another. Given the EIT conductivity reconstruction, $\boldsymbol{\sigma}$, and a sound speed map $\boldsymbol{c}_n$, the structure function of the structural

similarity index of the two images is given by

$$s(\boldsymbol{c}_n, \boldsymbol{\sigma}) = \frac{\sigma_{\boldsymbol{c}_n\boldsymbol{\sigma}} + C}{\sigma_{\boldsymbol{c}_n}\sigma_{\boldsymbol{\sigma}} + C}, \tag{3.12}$$

where $\sigma_{\boldsymbol{c}_n\boldsymbol{\sigma}}$ is the cross correlation of $\boldsymbol{c}_n$ and $\boldsymbol{\sigma}$, and $\sigma_{\boldsymbol{c}_n}$, $\sigma_{\boldsymbol{\sigma}}$ are the standard deviations of $\boldsymbol{c}_n$ and $\boldsymbol{\sigma}$, respectively, and C is a small, positive real number. In the traditional application of SSIM, one can produce a map showing the similarity between two images with respect to luminance, contrast, and structure. Our goal is to produce an SSIM regularization term which penalizes sound speeds far from $\boldsymbol{\sigma}$ with regards to only the structure of the two images. For this, we follow [68] and compute the cross correlations and standard deviations locally using an 11×11 window which moves pixel by pixel over the entire image. Let $\mathcal{I}_i$ denote the set of indices for the 11×11 window at pixel i. Further, let $w = \left\{ w_k \mid k \in \mathcal{I}_i \right\}$ denote the local weights obtained from a circular symmetric Gaussian weighting function with standard deviation equal to 1.5. Then we can write the locally computed standard deviation and cross correlation terms at pixel i as

$$(\sigma_{\boldsymbol{c}_n})_i = \left(\sum_{j \in \mathcal{I}_i} w_j ((\boldsymbol{c}_n)_j - (\mu_{\boldsymbol{c}_n})_i)^2 \right)^{1/2} \tag{3.13}$$

$$(\mu_{\boldsymbol{c}_n})_i = \sum_{j \in \mathcal{I}_i} w_j (\boldsymbol{c}_n)_j \tag{3.14}$$

$$(\sigma_{\boldsymbol{c}_n\boldsymbol{\sigma}})_i = \sum_{j \in \mathcal{I}_i} w_j ((\boldsymbol{c}_n)_j - (\mu_{\boldsymbol{c}_n})_i)((\boldsymbol{\sigma})_j - (\mu_{\boldsymbol{\sigma}})_i). \tag{3.15}$$

Once we construct $s(\boldsymbol{c}_n, \boldsymbol{\sigma})$ in this way, we have a two dimensional image where a high value in pixel i indicates that the sound speed map $\boldsymbol{c}_n$ is structurally similar to the EIT reconstruction $\boldsymbol{\sigma}$ at that pixel. A whimsical example of an image and the same image that has been blurred, as well as their structural similarity matrix, is given in figure 3.1.

A similar example demonstrates that it is only the structure of the images which contributes to the forming of the SSIM structure map. In figure 3.2 we see the same image which has now been scaled instead of blurred, leaving the structure between the images the same but the values

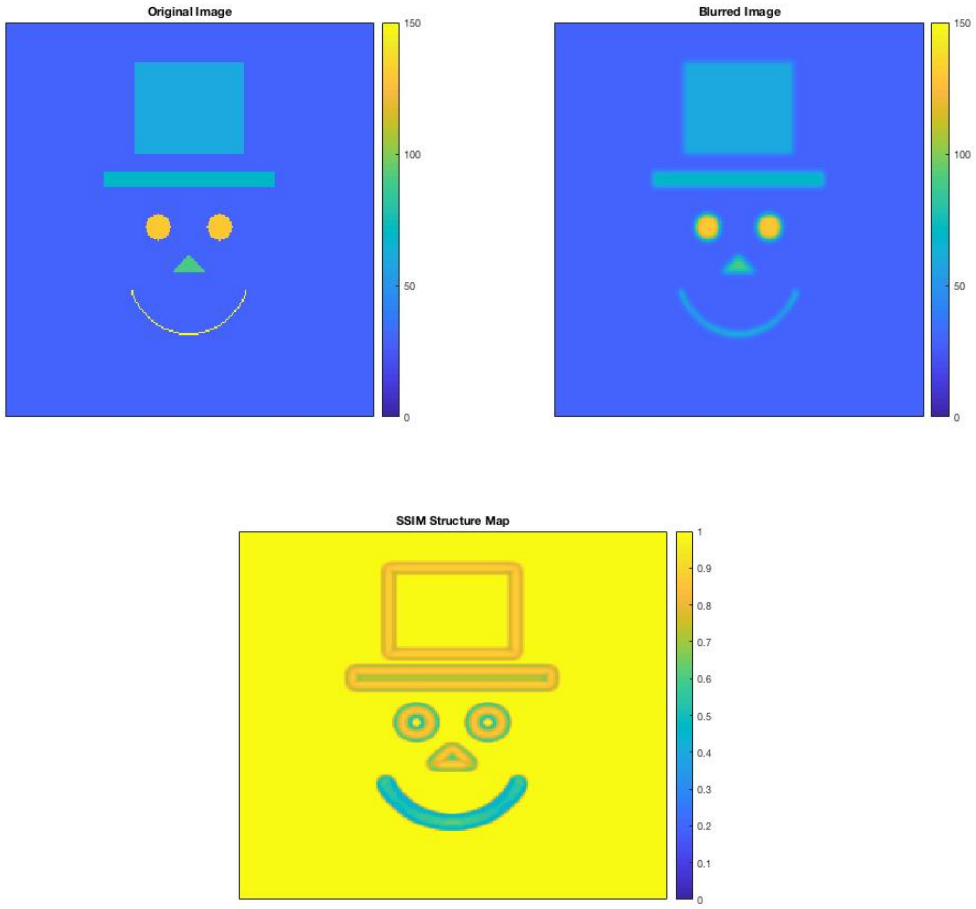


Figure 3.1: The original image in the top left was blurred to give the image in the top right. The SSIM structure map for these two images is shown in the bottom image.

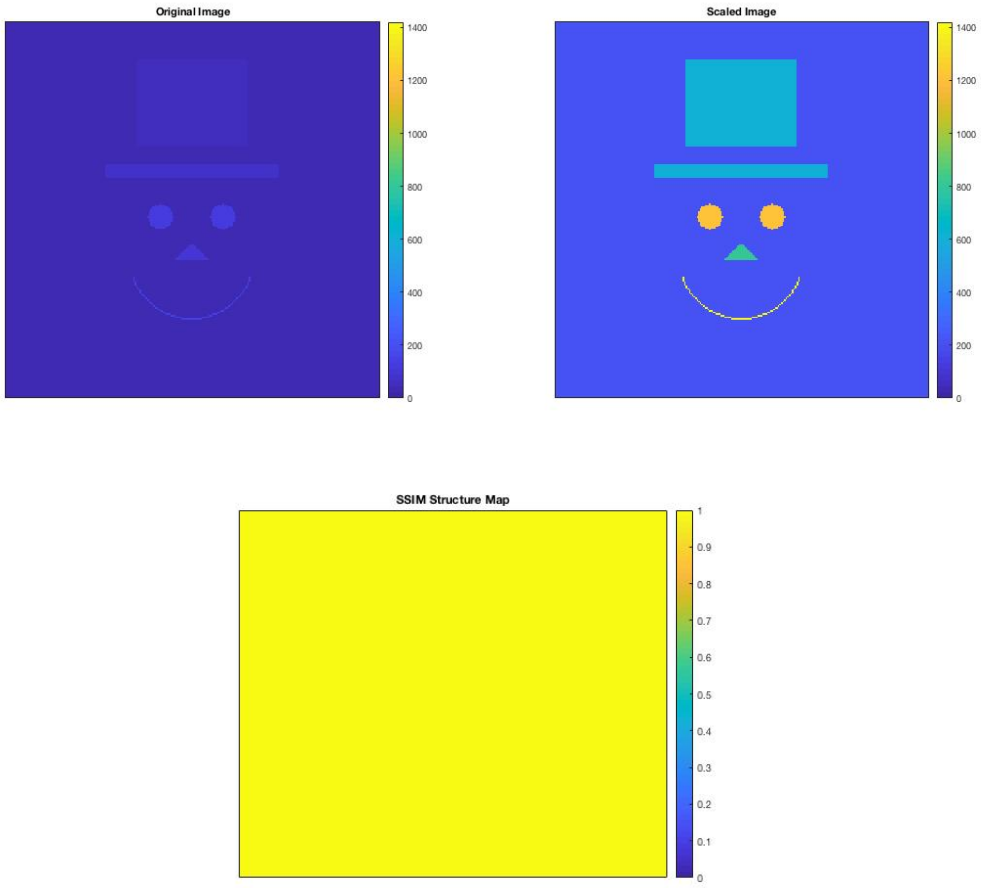


Figure 3.2: The original image in the top left was scaled to give the image in the top right. The SSIM structure map for these two images is shown in the bottom image and is identically equal to 1.

dramatically different. In this case, the SSIM structure map is 1 everywhere, indicating that the size and placement of the inhomogeneities within each image is identical.

With s defined, we must consider how to use it within a regularization term. We need the regularization term to be a real valued function of the discretized sound speed map $\boldsymbol{c}_n$, where large values of the regularizer indicate dissimilarity to the EIT reconstruction $\boldsymbol{\sigma}$. To accomplish this, we will follow [22] and let the SSIM regularization term be defined as

$$S(\boldsymbol{c}_n, \boldsymbol{\sigma}) = \frac{1}{\overline{s}(\boldsymbol{c}_n, \boldsymbol{\sigma})}, \tag{3.16}$$

where

$$\overline{s}(\boldsymbol{c}_n, \boldsymbol{\sigma}) = \frac{1}{N^2} \sum_{i=1}^{N^2} (s(\boldsymbol{c}_n, \boldsymbol{\sigma}))_i. \tag{3.17}$$

The term in (3.17) is a scalar value representing the mean of the map s over all pixels. Therefore, $S(\boldsymbol{c}_n, \boldsymbol{\sigma})$ is a real-valued function of $\boldsymbol{c}_n$ which decreases as $\boldsymbol{c}_n$ becomes more structurally similar to $\boldsymbol{\sigma}$ and achieves a minimum when $\boldsymbol{c}_n$ has the same structure as $\boldsymbol{\sigma}$.

With the SSIM regularization functional defined, we can write the new inverse problem in USCT using an EIT-SSIM structural prior as

$$c = \text{argmin}_{\tilde{c}, \tilde{\boldsymbol{c}}_n} \left\{ \frac{1}{2} \sum_{l=1}^{K} \sum_{k=1}^{K} \| F_{l,k}(\tilde{c}) - p_{\text{meas,l}}(\boldsymbol{x}_k, t) \|^2_{L^2([0,T])} + \alpha S(\tilde{\boldsymbol{c}}_n, \boldsymbol{\sigma}) \right\}. \tag{3.18}$$

The functional in (3.18) is a bit odd due to the use of both the continuous map $\tilde{c}$ and the discretized map $\tilde{\boldsymbol{c}}_n$, but upon implementation $\tilde{c}$ will of course be discretized to become $\tilde{\boldsymbol{c}}_n$. We believe the meaning of the minimization in (3.18) is clear. SSIM regularization can be understood as a balance between two requirements: $F_{l,k}(c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)$ should be small in norm, and $\boldsymbol{c}_n$ should be close in structure to the map $\boldsymbol{\sigma}$.

We now consider the derivative of (3.16) for use in gradient-based optimization methods. The calculation of the derivative of this term is included not only for completeness but because a discussion of this derivative is seemingly absent from the literature. Since we are computing the statistics $\sigma_{\boldsymbol{c}_n}$ and $\sigma_{\boldsymbol{c}_n \boldsymbol{\sigma}}$ separately on each pixel, we are also computing $s(\boldsymbol{c}_n, \boldsymbol{\sigma})$ separately on each pixel. Therefore $(s(\boldsymbol{c}_n, \boldsymbol{\sigma}))_i$ is a mapping from R to R, meaning we can calculate the derivative of $S(\boldsymbol{c}_n, \boldsymbol{\sigma})$ with respect to $\boldsymbol{c}_n$ on each pixel by using finite differences. We call this derivative $S'(\boldsymbol{c}_n, \boldsymbol{\sigma})$, where it is understood that the derivative is taken with respect to $\boldsymbol{c}_n$. The derivative on each pixel is found as

$$(S'(\boldsymbol{c}_n, \boldsymbol{\sigma}))_i = -\frac{1}{\overline{s}(\boldsymbol{c}_n, \boldsymbol{\sigma})^2} \left(\frac{1}{N} \sum_{i=1}^{N} (s(\boldsymbol{c}_n, \boldsymbol{\sigma}))'_i \right), \tag{3.19}$$

with

$$(s(\boldsymbol{c}_n, \boldsymbol{\sigma}))'_i = \frac{((\sigma_{\boldsymbol{c}_n})_i(\sigma_{\boldsymbol{\sigma}})_i) + C)(\sigma_{\boldsymbol{c}_n\boldsymbol{\sigma}})'_i - ((\sigma_{\boldsymbol{c}_n\boldsymbol{\sigma}})_i + C)(\sigma_{\boldsymbol{c}_n})'_i(\sigma_{\boldsymbol{\sigma}})_i}{((\sigma_{\boldsymbol{c}_n})_i(\sigma_{\boldsymbol{\sigma}})_i) + C)^2}, \tag{3.20}$$

where

$$(\sigma_{\boldsymbol{c}_n})'_i \coloneqq \frac{(\sigma_{\boldsymbol{c}_n+\epsilon\boldsymbol{e}_i})_i - (\sigma_{\boldsymbol{c}_n})_i}{\epsilon} \tag{3.21}$$

$$(\sigma_{\boldsymbol{c}_n\boldsymbol{\sigma}})'_i \coloneqq \frac{(\sigma_{(\boldsymbol{c}_n+\epsilon\boldsymbol{e}_i)\boldsymbol{\sigma}})_i - (\sigma_{\boldsymbol{c}_n\boldsymbol{\sigma}})_i}{\epsilon}, \tag{3.22}$$

with $\boldsymbol{e}_i$ being an array with 1 in the ith entry and 0 elsewhere. While the work laid out in equations (3.19), (3.20), (3.21), (3.22) is notationally off-putting, the actual computation of the derivative is straightforward.

Before continuing to a consideration of the forward problem, we point out that any of the regularization terms explained in section 3.1 can be combined with relative ease. For example, if we wanted to promote both spatial regularity of our solution and also enforce an SSIM prior, we could define our minimization problem as

$$c = \text{argmin}_{\tilde{c},\tilde{\boldsymbol{c}_n}} \left\{ \frac{1}{2} \sum_{l=1}^{K} \sum_{k=1}^{K} \|F_{l,k}(\tilde{c}) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\|^2_{L^2([0,T])} + \frac{\alpha_1}{2} \|c - c_0\|^2_{L^2(\Omega)} + \alpha_2 S(\tilde{\boldsymbol{c}}_n, \boldsymbol{\sigma}) \right\}. \tag{3.23}$$

We explore the quality of reconstructions obtained by solving (3.23) in section 3.5.3.

3.2 Computational Method for the Forward Problem

Having presented the inverse problem, we must now consider how to solve the minimization problems laid out in section 3.1. These solutions will entail an iterative method, and in almost all iterative methods a solution to the forward problem is required at least once during each iteration. The forward problem in USCT is stated as follows: given a sound speed map c and the initial pressure distribution on the transducers, determine the resulting time dependent pressures

$p_{\text{meas,l}}(\boldsymbol{x}, t)$ on the transducers. In actuality, we must solve K forward problems where K is the total number of transducers. This is because we use each transducer as an emitter and a receiver in order to gather data. Since we require the solution to at least K forward problems within each step of our iterative algorithm, we are in need of an efficient computational method for simulating wave propagation through a medium. For this purpose, we use the software package k-Wave [69], a MATLAB toolbox for the time domain simulation of acoustic wave fields. This section gives an overview of the k-Wave toolbox.

3.2.1 k-Wave

Typically when one thinks of a solution to the forward problem in the time domain for USCT, one thinks of a solution to the wave equation. However, k-Wave does not solve the wave equation but rather solves a system of coupled first order equations. In the isentropic case, these coupled first order equations are

$$\begin{aligned} \frac{\partial \boldsymbol{u}(\boldsymbol{x}, t)}{\partial t} &= -\frac{1}{\rho_0} \nabla p(\boldsymbol{x}, t) \quad && \text{(momentum conservation)} \\ \frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} &= -\rho_0 \nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t) \quad && \text{(mass conservation)} \\ p(\boldsymbol{x}, t) &= c_0^2 \rho(\boldsymbol{x}, t) \quad && \text{(pressure-density relation),} \end{aligned} \tag{3.24}$$

where $\boldsymbol{u}$ is the acoustic particle velocity, p is the acoustic pressure, ρ is the acoustic density, ρ_0 is equilibrium density, and c_0 is the isentropic sound speed.

The equations in (3.24) are used in the traditional derivation of the wave equation, so solving these equations is completely equivalent to solving the linearized wave equation. k-Wave solves this coupled system of equations instead of the wave equation because it allows for mass and force sources to easily be included and also allows for the simple inclusion of a perfectly matched layer (PML), which absorbs the acoustic waves when they reach the edge of the domain ensuring there is no interference from reflecting waves. The above equations can be made nonlinear and acoustic source terms can be added as well. We will specify these terms when discussing data generation

in section 3.5.1. A more detailed explanation of the linear and nonlinear first order equations governing acoustic waves can be found in [69–72].

We must now determine what numerical method will be used to solve the equations in (3.24). k-Wave solves these equations using the *k*-space pseudospectral (or just *k*-space) method. This technique combines the spectral calculation of spatial derivatives using the Fourier collocation method with a temporal propagator expressed in the spatial frequency domain (or *k*-space). Typically, spatial derivatives are approximated by fitting a polynomial to a number of grid points overlaying the domain and calculating the derivative of that fitted polynomial. The higher the degree of the polynomial, the more accurate the derivative. The Fourier collocation method fits a Fourier series to the grid points, and then takes the derivative of that Fourier series. The main advantage of this is that the amplitudes of the Fourier components can be calculated efficiently using the fast Fourier transform (FFT).

While the spatial derivatives are fairly straightforward to calculate, the temporal derivatives require a bit more care. To illustrate, consider the second order wave equation for homogeneous and lossless media:

$$\nabla^2 p(\boldsymbol{x}, t) - \frac{1}{c_0^2} \frac{\partial^2}{\partial t^2} p(\boldsymbol{x}, t) = 0.$$

If we take a spatial Fourier transform and discretize the time derivative using a second order accurate central difference, we obtain

$$\frac{p(\boldsymbol{k}, t + \Delta t) - 2p(\boldsymbol{k}, t) + p(\boldsymbol{k}, t - \Delta t)}{\Delta t^2} = -(c_0 k)^2 p(\boldsymbol{k}, t). \tag{3.25}$$

In a conventional finite difference discretization of the temporal derivatives, the time step Δt must be kept small to ensure that errors are not introduced into the numerical solution. *k*-space methods seek to relax the necessity for a small time step without sacrificing accuracy. Assume we have an exact solution to the homogeneous and lossless wave equation for initial pressure distribution

$p(\boldsymbol{k}, 0)$ given by

$$p(\boldsymbol{k}, t) = \cos(c_0 k t) p(\boldsymbol{k}, 0),$$

where $k^2 = \|\boldsymbol{k}\|$. By plugging this into the leapfrog finite difference scheme $p(\boldsymbol{k}, t + \Delta t) - 2p(\boldsymbol{k}, t) + p(\boldsymbol{k}, t - \Delta t)$ and rearranging, we obtain

$$\frac{p(\boldsymbol{k}, t + \Delta t) - 2p(\boldsymbol{k}, t) + p(\boldsymbol{k}, t - \Delta t)}{\Delta t^2 \text{sinc}^2(c_0 k \Delta t/2)} = -(c_0 k)^2 p(\boldsymbol{k}, t). \tag{3.26}$$

Comparing (3.25) and (3.26) we see that Δt^2 has been replaced by $\Delta t^2 \text{sinc}^2(c_0 k \Delta t/2)$. For large time steps, the term $\Delta t^2 \text{sinc}^2(c_0 k \Delta t/2)$ provides an exact solution. Thus the basis of the numerical method that k-Wave uses to solve (3.24) is to replace Δt in a first-order accurate forward difference with $\Delta t \text{sinc}(c_0 k \Delta t/2)$. It can be shown that the resulting numerical method is consistent with the continuous equations, and is stable and accurate under reasonable assumptions on $\Delta t \text{sinc}(c_0 k \Delta t/2)$. We leave the verification of these facts as well as a more in-depth derivation of the numerical method to the following sources: [73–77].

We have chosen to use the k-Wave software package for the following reasons. First, the *k*-space method both reduces the memory requirements and the number of time steps required to achieve accurate simulations for the forward problem [69]. This means that k-Wave is a reasonably fast and accurate way to compute solutions to the forward problem. The second and third reasons are the ease of usability in k-Wave and the cost of programmer time. Setting domains that are consistent with a cross-section of the human body is straightforward in k-Wave, and establishing the location and properties of each transducer is simple. This means that setting up the domain of interest and transducer locations over that domain is easy with knowledge of MATLAB and the k-Wave library. This should be contrasted with the difficulty in building a similar domain from scratch. There is great value in programming a forward solver without the use of pre-existing libraries, but the most expensive resource in a project is always programmer time. Instead of allocating this time to building a forward solver that may ultimately be inferior to k-Wave, we have

chosen to let k-Wave handle the forward problem while we focus on developing a useful algorithm for the inverse problem.

3.3 The Adjoint Method for Computing Derivatives

In order to apply a gradient based method for solving the minimization problems in section 3.1, we must be able to compute the derivative of the term

$$E(c) \coloneqq \frac{1}{2}\sum_{l=1}^{K}\sum_{k=1}^{K} \|F_{l,k}(c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\|^2_{L^2([0,T])}\,. \tag{3.27}$$

We stated in section 3.1 that $F_{l,k} : L^2(\Omega) \to L^2([0,T])$ is an operator which maps the sound speed c to the time dependent pressure on the transducer at location $\boldsymbol{x}_k$ when the transducer at location $\boldsymbol{x}_l$ transmits a pulse. For this section, it will be helpful to write $F_{l,k}(c)$ as $p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid c)$, read as "the computed time dependent pressure on transducer $\boldsymbol{x}_k$ when transducer $\boldsymbol{x}_l$ transmits a pulse, given the sound speed map c." With this change, $E(c)$ looks like

$$E(c) = \frac{1}{2}\sum_{l=1}^{K}\sum_{k=1}^{K} \|p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\|^2_{L^2([0,T])}\,. \tag{3.28}$$

We can then write $E(c)$ for a single transmitting receiver $\boldsymbol{x}_l$ as

$$E_l(c) = \frac{1}{2}\sum_{k=1}^{K} \|p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\|^2_{L^2([0,T])} \tag{3.29}$$

$$= \frac{1}{2}\sum_{k=1}^{K} \int_0^T (p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t))^2 dt. \tag{3.30}$$

Summing $E_l(c)$ over all transducers $\boldsymbol{x}_l$ recovers the term $E(c)$.

$E_l(c)$ maps the sound speed c to a number in $\mathbb{R}$, meaning $E_l(c)$ is a functional over $L^2(\Omega)$. We therefore seek the Fréchet derivative of $E_l(c)$, which we denote $\nabla_c E_l(c)$. Once we have $\nabla_c E_l(c)$ for each l, we can sum the result to give the Fréchet derivative of $E(c)$. We compute $\nabla_c E_l(c)$ using a technique known as the adjoint method [5, 78].

The Fréchet derivative $\nabla_c E_l(c)$ is implicitly defined by the relation

$$\delta E_l(c) = \int_\Omega \nabla_c E_l(c) \delta c(\boldsymbol{x}) d\boldsymbol{x}, \tag{3.31}$$

where $\delta E_l(c)$ is the variation in $E_l(c)$ due to a variation δc in the sound speed c. We proceed by perturbing the sound speed c by δc, which will produce a perturbation in the computed pressure $p_{\text{comp,l}}(\boldsymbol{x}, t \mid c)$, which in turn will produce the perturbation $\delta E_l(c)$. That perturbation will then be compared to (3.31) in order to identify the Fréchet derivative $\nabla_c E_l(c)$.

To start, consider the wave equation

$$\Delta p(\boldsymbol{x}, t) - \frac{1}{c^2(\boldsymbol{x})} \frac{\partial^2 p(\boldsymbol{x}, t)}{\partial t^2} = -S_l(\boldsymbol{x}, t), \tag{3.32}$$

where $S_l(\boldsymbol{x}, t)$ is a temporal and spatially dependent source variable. For us, $S_l(\boldsymbol{x}, t)$ will be a time varying signal sent into one of the K transducers. Note that we are using the linear wave equation here to model the wave-field instead of the first order equations in (3.24). This is reasonable, since solving the linear equations in (3.24) is equivalent to solving (3.32). Returning to our derivation, we assume the initial conditions

$$p(\boldsymbol{x}, 0) = 0 \tag{3.33}$$

$$\left. \frac{\partial p(\boldsymbol{x}, t)}{\partial t} \right|_{t=0} = 0. \tag{3.34}$$

We also introduce the term $v(\boldsymbol{x}) = (c_0/c)^2$, where c_0 is again the homogeneous background sound speed, turning (3.32) into

$$\Delta p(\boldsymbol{x}, t) - \frac{v(\boldsymbol{x})}{c_0^2} \frac{\partial^2 p(\boldsymbol{x}, t)}{\partial t^2} = -S_l(\boldsymbol{x}, t). \tag{3.35}$$

This simplifies the following calculations, as it is easier to find an expression for $\nabla_v E_l(v)$ and then obtain $\nabla_c E_l(c)$ through the relationship

$$\nabla_c E_l(c) = -2\frac{c_0^2}{c^3(\boldsymbol{x})}\nabla_v E_l(v). \tag{3.36}$$

To proceed, we need the following definition.

Definition 1. The *adjoint wave equation* is given by

$$\Delta\tilde{p}(\boldsymbol{x},t) - \frac{v(\boldsymbol{x})}{c_0^2}\frac{\partial^2\tilde{p}(\boldsymbol{x},t)}{\partial t^2} = -\tilde{S}_l(\boldsymbol{x},t) \tag{3.37}$$

with terminal conditions

$$\tilde{p}(\boldsymbol{x},T) = 0 \tag{3.38}$$

$$\left.\frac{\partial\tilde{p}(\boldsymbol{x},t)}{\partial t}\right|_{t=T} = 0, \tag{3.39}$$

and source

$$\tilde{S}_l(\boldsymbol{x},t) = \sum_{k=1}^{K}\left(p_{\text{comp,l}}(\boldsymbol{x}_k,t \mid v) - p_{\text{meas,l}}(\boldsymbol{x}_k,t)\right)\delta_{\text{dirac}}(\boldsymbol{x}-\boldsymbol{x}_k), \tag{3.40}$$

where δ_{dirac} is the Dirac-delta function.

The adjoint wave equation is identical to the wave equation in (3.35) except that it has terminal conditions in place of initial conditions and its source term is a superposition of time dependent sources at the receiving transducer locations where each individual signal is the difference between the computed and measured signals on that transducer. The adjoint equation is not physically realizable, but can be solved by time reversing the signals in source term (3.40), solving the wave equation with this source, and then time reversing the recorded signals. Thus the solution to a single adjoint equation is roughly the same computational cost as a single solution to the wave equation.

We are now ready to perturb the parameter v in an attempt to obtain $\delta E_l(v)$. We vary the parameter $v(\boldsymbol{x})$ by $\delta v(\boldsymbol{x}) \in H_0(\Omega)$. The space $H_0(\Omega)$ ($H_0(\Omega) = \{f \in H^2(\Omega) \ : \ f = 0, \ \nabla f = 0 \text{ on } \partial\Omega\}$) is a natural choice since the term $v(\boldsymbol{x}) + \delta v(\boldsymbol{x})$ will then still belong to $L^2(\Omega)$. This variation adds a perturbation $\delta p(\boldsymbol{x}_k, t)$ in the computed time dependent pressures $p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid v)$ and thus a perturbation in $E_l(v)$ given by

$$\begin{aligned} E_l(v + \delta v) &= \frac{1}{2} \sum_{k=1}^{K} \int_0^T (p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid v) + \delta p(\boldsymbol{x}_k, t) - p_{\text{meas,l}}(\boldsymbol{x}_k, t))^2 dt \\ &= E_l(v) + \frac{1}{2} \sum_{k=1}^{K} \int_0^T \left(2p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid v)\delta p(\boldsymbol{x}_k, t) - 2p_{\text{meas,l}}(\boldsymbol{x}_k, t)\delta p(\boldsymbol{x}_k, t)\right) dt \\ \Rightarrow \delta E_l(v) &= \sum_{k=1}^{K} \int_0^T \left(p_{\text{comp,l}}(\boldsymbol{x}_k, t \mid v) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\right) \delta p(\boldsymbol{x}_k, t) dt. \end{aligned} \tag{3.41}$$

Substituting (3.40) followed by (3.37) into (3.41) gives

$$\begin{aligned} \delta E_l(v) &= \int_\Omega \int_0^T \tilde{S}_l(\boldsymbol{x}, t)\delta p(\boldsymbol{x}, t) dt d\boldsymbol{x} \\ &= -\int_\Omega \int_0^T \left(\Delta\tilde{p}(\boldsymbol{x}, t) - \frac{v}{c_0^2}\frac{\partial^2 \tilde{p}(\boldsymbol{x}, t)}{\partial t^2}\right) \delta p(\boldsymbol{x}, t) dt d\boldsymbol{x}. \end{aligned} \tag{3.42}$$

Integrating (3.42) by parts with respect to space and time and using (3.33) as well as the fact that $\delta v(\boldsymbol{x}) \in H_0(\Omega)$ gives

$$\delta E_l(v) = -\int_\Omega \int_0^T \left(\Delta(\delta p(\boldsymbol{x}, t)) - \frac{v}{c_0^2}\frac{\partial^2(\delta p(\boldsymbol{x}, t))}{\partial t^2}\right) \tilde{p}(\boldsymbol{x}, t) dt d\boldsymbol{x}. \tag{3.43}$$

Now if we calculate the variation in the wave equation (3.35) and ignore higher order terms we see

$$\begin{aligned}
&\Delta(p(\boldsymbol{x},t)+\delta p(\boldsymbol{x},t)) - \frac{(v+\delta v)}{c_0^2}\frac{\partial^2(p(\boldsymbol{x},t)+\delta p(\boldsymbol{x},t))}{\partial t^2} = -S_l(\boldsymbol{x},t)\\
\Rightarrow &\Delta p(\boldsymbol{x},t) - \frac{v}{c_0^2}\frac{\partial^2 p(\boldsymbol{x},t)}{\partial t^2} + \Delta(\delta p(\boldsymbol{x},t)) - \frac{v}{c_0^2}\frac{(\partial^2(\delta p(\boldsymbol{x},t))}{\partial t^2} - \frac{\delta v}{c_0^2}\frac{\partial^2 p(\boldsymbol{x},t)}{\partial t^2} = -S_l(\boldsymbol{x},t)\\
\Rightarrow &\Delta(\delta p(\boldsymbol{x},t)) - \frac{v}{c_0^2}\frac{\partial^2(\delta p(\boldsymbol{x},t))}{\partial t^2} - \frac{\delta v}{c_0^2}\frac{\partial^2 p(\boldsymbol{x},t)}{\partial t^2} = 0. \qquad (3.44)
\end{aligned}$$

Substituting (3.44) into (3.43) gives

$$\delta E_l(v) = -\frac{1}{c_0^2}\int_\Omega\int_0^T \frac{\partial^2 p(\boldsymbol{x},t)}{\partial t^2}\tilde{p}(\boldsymbol{x},t)\delta v dt d\boldsymbol{x}.$$

Integrating this by parts one more time gives

$$\delta E_l(v) = \frac{1}{c_0^2}\int_\Omega\int_0^T \frac{\partial p(\boldsymbol{x},t)}{\partial t}\frac{\partial\tilde{p}(\boldsymbol{x},t)}{\partial t}\delta v dt d\boldsymbol{x}. \qquad (3.45)$$

If we compare (3.45) with (3.31), replacing c by v, we find that

$$\nabla_v E_l(v) = \frac{1}{c_0^2}\int_0^T \frac{\partial p(\boldsymbol{x},t)}{\partial t}\frac{\partial\tilde{p}(\boldsymbol{x},t)}{\partial t}dt.$$

At last, using (3.36) gives

$$\begin{aligned}
\nabla_c E_l(c) &= -\frac{2c_0^2}{c^3}\nabla_v E(v)\\
&= -\frac{2}{c^3}\int_0^T \frac{\partial p(\boldsymbol{x},t)}{\partial t}\frac{\partial\tilde{p}(\boldsymbol{x},t)}{\partial t}dt. \qquad (3.46)
\end{aligned}$$

There are a few things to note for (3.46) above. The first is that in order to calculate the functional derivative $\nabla_c E_l(c)$, we need only calculate the forward problem and the adjoint problem on the full computational grid once each. The second thing to note is that the gradient in (3.46) is independent of the number of receiving transducers. This means we only need to solve the forward

and adjoint problem once regardless of the number of transducers used to measure pressures on the boundary.

With each transducer acting as a source in succession, we obtain the derivative $\nabla_c E(c)$ by summing $\nabla_c E_l(c)$ over all $l = 1, ..., K$. This gives

$$\begin{aligned}\nabla_c E(c) &= \sum_{l=1}^{K} \nabla_c E_l(c) \\ &= -\frac{2}{c^2} \sum_{l=1}^{K} \int_0^T \frac{\partial p(\boldsymbol{x}, t \mid \boldsymbol{x}_l)}{\partial t} \frac{\partial \tilde{p}(\boldsymbol{x}, t \mid \boldsymbol{x}_l)}{\partial t} dt, \end{aligned} \tag{3.47}$$

where $p(\boldsymbol{x}, t \mid \boldsymbol{x}_l)$ and $\tilde{p}(\boldsymbol{x}, t \mid \boldsymbol{x}_l)$ represent the solution to the forward and adjoint problem on the full grid when the transducer at location $\boldsymbol{x}_l$ acts as a source.

For a single computation of the gradient $\nabla_c E(c)$, we must solve K forward problems and K adjoint problems. For large values of K this can be computationally expensive, but it is still far less computationally burdensome than calculating the derivative using a finite difference approach.

3.4 Steepest Descent Algorithm

With the problem outlined and the computational tools explained, we can now focus on solving the minimization problems presented in section 3.1 to obtain reconstructions. There are numerous methods for solving nonlinear optimization problems [79], but we use the straightforward steepest descent (or gradient descent) algorithm. Given an initial sound speed estimate $c_{\text{init}}(\boldsymbol{x})$ (with the discretized image $\boldsymbol{c}_0$) we update the sound speed at each iteration by taking a step in the direction of the negative gradient of the functionals in either (3.4), (3.7), (3.18), (3.23), depending on which regularization term is chosen. Since those functionals are all of the form $E(c) + \alpha\mathcal{R}(c)$ for the chosen regularization term $\mathcal{R}$ and regularization parameter α, these derivatives can be expressed as $\nabla_c E(c) + \alpha\nabla_c \mathcal{R}(c)$, where $\nabla_c \mathcal{R}(c)$ is the derivative of the regularization term with respect to the sound speed. We summarize this relatively simple algorithm below.

Algorithm 1. Given the current sound speed c_n, regularization term $\mathcal{R}$, and regularization parameter α, repeat the following steps:

1. Compute $\nabla_c E(c_n)$ using (3.47) and $\nabla_c \mathcal{R}(c_n)$.

2. Calculate a step length h_n.

3. Update the sound speed as $c_{n+1} = c_n - h_n(\nabla_c E(c_n) + \alpha \nabla_c \mathcal{R}(c_n))$.

If desired, a step can be added to the above algorithm which terminates the algorithm if the error term $E(c_{n+1}) + \alpha \mathcal{R}(c_{n+1})$ falls below a certain amount. We do not take that approach, and instead simply have the algorithm terminate after a fixed number of steps. For a reasonable choice of the starting sound speed c_{init} and parameter h_n, Algorithm 1 should converge to a minimizer c. A discussion of how to select the starting sound speed and parameters is discussed in the next section.

3.4.1 Parameter selection

We consider three parameters in Algorithm 1, the starting sound speed c_{init}, the step length parameter h_n, and the regularization parameter α. Often in FWI, the starting sound speed c_{init} is obtained using a fast, linearized method. This not only speeds convergence but also helps the algorithm avoid unwanted convergence to a local minimum [47]. We follow this convention and use a traditional time of flight algorithm to obtain the starting sound speed. This sound speed c_{init} will be used in both the noiseless case and the noisy case, and we also run the algorithm in the noiseless case with a less informed starting sound speed that will be detailed in section 3.5.

For the selection of h_n, we initially tried to implement a stabilized Barzilai-Borwein (BB) method [80, 81]. In the traditional BB method, we consider an update equation of the form

$$c_{n+1} = c_n - h_n \nabla F(c_n) \tag{3.48}$$

where h_n is the step length and $\nabla F(c_n)$ is the search direction. The BB method proceeds by considering (3.48) to be a quasi-Newton problem where h_n is replaced by D_n, which is an approx-

imation of the inverse Hessian matrix of the objective function. Then, the step-size h_n is computed by assuming that D_n is a rank one matrix, enforcing that D_n has the quasi-Newton condition, and solving the minimization problem

$$\min_{D_n} \left\| D_n^{-1} s_n - y_n \right\|, \tag{3.49}$$

where $s_n = c_n - c_{n-1}$ and $y_n = \nabla F(c_n) - \nabla F(c_{n-1})$. The solution to (3.49) is given by

$$h_n^{BB} = \frac{s_n^T s_n}{s_n^T y_n}$$

where s_n^T denotes the transpose of s_n. It has been shown [80] that the BB method outperforms the traditional backtracking line search algorithm in the quadratic case. However, the BB method often generates step lengths which are too long and may not converge even for certain convex functions [82]. This is concerning for us, since the expense of calculating the derivative demands that we waste as little computation as possible. A work-around to this heavy computational demand is given by a stabilization step outlined in [81]. It is possible that with proper tuning the stabilized BB method could provide decent results, but after implementation we determined that it resulted in step lengths which were far too conservative. Since we would like to reach convergence as soon as possible, this method is unfortunately impractical.

The alternative we found that yielded the best results is a more ad hoc method for determining h_n. If we trust that our choice of c_{init} is close (in some sense) to the true sound speed, we know that our step length should not result in a dramatically different new sound speed at each iteration. Further, we typically know a priori an upper and lower bound for the sound speeds in our domain. Thus our approach will be to choose h_n such that the largest change in the sound speed over the domain is a fixed, positive constant $s_{\max}$, with units of m/s. Mathematically, we can express this choice of step-length as

$$h_n = \frac{s_{\max}}{\max(\text{abs}(\nabla_c E(c_n) + \alpha \nabla_c \mathcal{R}(c_n)))}. \tag{3.50}$$

The value of h_n is dependent on the derivative $\nabla_c E(c_n) + \alpha \nabla_c \mathcal{R}(c_n)$, but since this derivative is necessarily computed at every step of the algorithm prior to choosing the step-length, we do not add any computational expense in the selection of h_n.

In addition, if the resulting update to the sound speed contains a region that is larger or smaller than our prior knowledge of the maximum and minimum sound speeds in the domain, then we shrink the step length by a factor of 10. While this is a without question a heuristic method, it is physically realistic and yields good results both qualitatively and quantitatively.

Finally, there is the matter of choosing the regularization parameter α. This is, in general, a delicate matter. For Tikhonov regularization the heuristic L-curve method is popular [83], and TV regularization has its own techniques for parameter selection [84]. In place of these more rigorous methods, we used a "kitchen sink" approach in which we tested many different regularization parameters and selected the best reconstruction based on qualitative and quantitative results. This is clearly not a foolproof method for selecting a regularization parameter, but yielded good reconstructions for our phantoms. Future work should include a method for more optimally choosing the regularization parameter α.

3.5 Breast Phantom

3.5.1 Data Generation

In order to test the inversion approach presented in Algorithm 1, we must have a viable data set. In lieu of real data, we generated data using the k-Wave software library. Here we lay out the setup and method used for generating data. k-Wave requires four inputs from the user in order to simulate the propagation of acoustic waves in a medium:

- kgrid: an object holding all information concerning the computational grid,
- medium: an array containing all information concerning the physical properties of the domain,

- source: an array containing the position and properties of pressure sources, and
- sensor: an array holding information about the location and properties of sensors which record the pressure.

We will briefly outline how we set the four inputs above in order to generate synthetic data.

kgrid: kgrid sets the computational domain, so there are a few physical quantities we must consider in order to create a reasonable grid. Our domain in this section is a cross section of a human breast, so an overall domain size of 0.1×0.1m is sensible. With the physical size of the computational grid in place, the quantities Nx and Ny (the number of grid points in the x and y directions) and the grid point spacing dx and dy (in meters) can be set by considering the Nyquist limit of two grid points per wavelength [85]. However, since we are considering a simulation with a heterogeneous media, we follow the guidance in [69] and set the grid to allow four points per wavelength. The final bits of information necessary to set the grid are the smallest sound speed in the domain c_{min} and the maximum temporal frequency transmitted f_{max}. We set the smallest sound speed to be 1300 meters/second (m/s) and the maximum temporal frequency to 1 MHz. Then the values Nx and dx are set as

$$dx = \frac{c_{\text{min}}}{(\text{points per wavelength}) * f_{\text{max}}}$$
$$Nx = \text{round}\left(\frac{0.1m}{dx}\right),$$

and for simplicity we set $Ny = Nx$ and $dy = dx$. Given the sound speed and maximum frequency listed above, this would result in a grid of size 240×240 where $dx = dy = 4.1667 \times 10^{-4}$. However, due to the complexity of the breast phantom (which is shown in figure 3.3) we instead set the grid to be of size 300×300 with $dx = dy = 3.333 \times 10^{-4}$. This grid is likely finer than necessary, but was not an exorbitant strain computationally as compared to a grid of size 240×240. The time grid was set by considering the maximum time it would take for a sound wave of the chosen frequency to propagate from one side of the domain to other, while keeping the Nyquist rate of two points

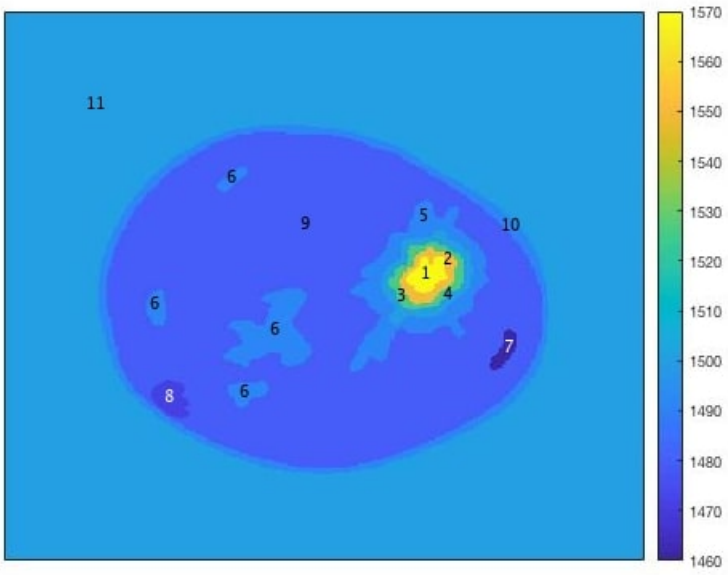


Figure 3.3: Breast phantom used to generate data, phantom adapted from [4].

per wavelength in mind. This led to a time grid spacing of $dt = 6.25 \cdot 10^8$ s with $Nt = 1099$ time steps.

medium: In order to ensure that we are generating data correctly, setting the physical properties of our domain is essential. The "medium" array in k-Wave holds all of the important physical characteristics used to model a desired domain. There are five properties that we must define: the sound speed, the mass density, the nonlinear parameter BonA, the power law absorption coefficient and the power law absorption exponent. The sound speed parameter is a matrix of size $Nx \times Ny$, and in our case will model the cross section of a human breast. The annotated sound speed phantom used to generate our data is shown in figure 3.3 (this phantom is adapted from one given in [4]). A table describing the tissue type and sound speeds of the annotated phantom is given in table 3.1. The sound speed values in table 3.1 match those presented in [86]. The ability to recover the shape, location, and sound speed of all of these inhomogeneities would clearly hold great value in a diagnostic setting.

Table 3.1: Parameters of the numerical breast phantom.

Structure Index	**Tissue Type**	**Sound Speed (m/s)**
1	Malignant Tumor	1570
2	Malignant Tumor	1550
3	Malignant Tumor	1526
4	Malignant Tumor	1500
5	Benign Lesion	1492
6	Benign Lesion/Parenchyma	1492
7	Fatty Tissue	1460
8	Fatty Tissue	1472
9	Parenchyma	1480
10	Adipose	1492
11	Background	1500

The mass density is also an $Nx \times Ny$ matrix. For simplicity, the medium density was set to a constant value of $1000\ g/m^3$. In equations (3.24), we presented the continuous equations that k-Wave solves in order to compute the forward problem. However, when the acoustic medium is heterogeneous with a spatially varying sound speed and ambient density, the equations in (3.24) must include additional terms [69]. If we account for acoustic absorptions, heterogeneities in our medium, and high enough energy acoustic waves to make our system nonlinear [71] (all three of these are present in our model), the system of first order partial differential equations becomes [72, 73, 79, 87]

$$
\begin{aligned}
\frac{\partial u(\boldsymbol{x},t)}{\partial t} &= -\frac{1}{\rho_0}\nabla p(\boldsymbol{x},t) \\
\frac{\partial \rho(\boldsymbol{x},t)}{\partial t} &= -(2\rho(\boldsymbol{x},t)+\rho_0)\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t) - \boldsymbol{u}(\boldsymbol{x},t)\cdot\nabla\rho_0 \\
p(\boldsymbol{x},t) &= c_0^2\left(\rho(\boldsymbol{x},t) + \boldsymbol{d}\cdot\nabla\rho_0 + \frac{B}{2A}\frac{\rho(\boldsymbol{x},t)^2}{\rho_0} - \boldsymbol{L}\rho(\boldsymbol{x},t)\right),
\end{aligned}
\tag{3.51}
$$

where $\boldsymbol{d}$ is the acoustic particle displacement, B/A is a nonlinear parameter which characterizes the relative contribution of finite-amplitude effects to the sound speed, and $\boldsymbol{L}$ is a linear integro-differential operator that accounts for acoustic absorption and dispersion [88, 89] given by

$$
\boldsymbol{L} = \tau\frac{\partial}{\partial t}(-\nabla^2)^{\frac{y}{2}-1} + \eta(-\nabla^2)^{\frac{y+1}{2}-1}
$$

with

$$
\begin{aligned}
\tau &= -2\alpha_0 c_0^{y-1}, \\
\eta &= 2\alpha_0 c_0^{y}\tan(\pi y/2).
\end{aligned}
\tag{3.52}
$$

Here, τ and η are coefficients representing absorption and dispersion, respectively. The constant α_0 is the power law prefactor with units $Np(rad/s)^{-y}m^{-1}$ and y is the power law exponent.

With this laid out, we see that the parameter BonA is the term B/A in (3.51), the power law absorption coefficient is α_0 in (3.52) and the power law absorption exponent is y in (3.52). To generate data, we set these parameters to the values

$$
\begin{aligned}
\text{BonA} &= 0.4 \\
\alpha_0 &= 0.0022 \\
y &= 1.1.
\end{aligned}
$$

We choose this value for BonA since this is the value associated with the sound speed of air [71]

and has shown to yield good results across multiple reconstructions.

source: The source input determines the location and properties of each transducer acting as a source. Supposing we use K transducers to model our system, then we would like the K transducers to symmetrically surround the breast phantom in order to ensure that we are probing the region from all angles. We then treat each of these transducers as a source in K different experiments and record resulting pressures on each of the K transducers. To accomplish this, we set the location of the K transducers using a binary source mask held in source.p_mask. An entry of 1 in source.p_mask indicates that there is a transducer which will emit a time varying pressure wave at that location. The properties of the time varying pressure wave are set using source.p.

To generate data we used $K = 128$ transducers, where each transducer has an area equal to one grid square. We note that while 128 transducers may seem like a large number, since we are modeling the transducers as being quite small, using this number is quite reasonable [20, 48]. An alternative approach would be to use fewer transducers but have the transducers be larger. This approach is valid, but for the purposes of testing our algorithm, using 128 transducers with area equal to one grid square is sufficient. The locations of the transducers are stored in K different binary matrices where each matrix has a single nonzero entry corresponding to the location of the k^{th} transducer. These K binary matrices are then stored in an $Nx \times Ny \times K$ array and are used to activate the correct transducer during each of the K forward problems. After choosing the number of transducers, the next property to consider is the location of the transducers. This is obviously problem dependent, but the most common approach [90] is to use a ring array of transducers. That is the tactic we took, as we let the transducers be spaced equidistantly around a circle of radius $0.042\ m$. This ensures that the ring array of transducers lies completely within the domain and also places the transducers at a slight distance from the boundary of the breast. This matches the physical setup for imaging a breast as depicted in figure 1.1b. A map of the transducer locations and their relation to the breast phantom is shown in figure 3.4. Note that in figure 3.4 the transducers are being modeled as points, but k-Wave allows for transducers to be modeled as larger elements as well.

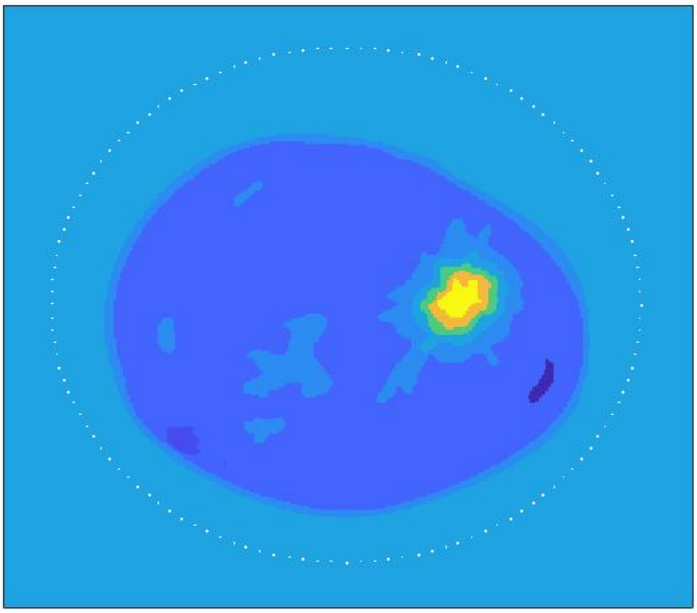


Figure 3.4: Transducer locations for K=128 surrounding the breast phantom.

The time-dependent pressure wave transmitted on a single transducer is set by consecutively using the k-Wave functions toneBurst and filterTimeSeries. The function toneBurst takes in a sampling frequency, source frequency, number of sinusoidal cycles and an optional signal length parameter which controls the number of samples. The sampling frequency is set to one over the time step for the simulation (which was set automatically with the call to kWaveGrid), the source frequency is 1 MHz, the number of sinusoidal cycles is 1 and we set the optional signal length parameter to the largest listening time over the size of the time step. The optional signal length parameter allows us to pad the signal with zeros after the initial pulse so that the emitting transducer can also act as a receiver. In addition to this padding, we have also added a delay to the beginning of the signal as this led to better reconstructions in all cases. There is some care in choosing the strength of the signal, as choosing a stronger signal strength obviously increases the strength of the recorded signals at each transducer. This leads to larger differences in the simulated and recorded pressures and will thus lead to larger values in the derivative in equation (3.47). Using a stronger signal will also decrease the degradation of the signal due to noise, which is an undeniably

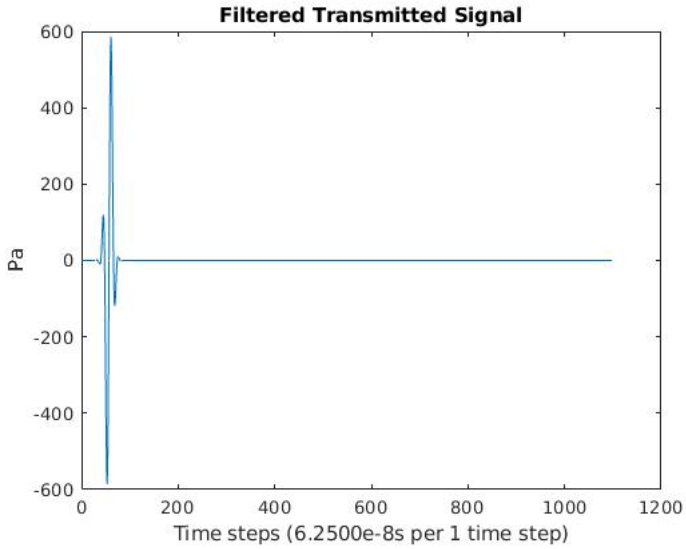
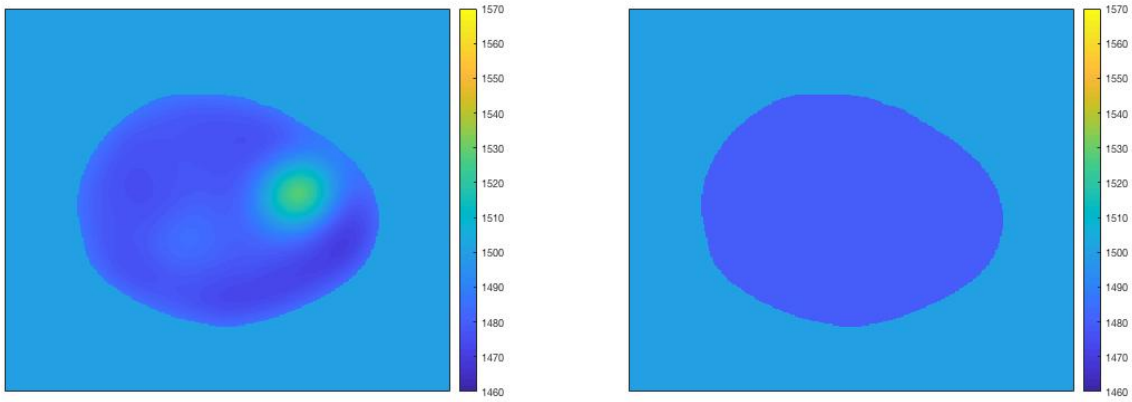


Figure 3.5: Filtered signal transmitted through a single transducer during data simulation.

attractive property. For this reason, we use a signal strength of 1000 Pa. After creating this signal, it is passed through the filterTimeSeries method. This ensures that the signal is restricted to the range of frequencies supported by the grid and also dampens the strength of the signal. The signal inputted at a single transducer during the simulation of data is shown in figure 3.5.

sensor: Considering the work already done to set the location of the transducers as sources, and also keeping in mind that we want to use each transducer as both an emitter and a receiver, setting the sensor locations for each simulation is straightforward. The binary matrix sensor.mask holds the location of each transducer (same as in figure 3.4), and we then have the option of telling the sensors what type of data we want to collect. We have each sensor collect the time dependent pressure at its location, so that we have a total of $K * K$ time dependent signals upon completion of the data collection process (K transducers gathering data after each of the K transducers emits a signal).

After following the above steps, we have a data set of size $K \times K \times Nt$. This is a large data set, but can be stored without issue and gives us a good starting point for the inverse problem.



(a) c_{init} obtained from a TOF algorithm.

(b) c_{init} obtained from only knowledge of the boundary of the breast.

Figure 3.6: Initial sound speeds c_{init} for the breast phantom.

3.5.2 Inverse Problem Setup

The inverse problem of recovering the sound speed map from simulated data was solved in the case of both noiseless and noisy data. For the noiseless case, the same grid was used to generate the data and solve the inverse problem. This is clearly an inverse crime, but is useful in demonstrating the capabilities of the algorithm. In the noiseless case, two different starting sound speeds c_{init} were analyzed. First, a sound speed map c_{init} was obtained from a TOF algorithm as presented in section 2.1. That sound speed is shown in figure 3.6a.

Then, a sound speed map c_{init} was chosen such that the contour and interior of the breast is known exactly with regards to both its shape and sound speed. That is, the background sound speed and interior of the breast are both given their exact sound speeds and no other information concerning the inhomogeneities is assumed. This is not an unreasonable assumption to make, and allows us to test the performance of Algorithm 1 when little information is provided a priori. That starting sound speed is shown in figure 3.6b.

The noisy inverse problems were carried out on a grid of size $Nx = Ny = 300$ while the data for all but one of the problems was generated on a grid of size $Nx = Ny = 302$, in order to avoid an inverse crime. For the final reconstruction, data was generated on a grid of size $Nx = Ny = 400$

in order to test the algorithms robustness against large experimental error. Gaussian noise $\eta \in \mathbb{R}^{Nt}$ was added to each time dependent pressure measurement, where η was drawn from the distribution $\mathcal{N}(\mathbf{0}, \sigma \mathbb{I}_{Nt})$ with σ equal to $0.001 * \max(p_{\text{meas,1}})$.

After adding noise to the simulated data, a calibration step was performed to account for the use of different grids in the simulation of data and solution of the inverse problem. This calibration step begins by running the forward problem on the domain of size 302×302 with a homogeneous background and then running the forward problem with the exact same source on the domain of size 300×300, again with a homogeneous background. Then, the time dependent signal on each transducer for both of these simulations are compared, and in particular we determine the difference between the maximum value of the signal on each grid. This gives a scalar value which represents the shift each time dependent signal undergoes as it moves to the different grid. We then shift the noisy data by this amount so that it better aligns with the grid on which we solve the inverse problem. This technique proved to significantly enhance the reconstruction results. We note that these steps could easily transfer over to the case of true data collection, since data could be collected without the breast present. It is perhaps worth pointing out that this technique would break down if the noise level were high enough to distort the maximum value of the signal, but if this were the case it is unlikely a viable reconstruction could be obtained with or without calibration.

Two different regularization terms were tested for the breast phantom: the Tikhonov regularization term given in equation (3.4) and the total variation term given in equation (3.7). The noiseless reconstruction requires no regularization parameter, but the regularization parameter in each of the noisy reconstructions was chosen by analyzing the results over a variety of choices of α and comparing the results. This led to an understanding of what range of regularization parameters were suitable for each regularization functional. In this work, we present the results from using $\alpha = 10^6, 10^7, 2 \cdot 10^7, 4 \cdot 10^7, 5 \cdot 10^7, 10^8$ for (3.4), and $\alpha = 5 \cdot 10^6, 10^7, 5 \cdot 10^7, 10^8, 10^9, 10^{10}$ for (3.7).

The step lengths h_n were chosen using the method outlined in section 3.4.1. The parameter $s_{\max}$ was set to 1 for the noiseless case and 3 for the noisy reconstructions. 64 iterations were run for the noiseless reconstructions while 40 iterations were run for each of the noisy reconstructions. Fewer iterations were used for the noisy instances both because a larger value of $s_{\max}$ was used in these reconstructions and because the noiseless tests suggested that fewer iterations were sufficient. The results of the inverse problems using Algorithm 1 with these setups are displayed in the next section.

3.5.3 Results

Noiseless

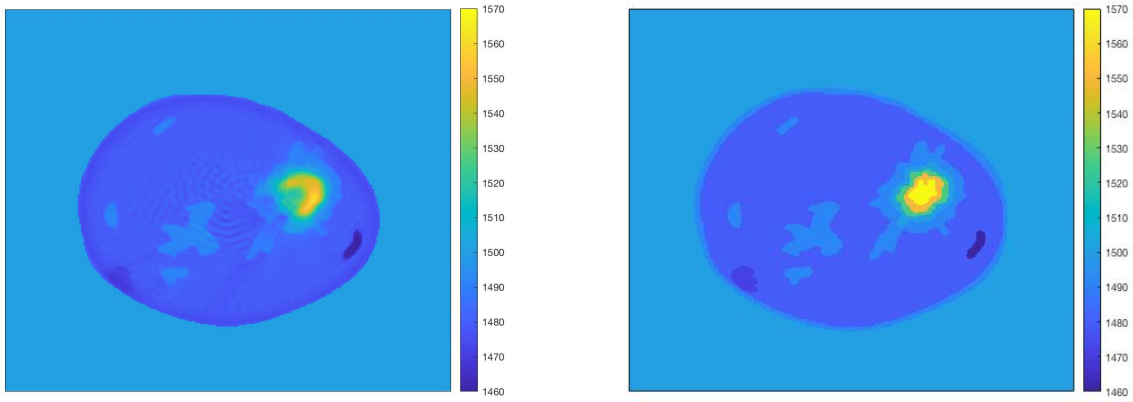
The sound speed after 49 iterations, a reminder of the phantom (ground truth), and a plot of the error functional

$$E(c) = \frac{1}{2}\sum_{l=1}^{K}\sum_{k=1}^{K} \|F_{l,k}(c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\|^2_{L^2([0,T])}$$

for the noiseless reconstruction using c_{init} obtained from a TOF algorithm are shown in figure 3.7.

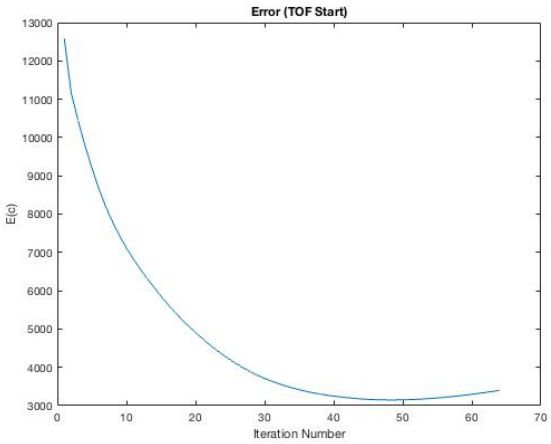
These results are excellent, and suggest that in an ideal case with a starting sound speed that is reasonably close to the true sound speed, Algorithm 1 is capable of almost exactly reconstructing the sound speed profile. In particular, the position, shape, and sound speeds of all inhomogeneities in figure 3.7a almost precisely match the true sound speed profile. This is confirmed by analyzing figure 3.8a, which displays the magnitude of the difference on each pixel between the resulting sound speed from algorithm 1 (called $c_{\text{reconstructed}}$ in the figure) and the true sound speed (called c_{true} in the figure).

We will clearly not typically know the true sound speed, but for the purposes of testing the capabilities of the algorithm this figure is helpful. Note that the largest difference in the sound speed from the reconstruction as compared to the true sound speed is at the location of the large tumor. This is somewhat expected, as that region contains large sound speeds which overlap in complexly shaped layers. Figure 3.8b displays the norm $||c_{\text{true}} - c_{\text{reconstructed}}||_2$ at each iteration of



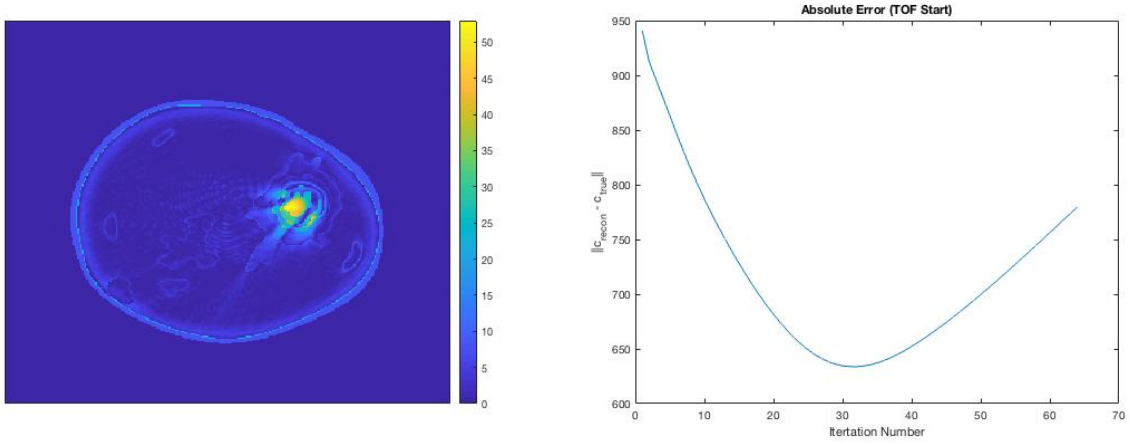
(a) Sound speed map after 49 iterations.

(b) Sound speed phantom.



(c) Error term $E(c)$.

Figure 3.7: Reconstructed sound speed, sound speed phantom, and error $E(c)$ from noiseless data after 49 iterations with c_{init} obtained using a TOF algorithm.



(a) $|c_{\text{true}} - c_{\text{reconstructed}}|$.

(b) Absolute error $||c_{\text{true}} - c_{\text{reconstructed}}||$ at each iteration.

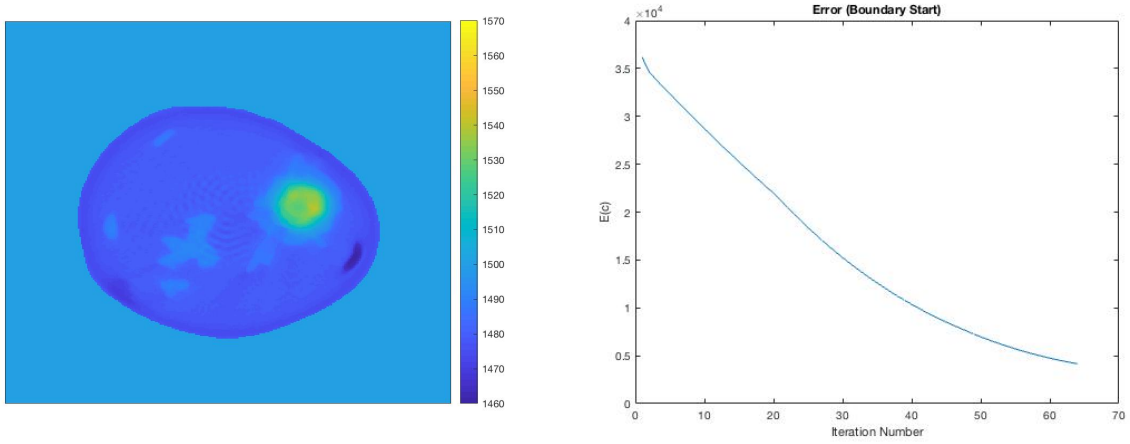
Figure 3.8: The difference in magnitude and the absolute error between the true and reconstructed sound speeds where c_{init} is obtained using a TOF algorithm.

the algorithm. Interestingly, the minimum in the graph occurs at iteration 31 instead of at iteration 49, which is where figure 3.7c suggests the best reconstruction is located.

In addition to the results of the algorithm using a sound speed map c_{init} given from a TOF reconstruction, we also show the results of the algorithm in the noiseless case using a sound speed map c_{init} that assumes we only know the sound speed of the boundary of the breast. The result after 64 iterations is shown in figure 3.9a while figure 3.9b displays the error term $E(c)$ as a function of iteration number.

The results here are also surprisingly good, as the position and shape of the inhomogeneities are picked out well. However, the algorithm was not able to match the true sound speeds of the domain as well as in figure 3.7a, as seen from figure 3.10a.

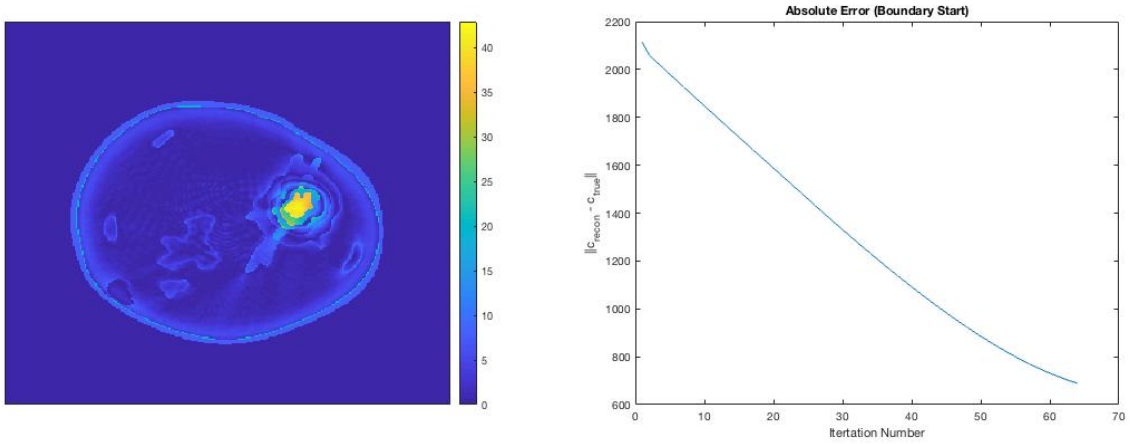
This leads to the somewhat obvious conclusion that using a starting point c_{init} which is closer to the true sound speed will result in faster convergence. The fact that Algorithm 1 was able to still give a quality reconstruction in this case, however, suggests that a precise beginning sound speed c_{init} is not strictly necessary to produce an image of high fidelity.



(a) Sound speed map after 64 iterations.

(b) Error term $E(c)$.

Figure 3.9: Reconstructed sound speed and error from noiseless data after 64 iterations where c_{init} is given as a piecewise function with the correct background sound speed and correct sound speed in the interior of the breast.



(a) $|c_{\text{true}} - c_{\text{reconstructed}}|$.

(b) Absolute error $||c_{\text{true}} - c_{\text{reconstructed}}||$ at each iteration.

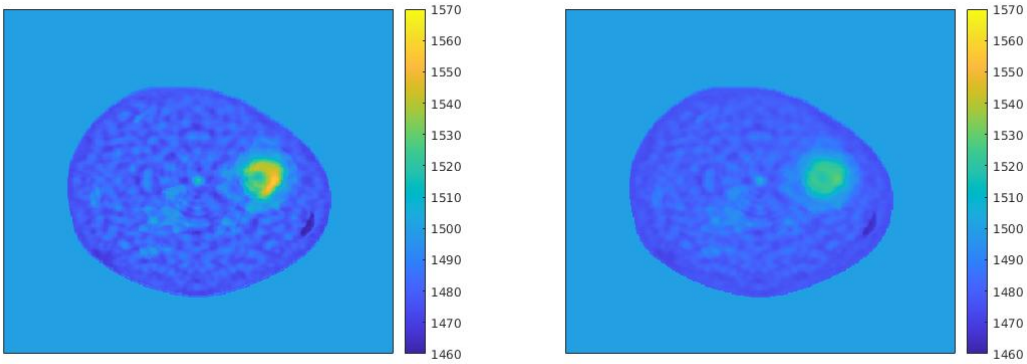
Figure 3.10: The difference in magnitude and the absolute error between the true and reconstructed sound speeds where c_{init} is given as a piecewise function with the correct background sound speed and correct sound speed in the interior of the breast.

Tikhonov (Noisy Data)

Figure 3.11 shows the final iteration of seven different reconstructions using Tikhonov regularization with noisy data in the absence of an inverse crime. Each of these reconstructions used the initial sound speed obtained from a TOF reconstruction shown in figure 3.7. The regularization parameters were selected by first coarsely sweeping over values from $1 \cdot 10^{-10}$ up to $1 \cdot 10^{10}$, running short reconstructions to test for performance, and then refining our selection of α based on that initial sweep. Qualitatively, these reconstructions suggest that the algorithm was able to identify the shape and location of each of the inhomogeneities quite well. There is speckling present in all of the reconstructions, with the most prominent speckling occurring at the lowest regularization parameter $\alpha = 10^6$. As the regularization parameter is increased, the speckling is reduced at the cost of a reduction in the sound speed of the malignant tumor. This trade-off is unavoidable, and finding a balance between the two factors is important in the reconstruction. We also note that there is an unfortunate artifact near the center of the reconstructed breast which is not present in the original phantom. This artifact appears to be in the direct center of the transducers, and could be due to a focusing phenomenon because of the choice of a circular array of transducers.

Figures 3.12 and 3.13 show the values of the error terms $E(c)$ and $\mathcal{R}(c)$ over each iteration for each of the six regularization parameters. Note that these values appear large due to the strong signal amplitude of 1000 Pa used in the solution to the inverse problem. As this amplitude shrinks, so do the values of $E(c)$ and $R(c)$. As expected, increasing the regularization parameter leads to an increase in the value of the loss term $E(c)$ while the regularization term $\mathcal{R}(c)$ begins to decrease. At $\alpha = 10^7$, however, we see a steady decrease in both the loss term and the regularization term. The reconstructed sound speed in figure 3.11b shows the best balance between speckle reduction and identifiable features of the phantom, even if it underestimates the sound speed value of the malignant tumor.

Figure 3.14 displays the difference in norm between the reconstructed sound speed and the true sound speed (ground truth) at each iteration. Clearly, we cannot find this error value in an experimental setting. However, since we know the ground truth in this case calculating this difference



(a) $\alpha = 10^6$ **(b)** $\alpha = 10^7$

(c) $\alpha = 2 \cdot 10^7$ **(d)** $\alpha = 4 \cdot 10^7$

(e) $\alpha = 5 \cdot 10^7$ **(f)** $\alpha = 10^8$

Figure 3.11: Reconstructions from noisy data after 40 iterations for six different Tikhonov regularization parameters α.

is illuminating as a measure of accuracy for our reconstructions. These plots are similar in shape to those in 3.12, but we see that the smallest difference between the reconstructions and the true sound speed occur for he lowest regularization parameter $\alpha = 10^6$.

Finally, figure 3.15 shows the sound speed and all error plots for the reconstruction using data generated on a grid of size 400×400. Due to the difficulty of setting up the breast phantom on that fine of grid, the inhomogeneity at location 8 with reference to table 3.1 is missing from the phantom. Unfortunately, we were only able to test a single regularization parameter ($\alpha = 10^6$) for this setup at this time. However, the sound speed in figure 3.15a displays all of the inhomogeneities present in earlier reconstructions, suggesting that the large experimental error did not significantly degrade the image quality. Interestingly, the error plot in figure 3.15b does not reach the same minimum as those in figure 3.12, but the absolute error plot in figure 3.15d achieves lower values than those in figure 3.14. Since there were changes to the phantom for this simulation, there is likely little to make of this. The takeaway from figure 3.15a should be that Algorithm 1 is robust against larger experimental errors.

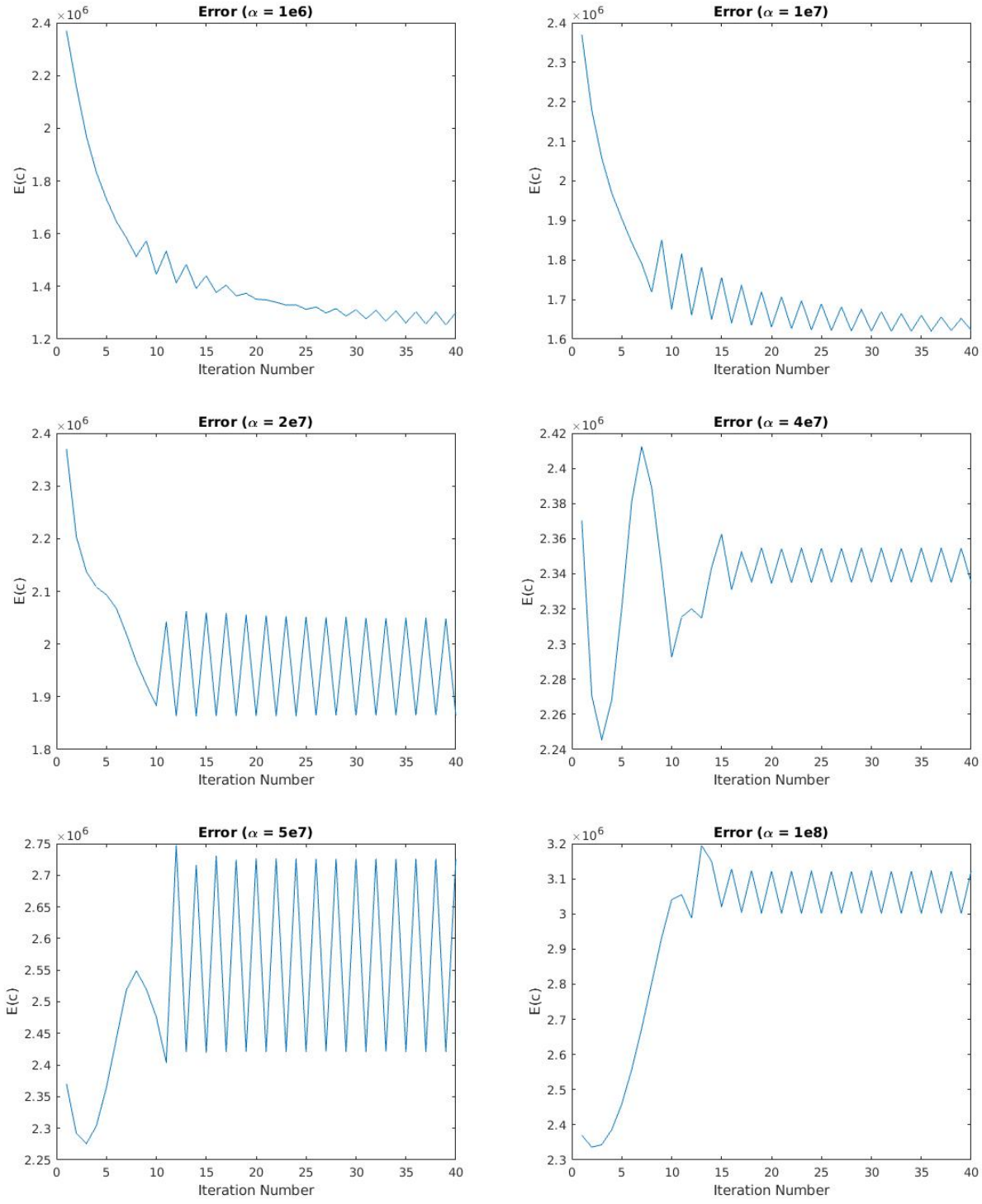


Figure 3.12: Values of the error term $E(c)$ for six different Tikhonov regularization parameters α.

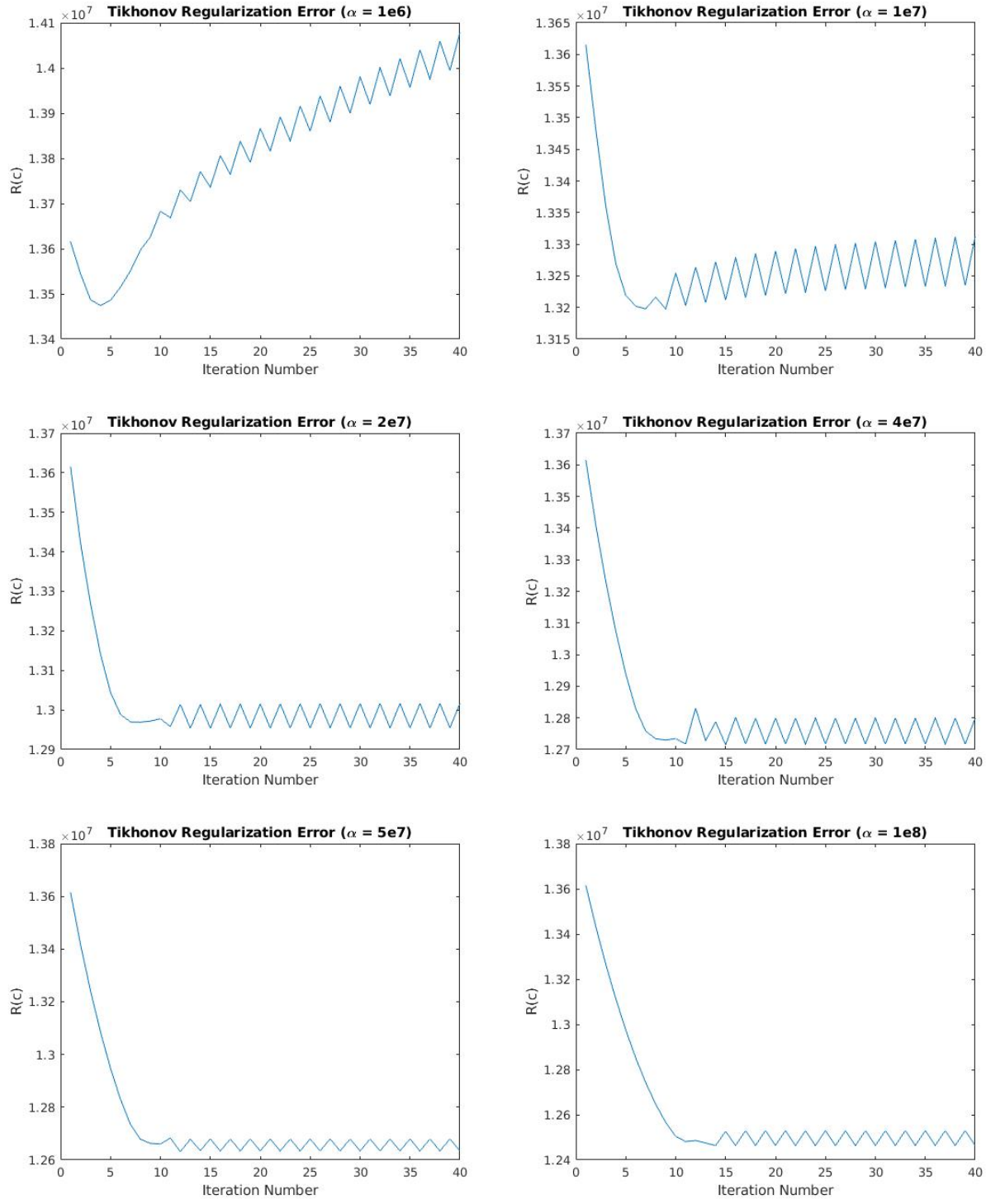


Figure 3.13: Values of the Tikhonov error term $\mathcal{R}(c)$ for six different regularization parameters α.

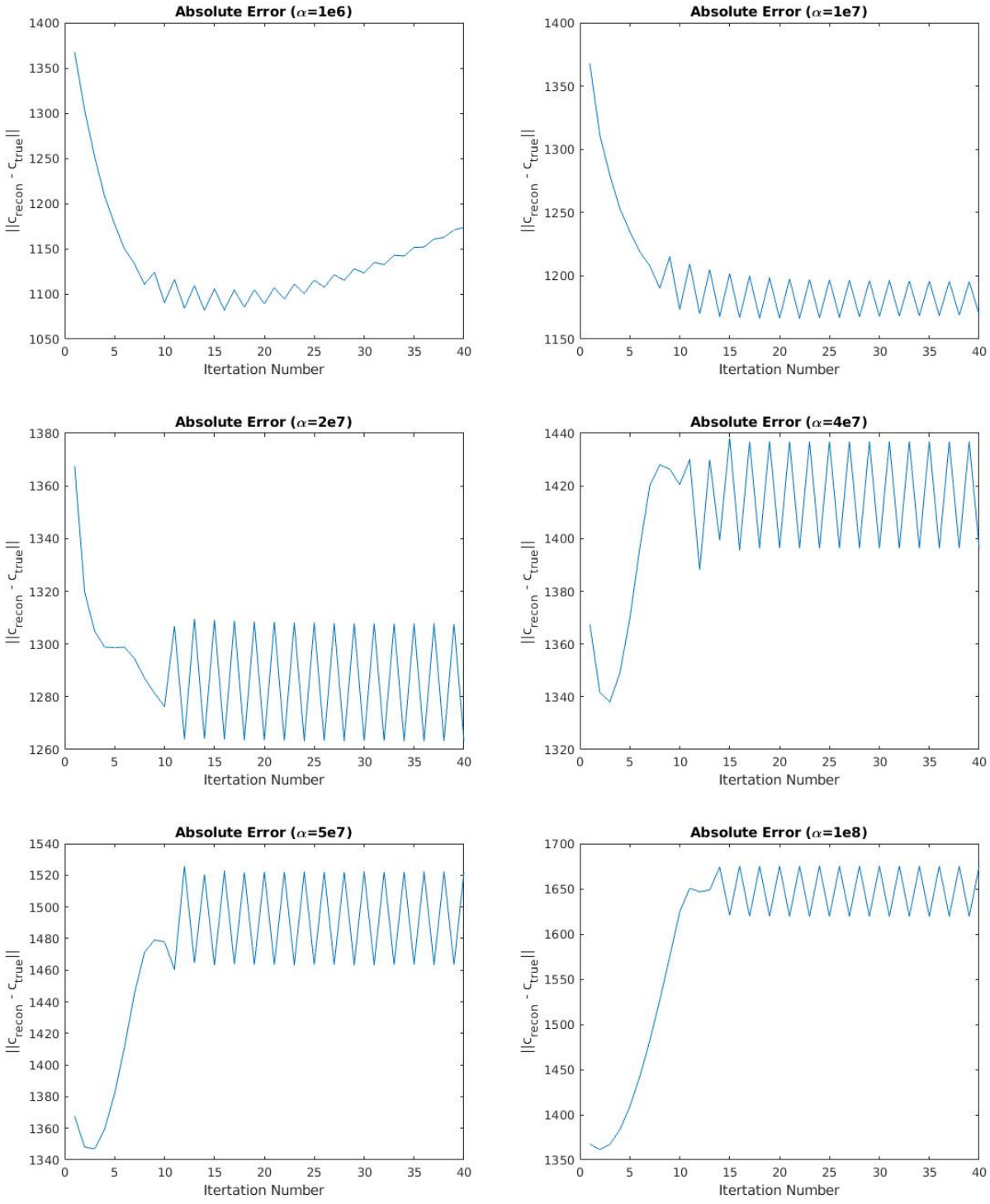


Figure 3.14: The error from ground truth $\|c_{\text{recon}} - c_{\text{true}}\|$ for six different Tikhonov regularization parameters α.

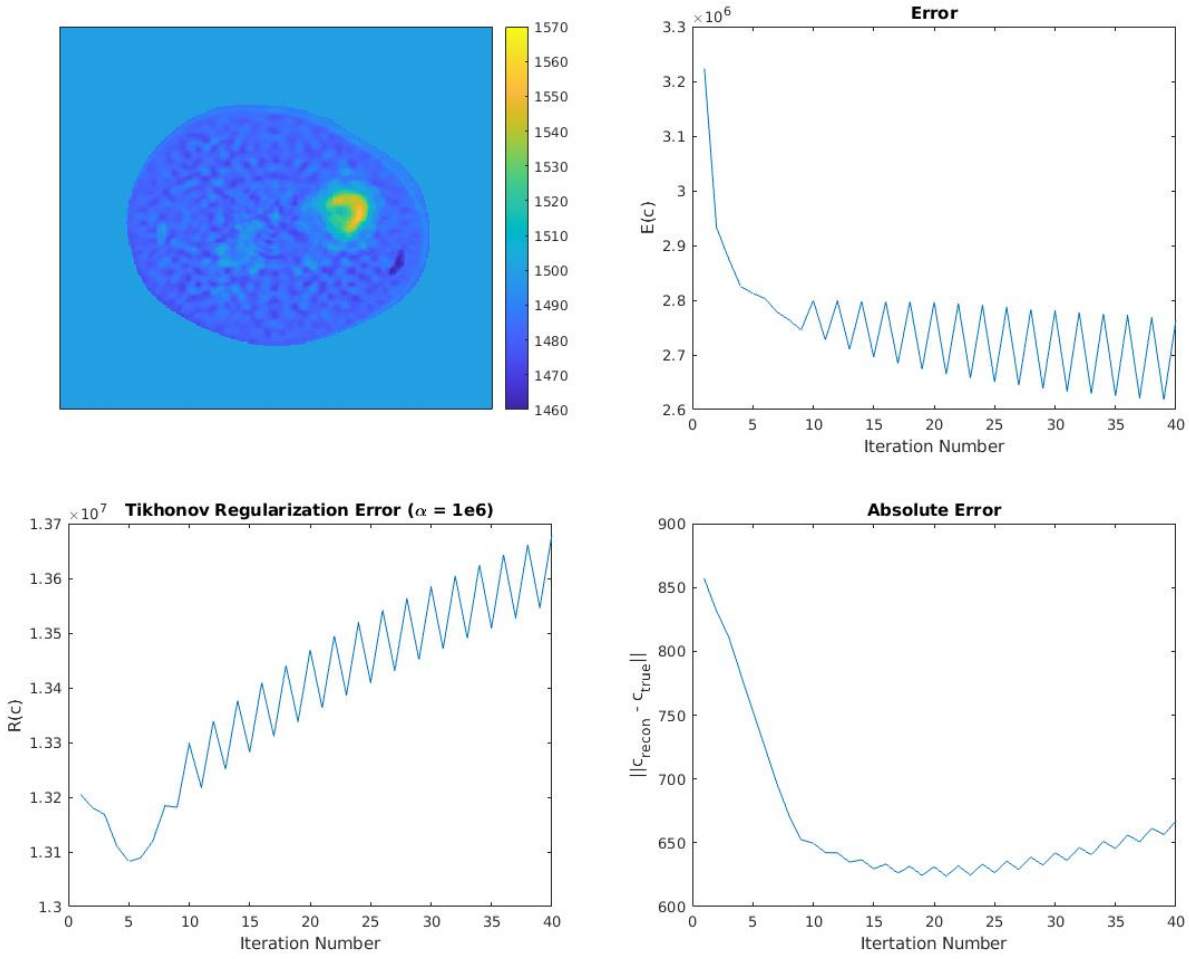


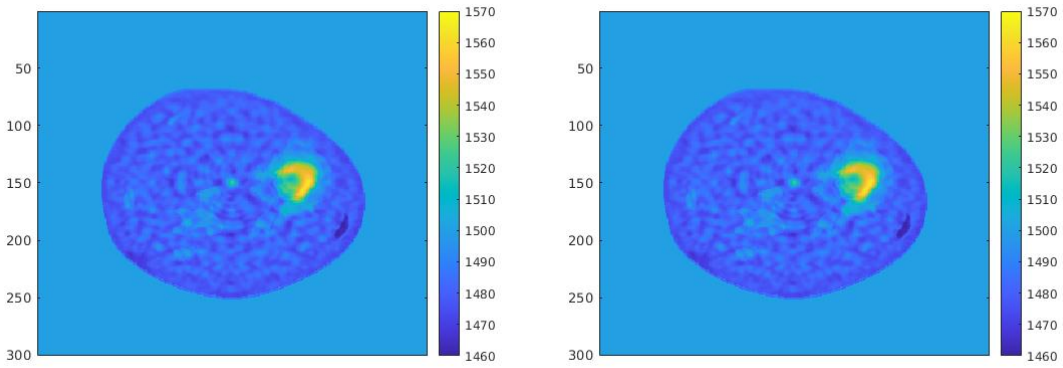
Figure 3.15: The reconstruction, error plot $E(c)$, Tikhonov error plot $R(c)$, and absolute error plot for $\alpha = 10^6$ after 40 iterations for the breast phantom using noisy data generated on a grid of size 400×400.

Total Variation (Noisy Data)

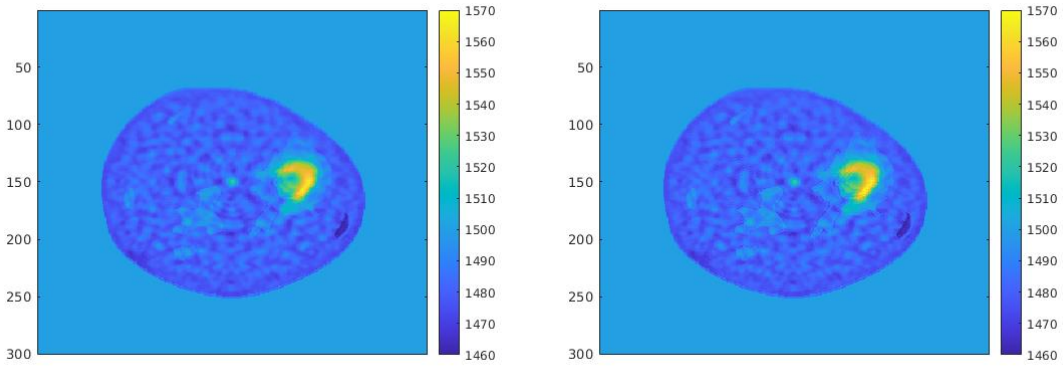
The sound speed reconstructions for six different TV regularization parameters α are shown in figure 3.16. The similarity in these results for $\alpha = 5 \cdot 10^6, 10^7, 5 \cdot 10^7$ and 10^8 suggest that a large TV regularization parameter is needed for this setup. Plots of the loss function $E(c)$ and TV regularization term $R(c)$ are displayed in figures 3.17 and 3.18.

Several of the sound speed maps in figure 3.16 display the location and shape of the inhomogeneities within the phantom well. In particular, the reconstruction for $\alpha = 10^9$ places the inhomogeneities perfectly and also does a decent job at reducing the speckle of the reconstruction. By looking closely, one can also notice that the TV regularization term does a slightly better job at edge preservation as compared to the Tikhonov reconstructions. This can be seen in the layers of decreasing sound speed surrounding the malignant tumor and in the reconstruction of the parenchyma surrounding the inside of the breast.

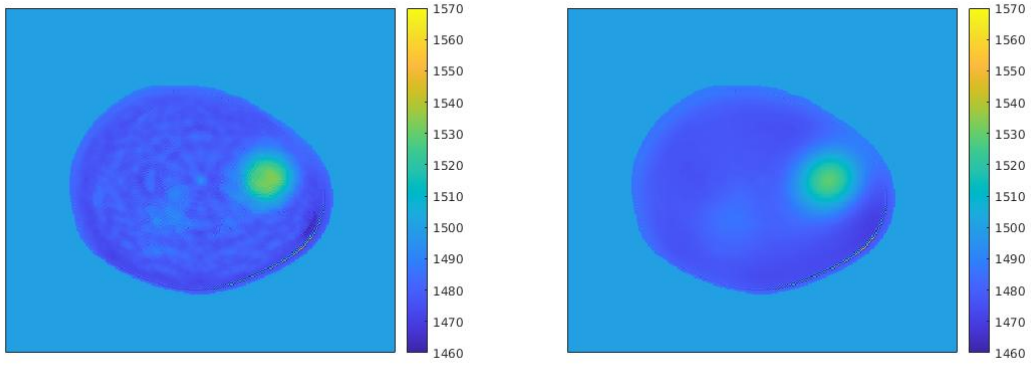
Figure 3.19 displays the difference in norm between the reconstructed sound speed and the ground truth at each iteration. The plots for $\alpha = 5 \cdot 10^6, 10^7$, and $5 \cdot 10^7$ are almost identical and match closely the absolute error plot in figure 3.14a, suggesting that all of these reconstructions are under regularized. However, these plots also display the smallest error between the reconstructions and the ground truth. This could imply that the regularization terms are primarily useful for controlling speckling in our reconstructions, which is still a useful property.



(a) $\alpha = 5 \cdot 10^6$ **(b)** $\alpha = 10^7$



(c) $\alpha = 5 \cdot 10^7$ **(d)** $\alpha = 10^8$



(e) $\alpha = 10^9$ **(f)** $\alpha = 10^{10}$

Figure 3.16: Reconstructions from noisy data after 40 iterations for six different TV regularization parameters α.

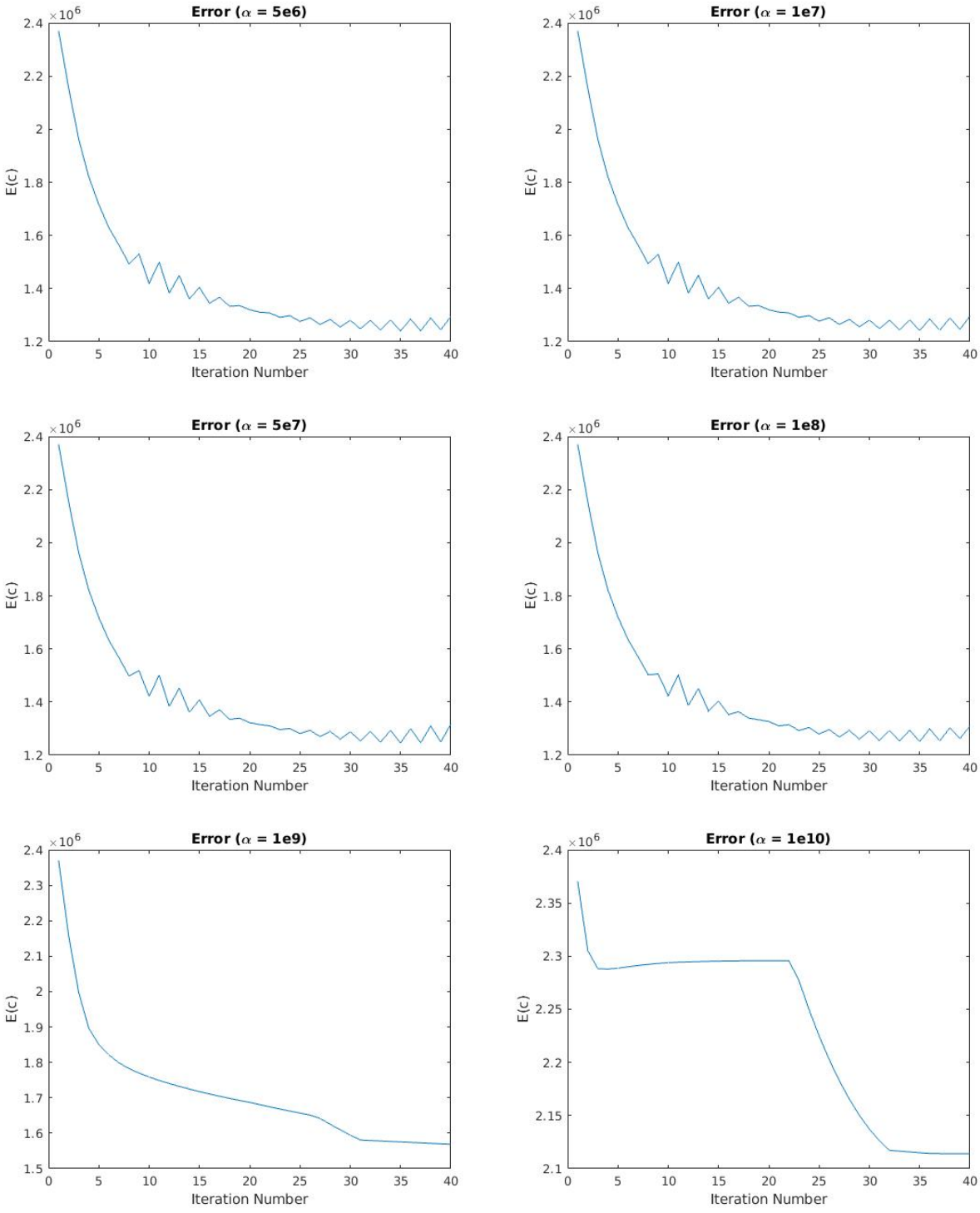


Figure 3.17: Values of the error term $E(c)$ for six different TV regularization parameters α.

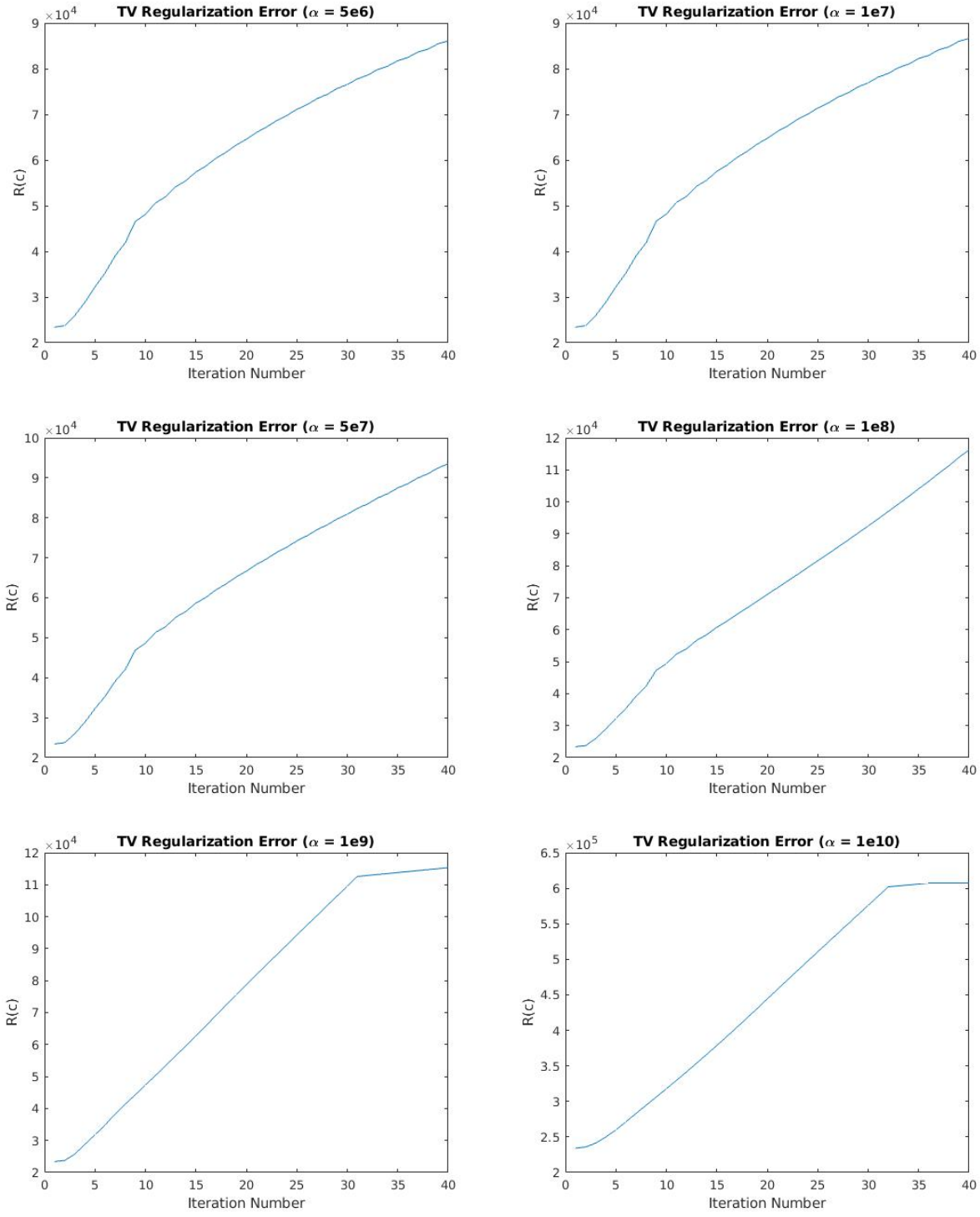


Figure 3.18: Values of the TV error term $R(c)$ for six different TV regularization parameters α.

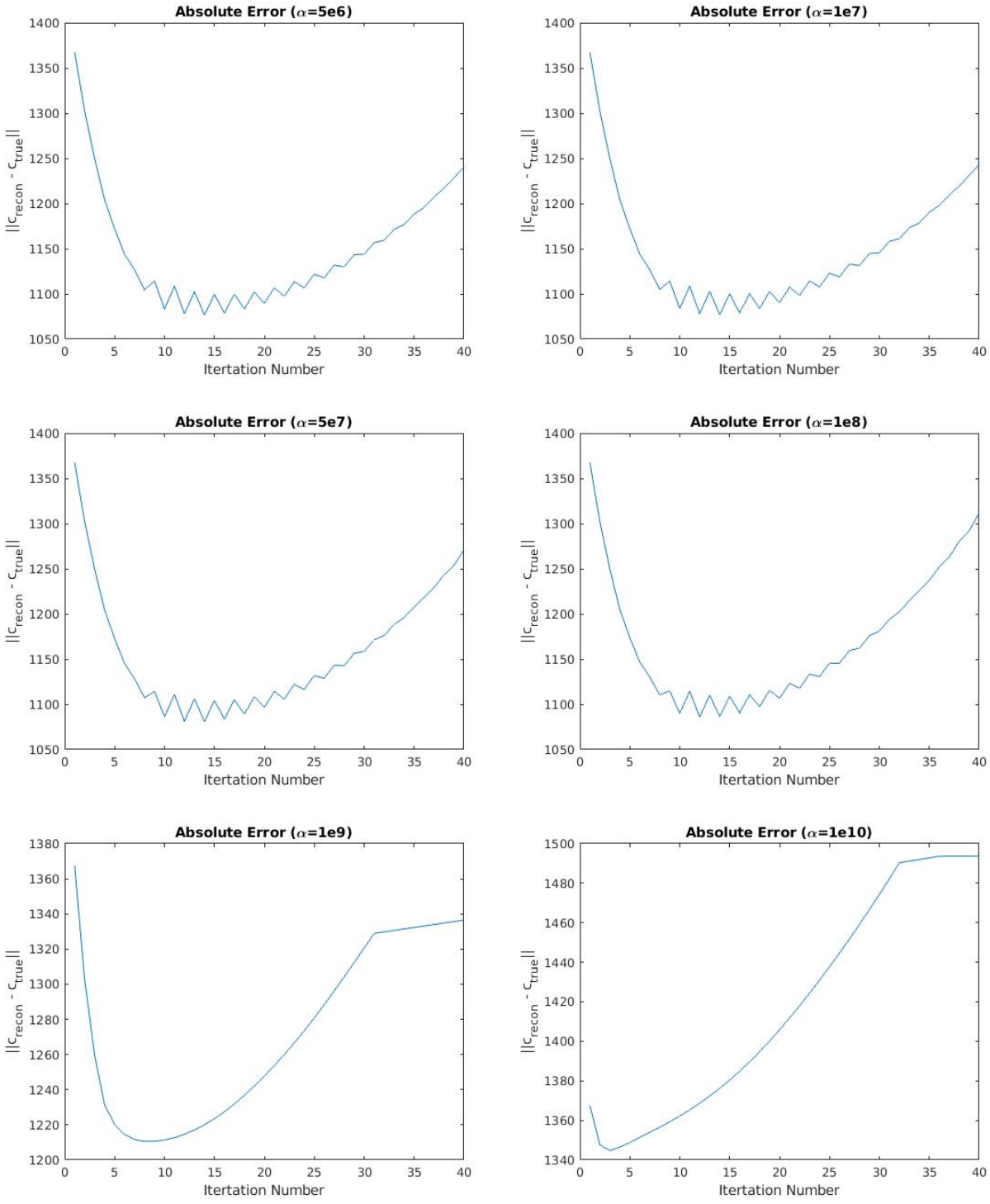


Figure 3.19: The error from ground truth $\|c_{\text{recon}} - c_{\text{true}}\|$ for six different TV regularization parameters α.

3.6 Lung Phantom

We now move on to displaying the results of Algorithm 1 on a phantom representing the cross section of a human thorax. This thorax consists of a heart, spine, one healthy lung, and one lung displaying a pleural effusion, which is a condition in which water is present between the lungs and the chest wall. We begin by describing the four fields necessary for data generation for this phantom.

3.6.1 Data Generation

kgrid: We again have an overall domain size of $0.1 \times 0.1m$. This is clearly smaller than a human thorax, but computational and time constraints precluded us from implementing the algorithm on a larger domain size. We note, however, that $0.1m$ is on the same order of magnitude as a more realistic human thorax (which would have a side length of roughly $0.4m$) and thus the results from the lung phantom are still illuminating. Due to the lack of fine details in the lung phantom, we use a coarser grid than was implemented for the breast. We set $Nx = 242$, meaning $dx = 4.1322 \times 10^{-4}$, and again for simplicity we set $Ny = Nx$ and $dy = dx$. In addition, data was also simulated on a grid of size $Nx = Ny = 340$, with $dx = dy = 2.9412 \times 10^{-4}$ in order to test the sensitivity of the algorithm to varying grid sizes.

medium: Four of the five properties defining the medium in k-Wave are identical to the phantom in section 3.5, with the main difference being the sound speed. The annotated lung sound speed phantom is shown in figure 3.20 and a table describing its tissue type and sound speed is given in table 3.2. The values in table 3.2 are all consistent with typically used sound speeds of human organs [91].

Figure 3.20: Lung phantom used to generate data.

Table 3.2: Parameters of the numerical lung phantom.

Structure Index	Tissue Type	Sound Speed (m/s)
1	Healthy Lung	1440
2	Pleural Effusion	1500
3	Heart	1560
4	Spine	1640
5	Adipose	1532
6	Background	1540

source: For the breast phantom, we assumed a measurement scenario in which the patient would place their breast in a water tank where a circular array of transducers surrounds the breast from a small distance. This is not practical when imaging the lungs, and instead we must develop the transducer locations to model a situation in which the sensors are placed directly on the human body. For this reason, we have the transducers form an ellipse, as in figure 3.21. This is clearly

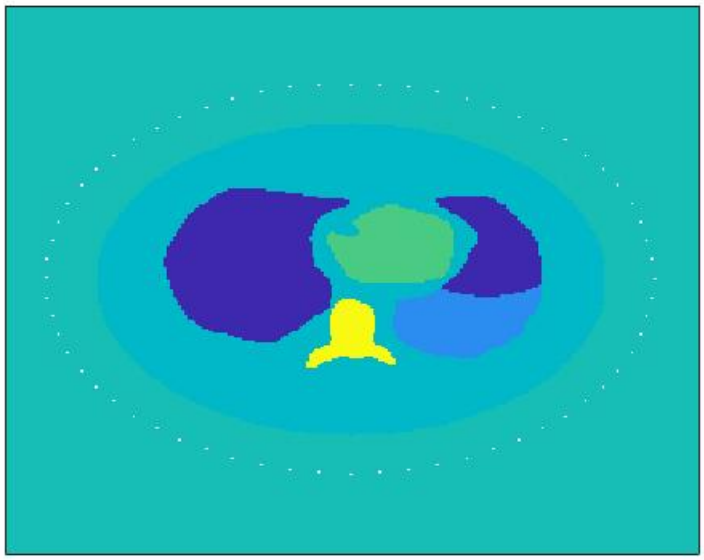


Figure 3.21: Transducer locations for K=64 surrounding the lung phantom.

an idealization of the boundary of the human body, but proved adequate for testing our reconstruction algorithm. We also point out that in our modeling the transducers do not directly touch the boundary of the human body, but are instead placed a small distance from this partition. This is a convenience in the reconstruction, as we do not want to have the transducers present in the region we are trying to reconstruct. While this is not entirely physically realistic, in an experimental setup acoustically conductive ultrasonic gel is used to improve acoustic contact between the transducers and the surface of the body. Thus a small gap between the transducers and boundary of the human thorax is not completely unreasonable.

Due to the lower complexity of the lung phantom and coarser grid used to simulate data, we used only $K = 64$ transducers for this setup instead of the $K = 128$ used for the breast phantom. The source for the lung phantom is similar to the source used for the breast phantom in that it is a wave packet appended with zeros. However, as noted in [12], only sound waves in the frequency range 10-750kHz are able to penetrate the lung pleura and thus provide data which will lead to accurate reconstructions. We therefore sent a wave packet constructed in an identical manner to

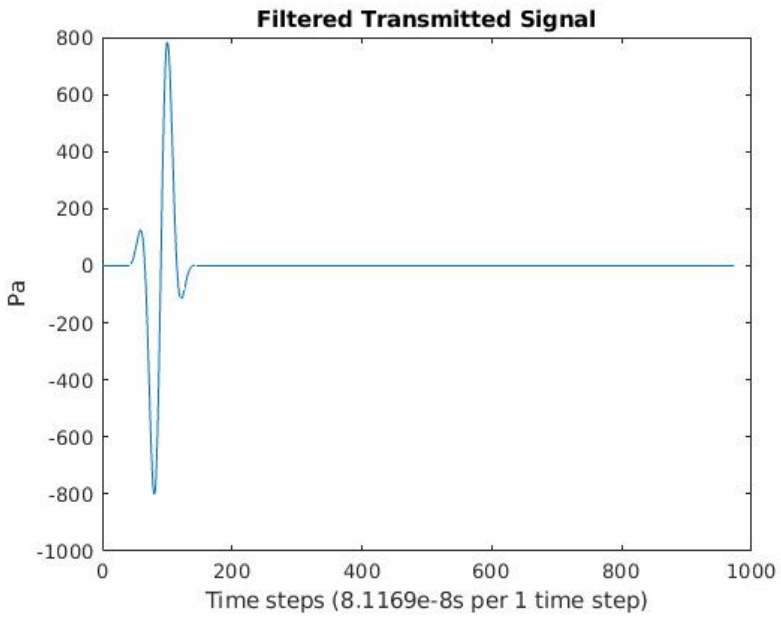


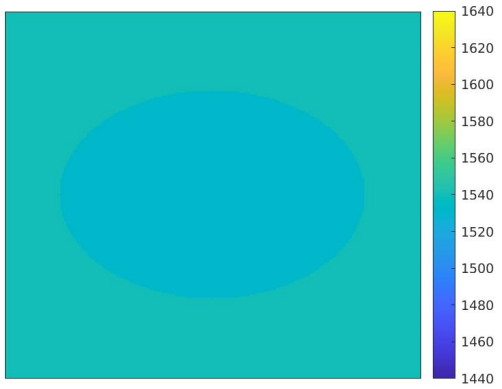
Figure 3.22: Filtered signal transmitted through a single transducer during the simulation of data for the lung phantom.

that in figure 3.5 but with the new lower central frequency of $250kHz$. This signal is shown in figure 3.22.

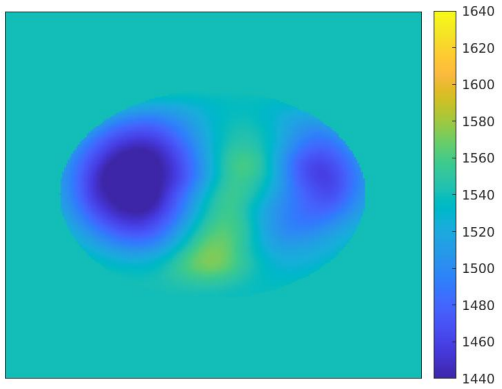
sensor: We again had each transducer act as both an emitter and receiver, which lead to a data set of size $64 \times 64 \times Nt$ where $Nt = 974$ for this simulation. Note that this data set is much smaller than the one used in section 3.5, meaning data generation and the inverse problem were completed in significantly less time.

3.6.2 Inverse Problem Setup

The inverse problem was solved on a grid of size 240×240 in order to avoid an inverse crime, and Gaussian noise $\eta \in \mathbb{R}^{Nt}$ was added to each time dependent pressure measurement, where η was again drawn from the distribution $\mathcal{N}(\mathbf{0}, \sigma\mathbb{I}_{Nt})$ with σ equal to $0.001 * \max(p_{\text{meas,1}})$. The same data calibration step was used as in section 3.5.2 on both the data generated on a grid of size 242×242 and the data generated on a grid of size 340×340. This calibration step may be more difficult for imaging the lungs in a real-world scenario as compared to the breast, since the imaging region no longer lies in a water bath. However, data could be collected on a dummy with



(a) Piecwese starting sound speed c_{init} in which the background sound speed and sound speed of the interior of the thorax are known exactly.



(b) Starting sound speed c_{init} obtained from a TOF algorithm.

Figure 3.23: Starting sound speeds c_{init} used for reconstructions of the lung phantom.

a known homogeneous sound speed and shape similar to the object being imaged. This would give both a data set with the region of interest present and a data set with a homogeneous sound speed, allowing the data calibration step to proceed without issue.

Three different regularization terms were tested, a Tikhonov regularization functional as in equation (3.3), a TV regularization term as in equation (3.6), and a Tikhonov+SSIM-EIT regularization function as in equation (3.23). For the Tikhonov regularization term alone, two different starting sound speeds c_{init} were investigated, one in which the sound speeds of the boundary and interior of the human body are assumed to be known exactly and the other obtained from a TOF reconstruction. These starting sound speeds c_{init} are shown in figure 3.23. Only the initial sound speed obtained from the TOF algorithm was used for the reconstructions obtained using the TV and Tikhonov+SSIM-EIT regularization term.

For the reconstructions using the TV regularization term, six different regularization parameters were tested: $\alpha = 10^6, 4 \cdot 10^6, 8 \cdot 10^6, 10^7, 5 \cdot 10^7, 10^8$. These values were obtained by running very few iterations with a large number of parameters and refining after evaluating the performance.

Six different regularization parameters were tested for the Tikhonov term, for a total of twelve different reconstructions (six for the homogeneous starting sound speed and six for the TOF start-

ing sound speed). These parameters were selected by first sweeping over a large range of values of α, evaluating preliminary performance, and then refining the values as necessary. This resulted in solutions for the parameters $\alpha = 10^3, 10^4, 10^5, 2 \cdot 10^5, 5 \cdot 10^5, 10^6$. These six parameters were used for reconstructions when the data was generated on a grid of size 242×242, but an additional Tikhonov reconstruction with $\alpha = 10^5$ was run when data was generated on a grid of size 340×340.

After obtaining results from using only the Tikhonov regularization term, the best performing parameter α was selected as $\alpha = 10^5$ and labeled α_t for use in the Tikhonov+SSIM-EIT reconstruction. The EIT phantom and conductivity reconstruction were provided by Dr. Talles Rattis Santos using one step of a Newton-Rhapson method with a high pass Gaussian filter as a regularizer. The domain for the EIT phantom is of size 0.1 m $\times$ 0.1 m $\times$ 0.0045 m. Thirty two electrodes with width 0.0013 m and height 0.0035 m were used to generate data. For the data generation, a skip-0 (adjacent) current pattern was used with a current amplitude of 10 mA using the complete electrode model in three dimensions [92] with electrode conductivity 0.02 S/m. One iteration of the Newton-Rhapson algorithm was run for the inverse problem with a regularization parameter of $1 \cdot 10^{-3}$. Voltages were obtained for both the phantom in figure 3.24a and the reference conductivity distribution in figure 3.24b. Those were used to give a reconstruction of the difference image in figure 3.24c, where the reference is considered to be muscle. The original conductivity difference reconstruction is shown in figure 3.24d and the same reconstruction adapted to our grid is given in figure 3.24e.

Unfortunately, this reconstruction has poor resolution and only vaguely identifies the outline of the lungs. This necessitated a careful selection of the SSIM-EIT regularization parameters α_s. Five different fixed values of α_s were tested: $\alpha_s = 10^5, 4 \cdot 10^5, 8 \cdot 10^5, 10^6$ and 10^7. Then, a sixth version of the algorithm was run with an adaptive regularization term α_s which began at $\alpha = 10^6$ and shrank linearly to $\alpha = 10^4$ throughout the duration of the algorithm. The scaled parameter was tested with the thought that due to the poor resolution of the EIT reconstruction, the SSIM-EIT

term may contribute more meaningfully near the beginning of the algorithm and be less necessary as Algorithm 1 began to more accurately reconstruct the size and shape of the organs.

Finally, The step lengths h_n were again chosen using the method outlined in section 3.4.1 with $s_{\max}$ set to 3. 40 iterations were run for each of the reconstructions without a stopping criterion. The results of the inverse problems on the lung phantom with these setups are displayed in the next section.

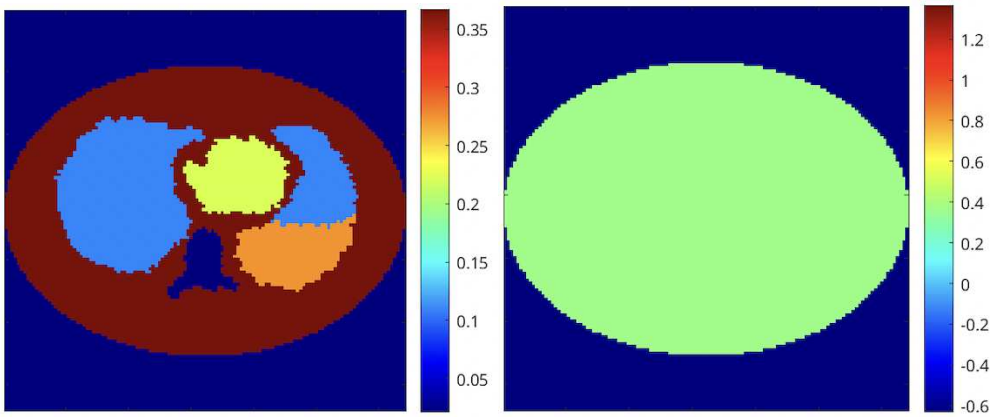
3.6.3 Results

Tikhonov

Reconstructions of the lung phantom for six different values of α with the homogeneous starting sound speed shown in figure 3.23a are shown in figure 3.25. Note that the reconstruction for $\alpha = 10^6$ seems to be completely homogeneous because the regularization term was far too large, causing the sound speed to drive towards the homogeneous background.

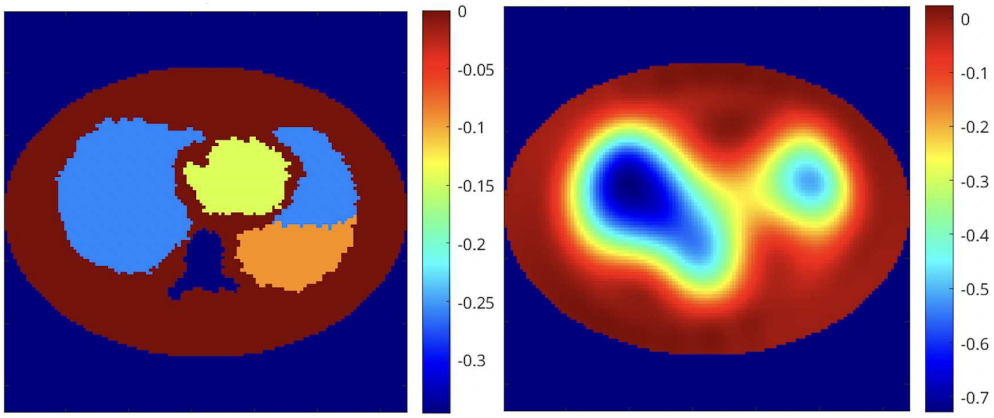
These reconstructions do a decent job of identifying the lungs, as both the healthy left lung and unhealthy right lung are visible in the image. However, the spine and heart are completely absent. Plots of the loss function $E(c)$ and Tikhonov regularization term $R(c)$ are shown in figures 3.26 and 3.27. The plots of $E(c)$ show that all values of α lead to a decrease in the loss function with the exception of $\alpha = 10^6$, suggesting that reconstruction is over-regularized. This is confirmed by viewing the plots of $R(c)$, which show the values of the Tikhonov regularization term increasing for each of the parameters α, with the exception of $\alpha = 10^6$.

Figure 3.28 displays the difference in norm between the reconstruction at each iteration and the true lung phantom. Again, we cannot calculate this error in a real-world situation, but since we know the ground truth in this case it may prove insightful. These plots show that a lower regularization parameter leads to a decrease in the absolute error up to a certain level. The shape of the plots also match well with those in 3.26, suggesting that the error term presented in those plots is a good indicator of an accurate reconstruction.



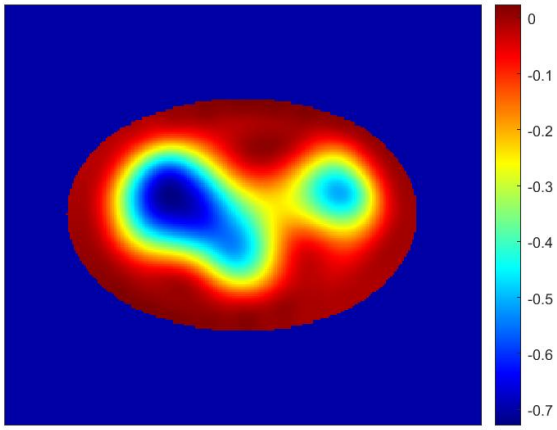
(a) EIT conductivity phantom.

(b) EIT reference phantom.



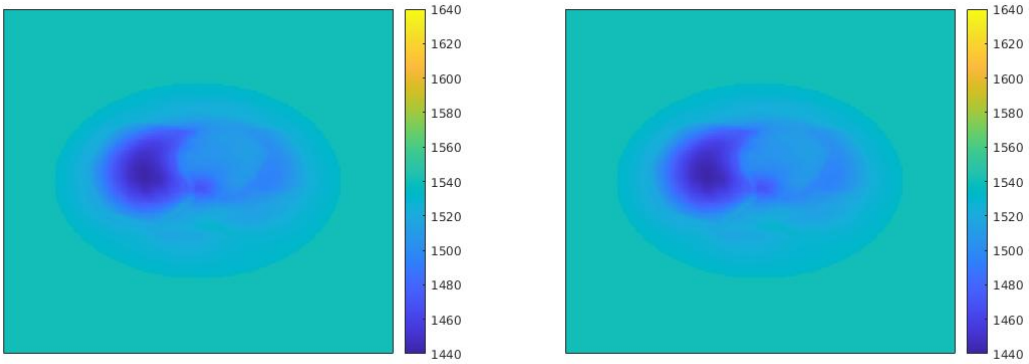
(c) EIT difference image ground truth.

(d) Original EIT conductivity difference reconstruction.



(e) EIT conductivity difference reconstruction adapted to our grid.

Figure 3.24: EIT phantom, reference phantom, difference phantom, original conductivity reconstruction, and conductivity map adapted to our grid.



(a) $\alpha = 10^3$ **(b)** $\alpha = 10^4$

(c) $\alpha = 10^5$ **(d)** $\alpha = 2 \cdot 10^5$

(e) $\alpha = 5 \cdot 10^5$ **(f)** $\alpha = 10^6$

Figure 3.25: Reconstructions from noisy data after 40 iterations for six different Tikhonov regularization parameters α with a piecewise starting sound speed c_{init}.

While the reconstructions obtained for a homogeneous sound speed c_{init} were not complete nonsense, their lack of detail in the spine and heart suggest that a closer starting sound speed c_{init} may prove beneficial.

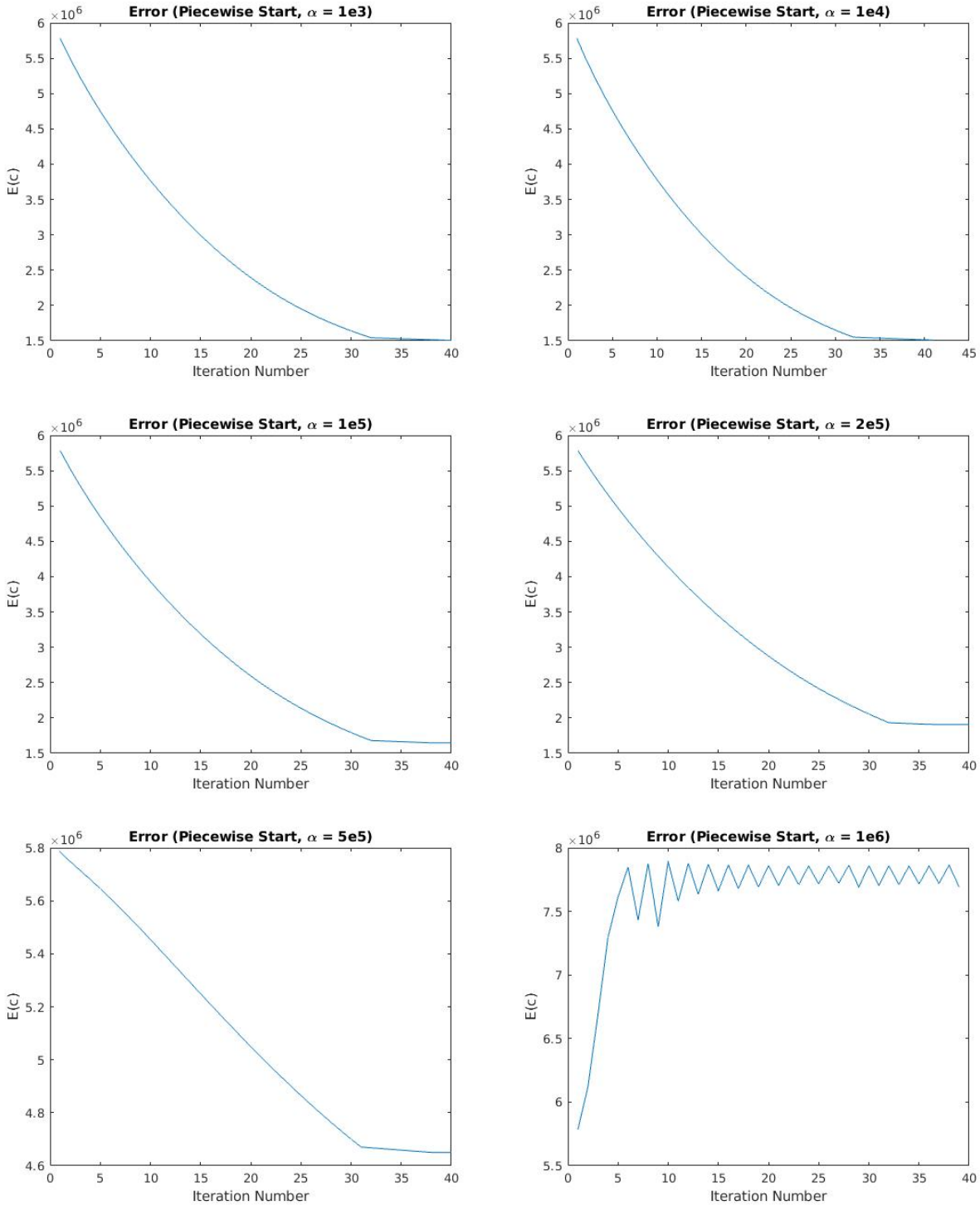


Figure 3.26: Values of the loss function $E(c)$ for six different Tikhonov regularization parameters α with a piecewise starting sound speed c_{init}.

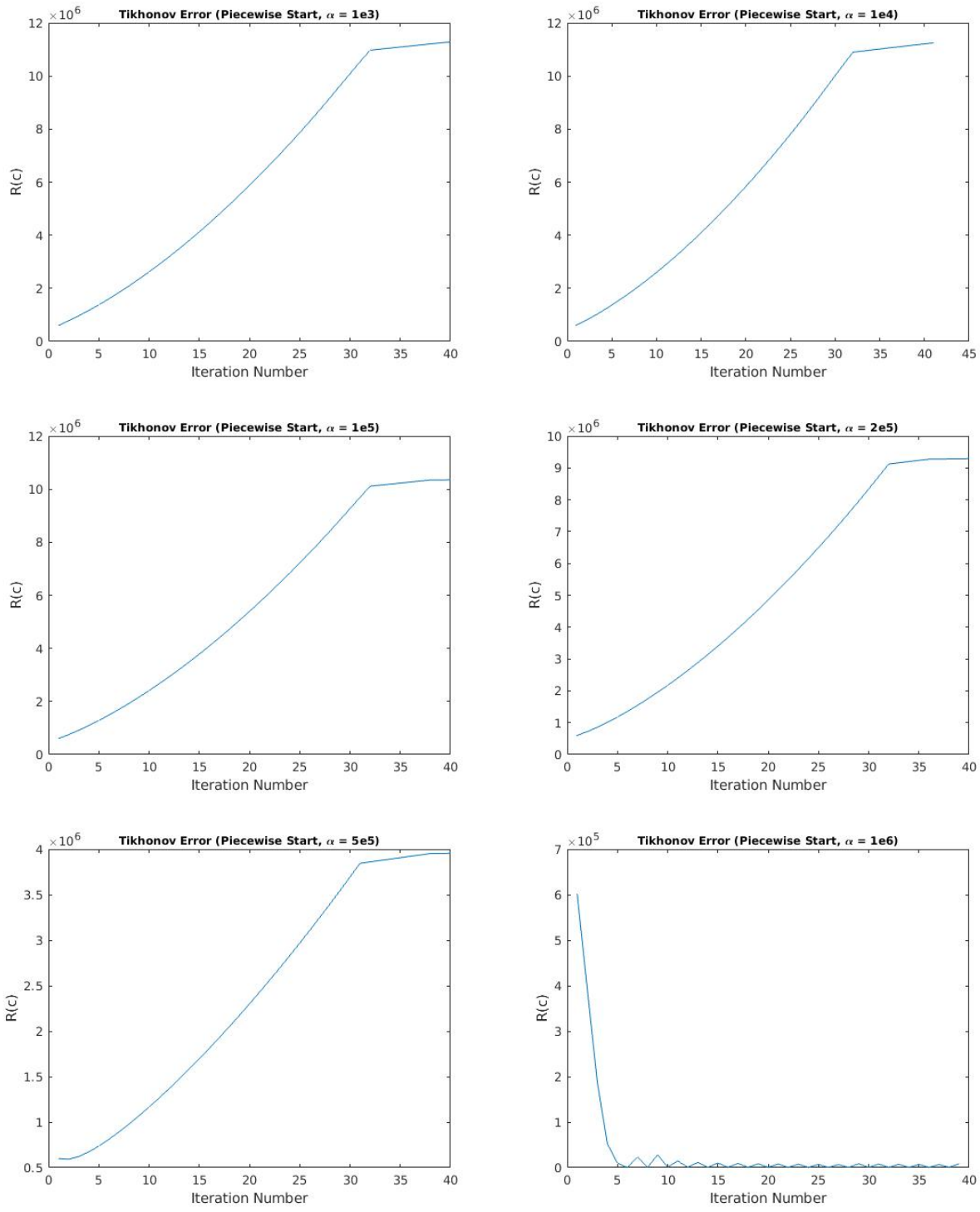


Figure 3.27: Values of the Tikhonov regularization term $R(c)$ for six different regularization parameters α with a piecewise starting sound speed c_{init}.

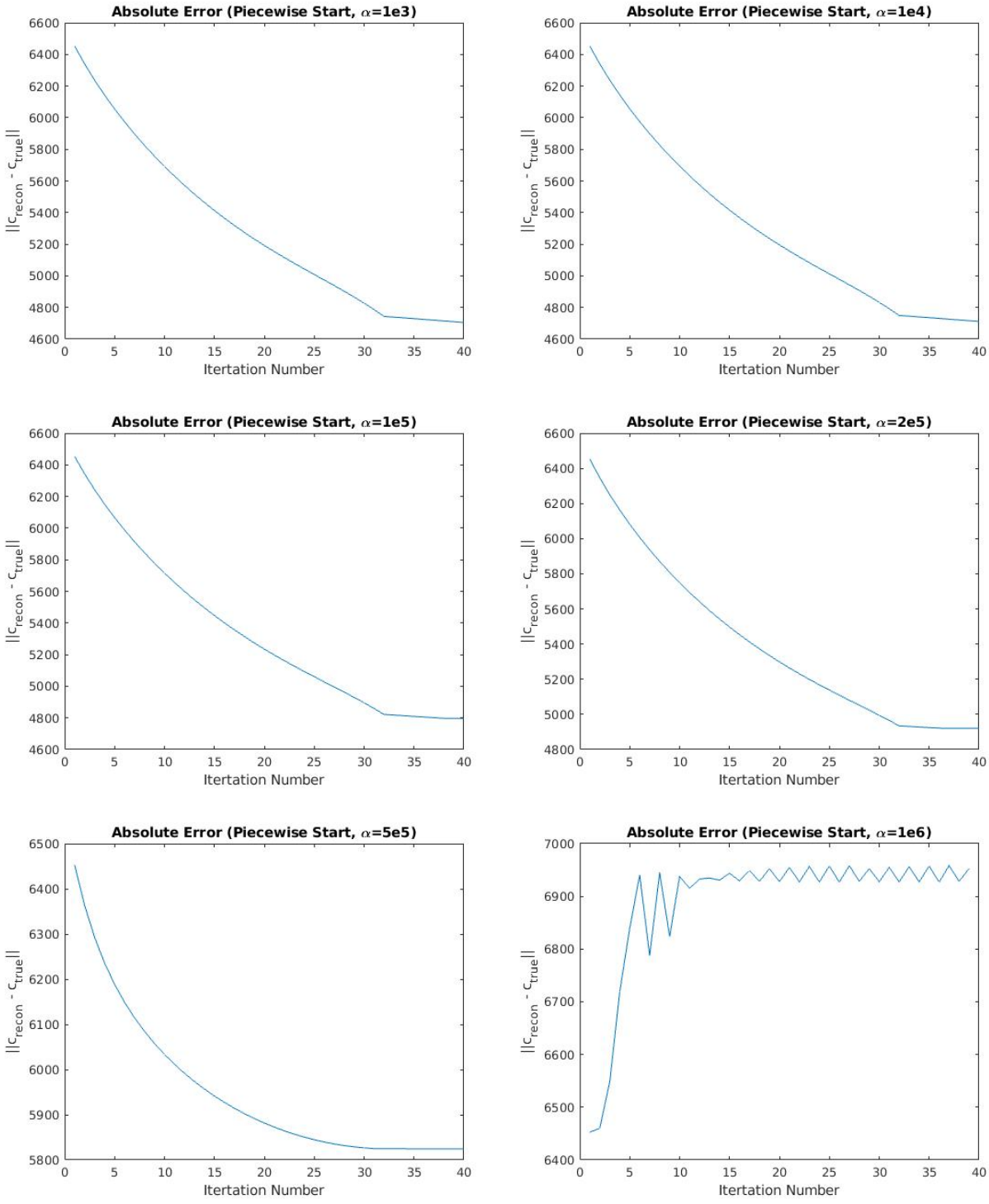


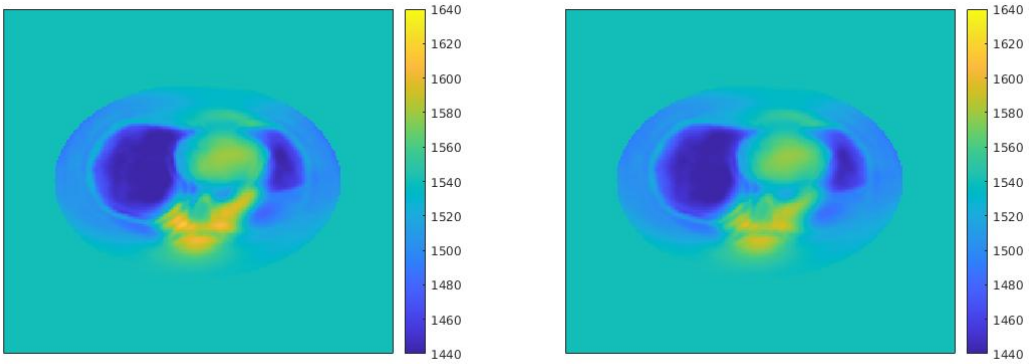
Figure 3.28: The error from ground truth $\|c_{\text{recon}} - c_{\text{true}}\|_2$ for six different Tikhonov regularization parameters α with a piecewise starting sound speed c_{init}.

The results of using c_{init} obtained from a TOF algorithm are shown in figure 3.29. These sound speed maps show much better agreement with the true sound speed profile, as the lungs are almost completely resolved and the spine and heart are present at lower values of α. The reconstruction at $\alpha = 10^6$ shows clear signs of over-regularization.

Plots of the loss function $E(c)$ and Tikhonov regularization function $R(c)$ are presented in figures 3.30 and 3.31. These plots are encouraging, as multiple values of α show a decrease in both the loss function and Tikhonov regularization term. This suggests that any of the reconstructions for $\alpha = 10^4, 10^5, 2 \cdot 10^5$ are sufficiently accurate both qualitatively and quantitatively. We believe that $\alpha = 10^5$ displays the best ability to accurately reconstruct the size, shape, and sound speed values of the organs, in addition to very clearly exhibiting the pleural effusion in the right lung.

Figure 3.32 displays plots of the difference in norm between the reconstructed sound speed and true sound speed at each iteration. These plots suggest that the most accurate reconstruction is at $\alpha = 2 \cdot 10^5$ with $\alpha = 10^5$ also performing very well. This matches our qualitative assessment and also matches the performance seen from the error plots in figure 3.30.

Finally, figure 3.33 displays the reconstructed sound speed map and all errors plots for the inversion calculated from data generated on a grid of size 340×340. This reconstruction again looks excellent qualitatively, and the error plots show that the algorithm had no issue with the larger change in grids between data generation and the solution to the inverse problem. Of note is that the error plot $E(c)$ does not hit the same minimum as in figure 3.30c, but the absolute error plot actually achieves a lower minimum than in figure 3.32c. This is encouraging, and suggests that for this particular phantom and setup Algorithm 1 is robust against experimental error.



(a) $\alpha = 10^3$ **(b)** $\alpha = 10^4$

(c) $\alpha = 10^5$ **(d)** $\alpha = 2 \cdot 10^5$

(e) $\alpha = 5 \cdot 10^5$ **(f)** $\alpha = 10^6$

Figure 3.29: Reconstructions from noisy data after 40 iterations for six different Tikhonov regularization parameters α with a starting sound speed c_{init} obtained from a TOF algorithm.

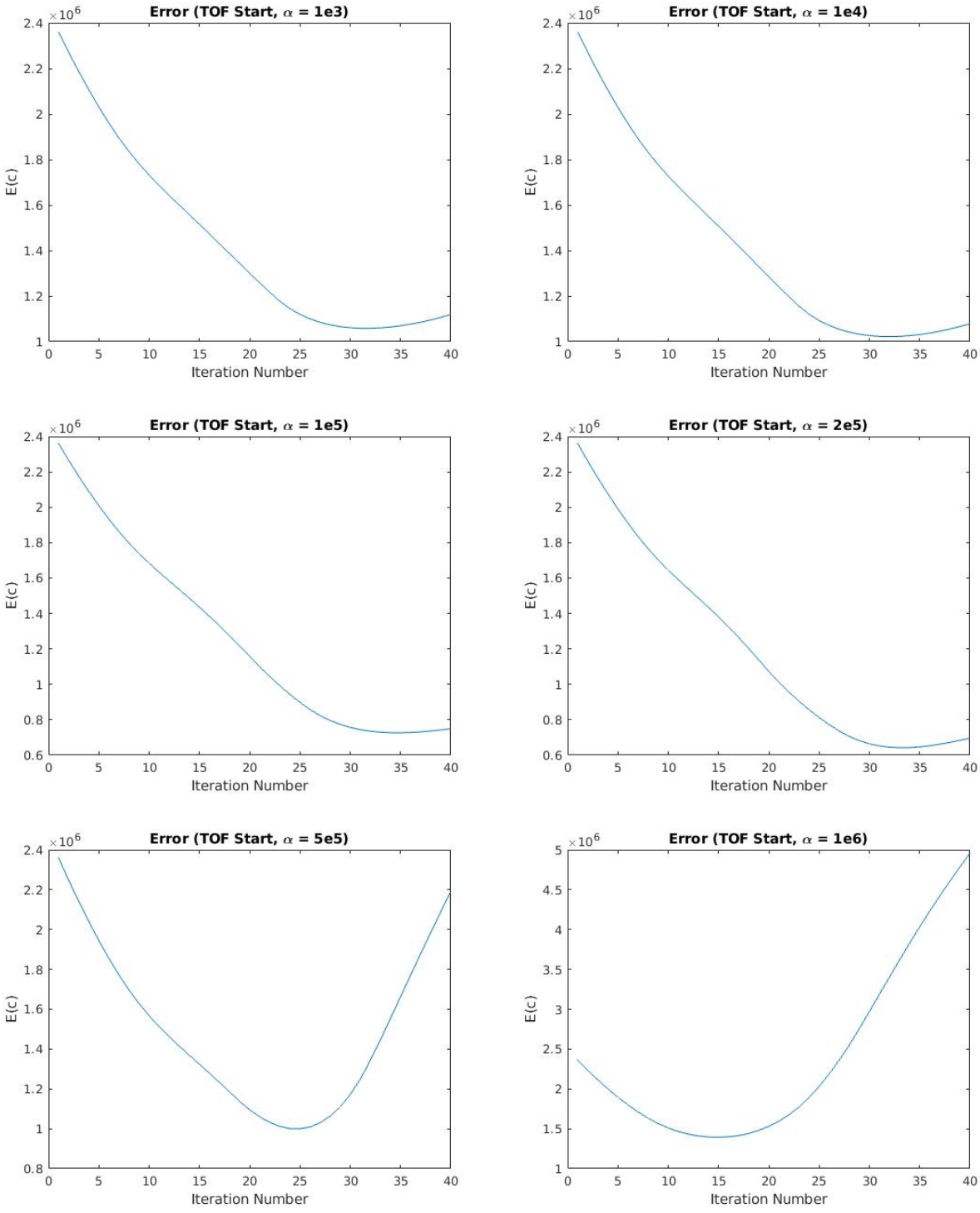


Figure 3.30: Values of the loss function $E(c)$ for six different Tikhonov regularization parameters α with a starting sound speed c_{init} obtained using a TOF algorithm.

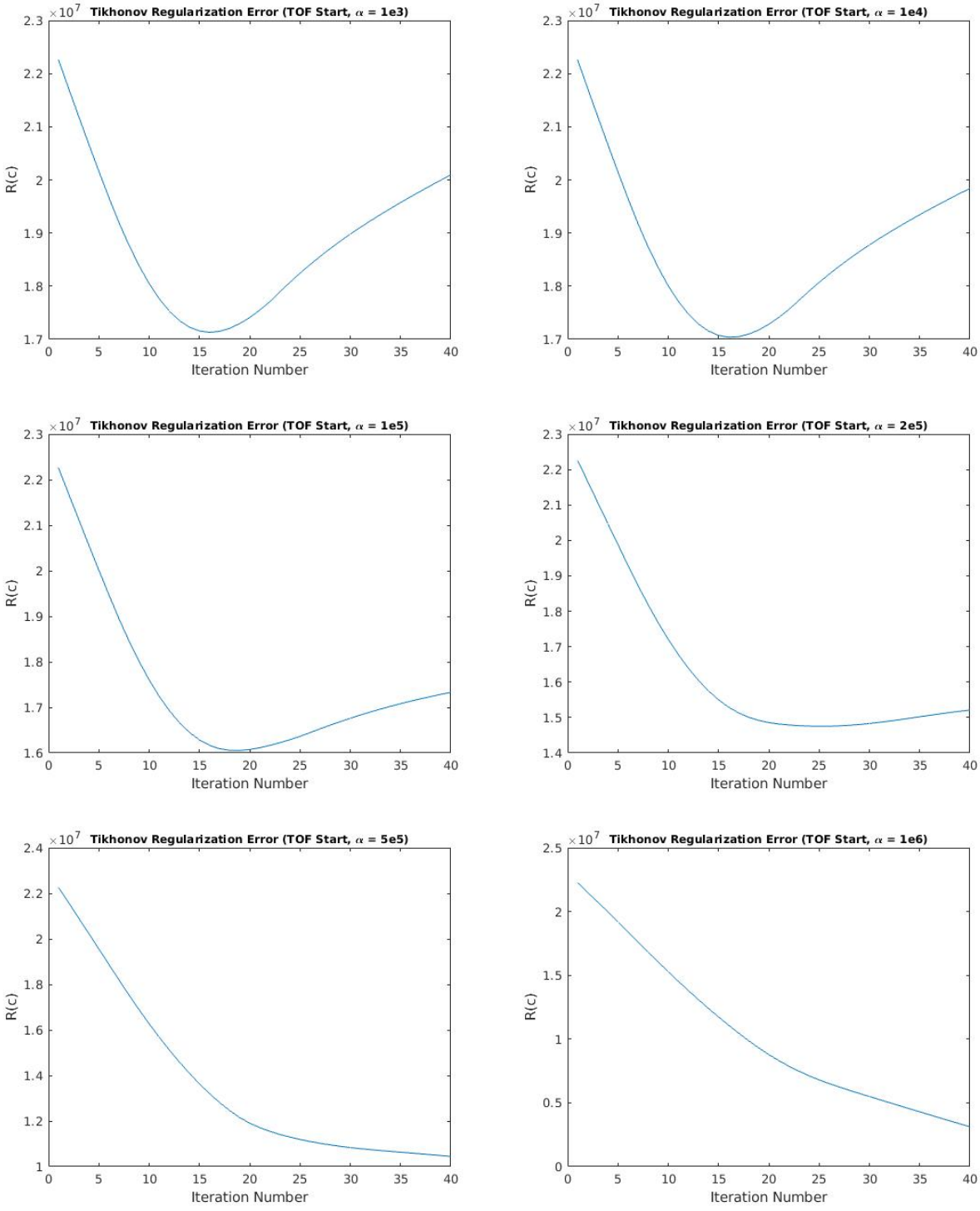


Figure 3.31: Values of the Tikhonov regularization term $R(c)$ for six different regularization parameters α with a starting sound speed c_{init} obtained using a TOF algorithm.

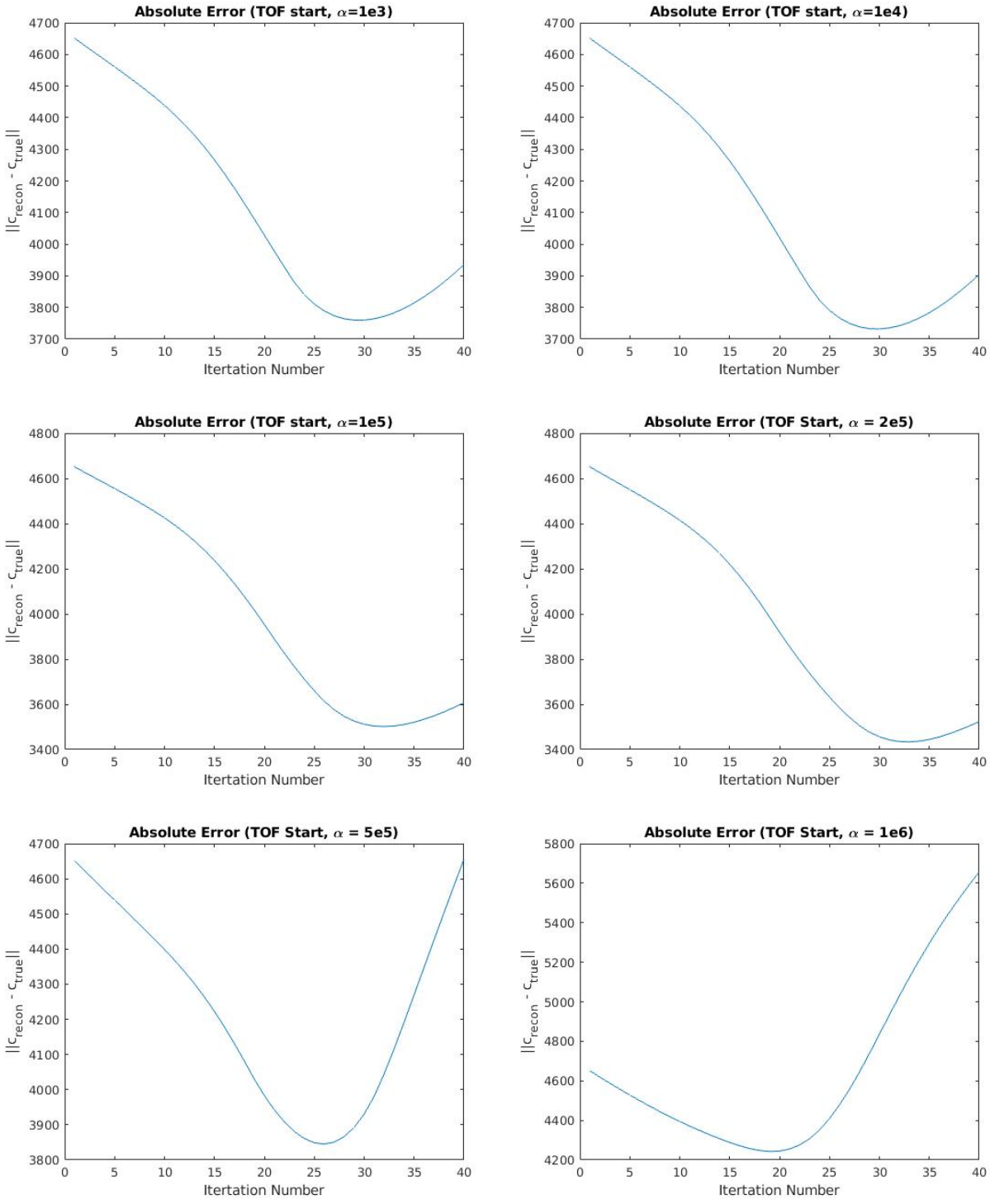


Figure 3.32: The error from ground truth $\|c_{\text{recon}} - c_{\text{true}}\|_2$ for six different Tikhonov regularization parameters α with a starting sound speed c_{init} obtained from a TOF algorithm.

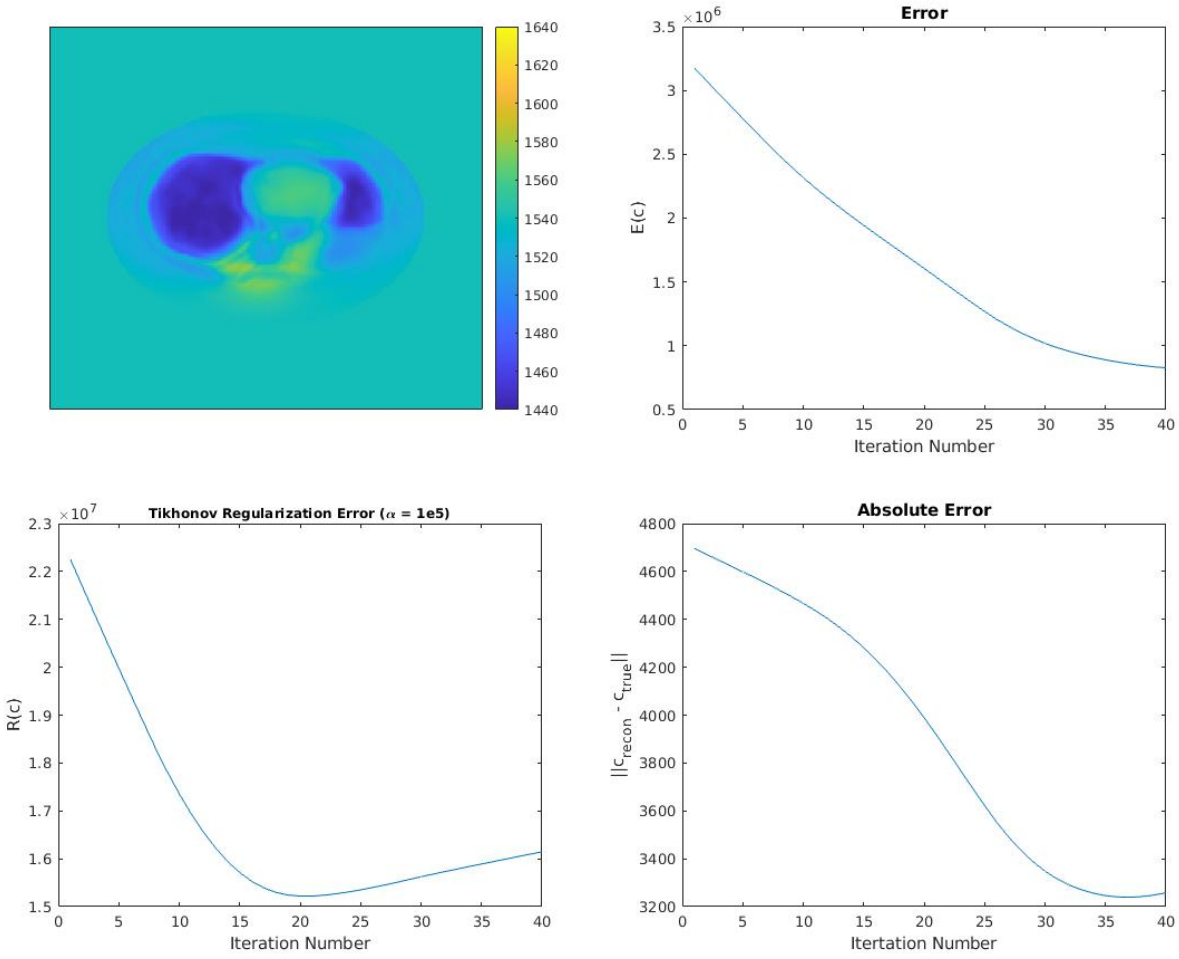


Figure 3.33: The reconstruction, error plot $E(c)$, Tikhonov error plot $R(c)$, and absolute error for regularization parameter $\alpha = 10^5$ after 40 iterations for the lung phantom using noisy data generated on a grid of size 340×340.

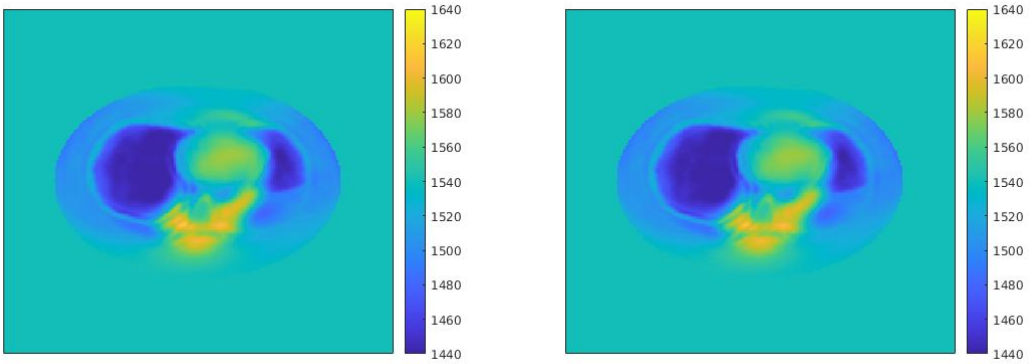
TV

Results obtained using a TV regularization term for six different values of the regularization parameter α are shown in figure 3.34. These results do not differ greatly qualitatively from the results shown in figure 3.29, but seem to show a bit better edge reconstruction when $\alpha = 8{\cdot}10^6, 10^7$. The sound speed maps for $\alpha = 5{\cdot}10^7, 10^8$ display an artifact along the right boundary of the thorax, suggesting those results were over-regularized.

Plots of the error term $E(c)$, the TV regularization term $R(c)$, and the l^2 error of the difference between the reconstruction and ground truth at each iteration are shown in figures 3.35, 3.36, and 3.37. These plots are mostly as expected, but note that the plots of the error term $E(c)$ in 3.35 do not ever fall below 1 as they do in figure 3.30, and the absolute error plots in 3.37 also do not achieve the same minimums as in figure 3.32. This suggests that for this particular phantom and experimental setup, the Tikhonov regularization term out-performed the TV regularization term with respect to our error quantifications.

Tikhonov+SSIM-EIT

The sound speed reconstructions using the Tikhonov+SSIM-EIT regularization terms are shown in figure 3.38. These results are very intriguing, as they resemble the Tikhonov reconstruction in figure 3.29c, but seem to do a better job of bringing clarity and definition to the organs. In particular, the reconstructions for parameters $\alpha_s = 10^5, 4 \cdot 10^5, 8 \cdot 10^5$ and the adaptive (scaled) parameter display good edge reconstruction for the lungs, spine, and heart as compared to the Tikhonov parameter alone. When $\alpha_s = 10^7$, the reconstruction is overly influenced by the EIT prior and matches its structure to the detriment of the image quality. Figures 3.39, 3.40, and 3.41 display steps 7, 14, 21, and 32 of Algorithm 1 for the SSIM-EIT regularization parameters $\alpha_s = 10^5, 8{\cdot}10^5$, and the scaled α_s. These figures reveal that the reconstructions are quite similar at the seventh step, and only begin to differ greatly at around step 21. They also show that the intermediate reconstructions do not appear better qualitatively than the final reconstruction, suggesting that the error term $E(c)$ gives a good indication of image quality.



(a) $\alpha = 10^6$ **(b)** $\alpha = 4 \cdot 10^6$

(c) $\alpha = 8 \cdot 10^6$ **(d)** $\alpha = 10^7$

(e) $\alpha = 5 \cdot 10^7$ **(f)** $\alpha = 10^8$

Figure 3.34: Reconstructions from noisy data after 40 iterations for six different TV regularization parameters α.

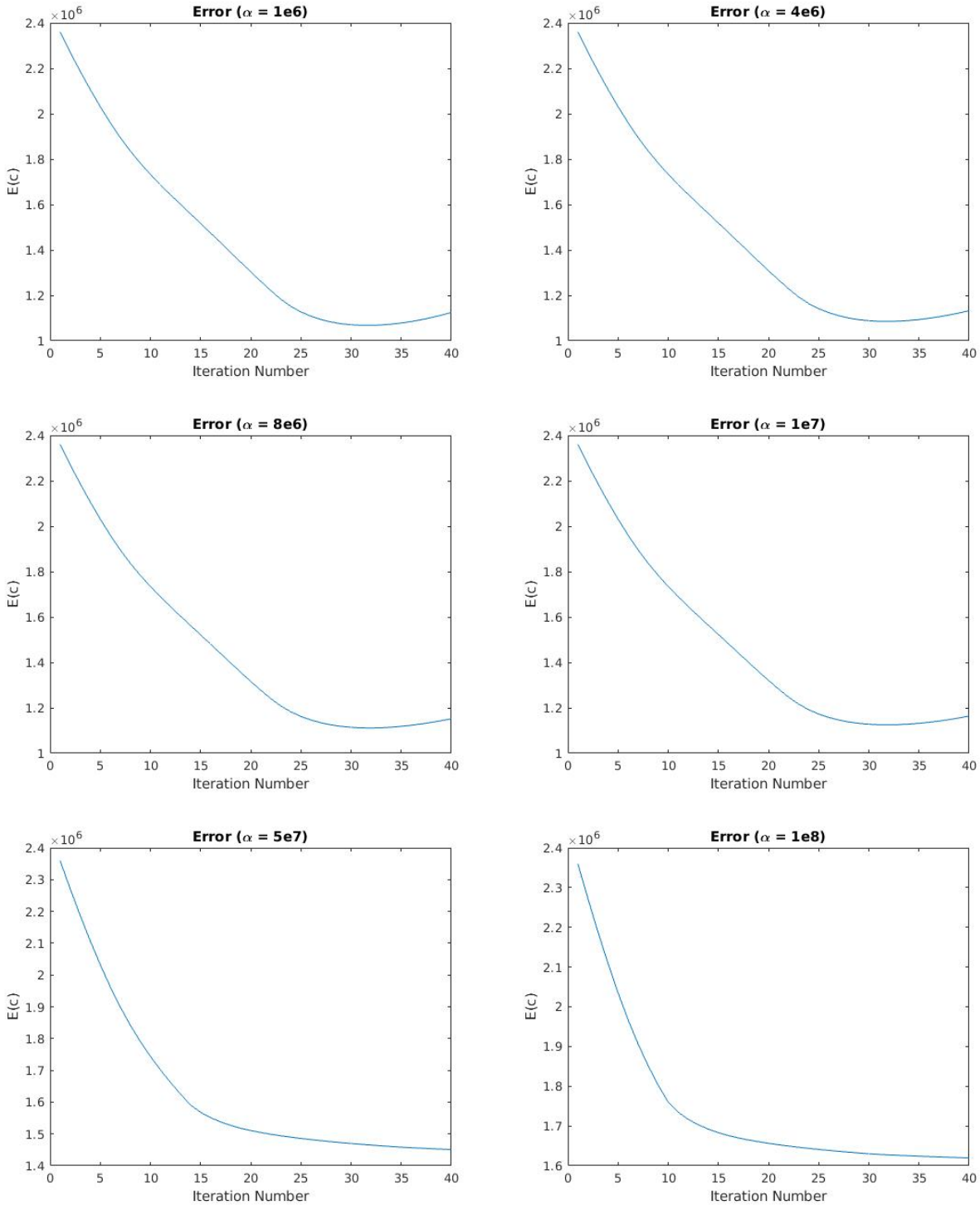


Figure 3.35: Values of the loss function $E(c)$ for six different TV regularization parameters α.

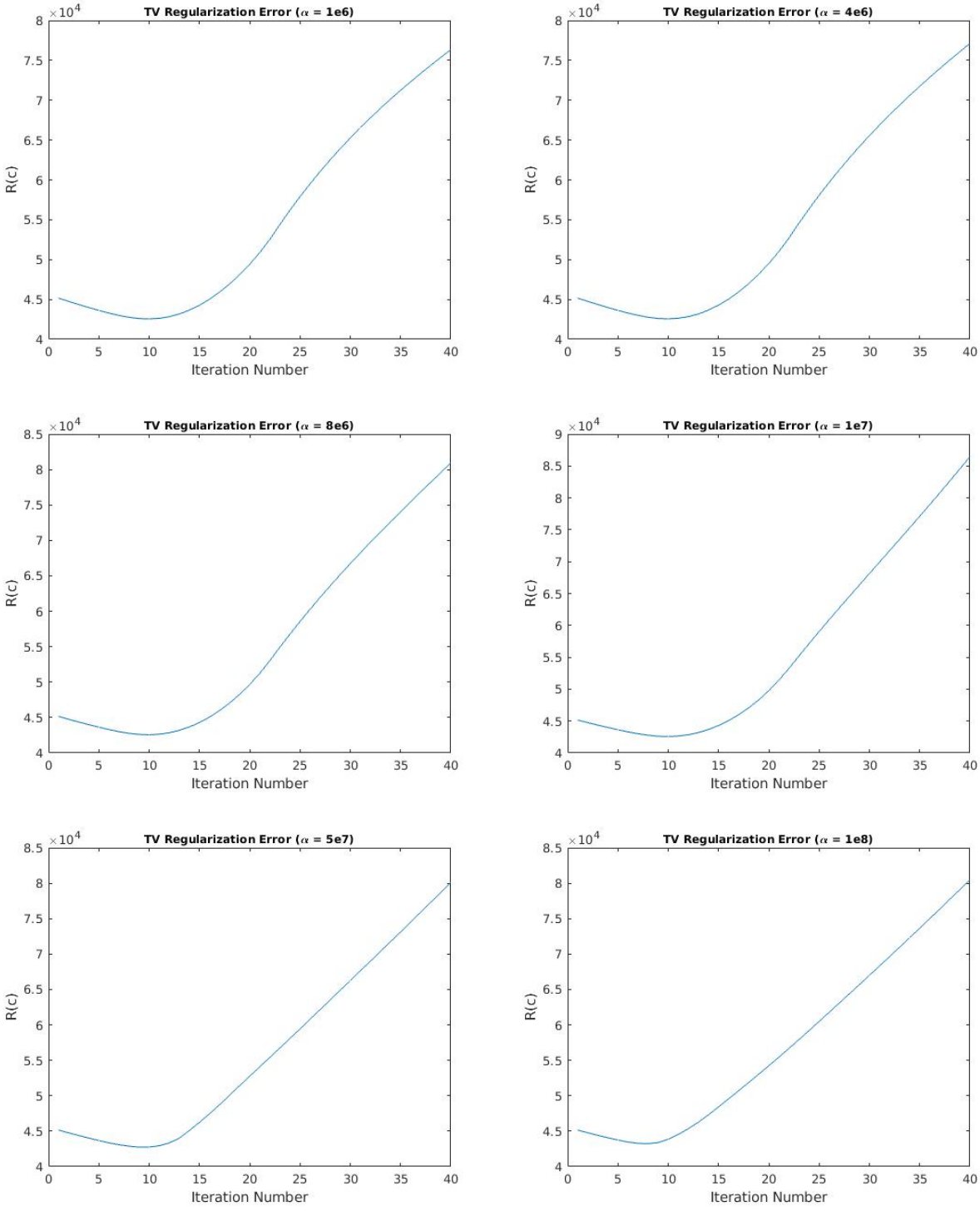


Figure 3.36: Values of the TV regularization term $R(c)$ for six different regularization parameters α.

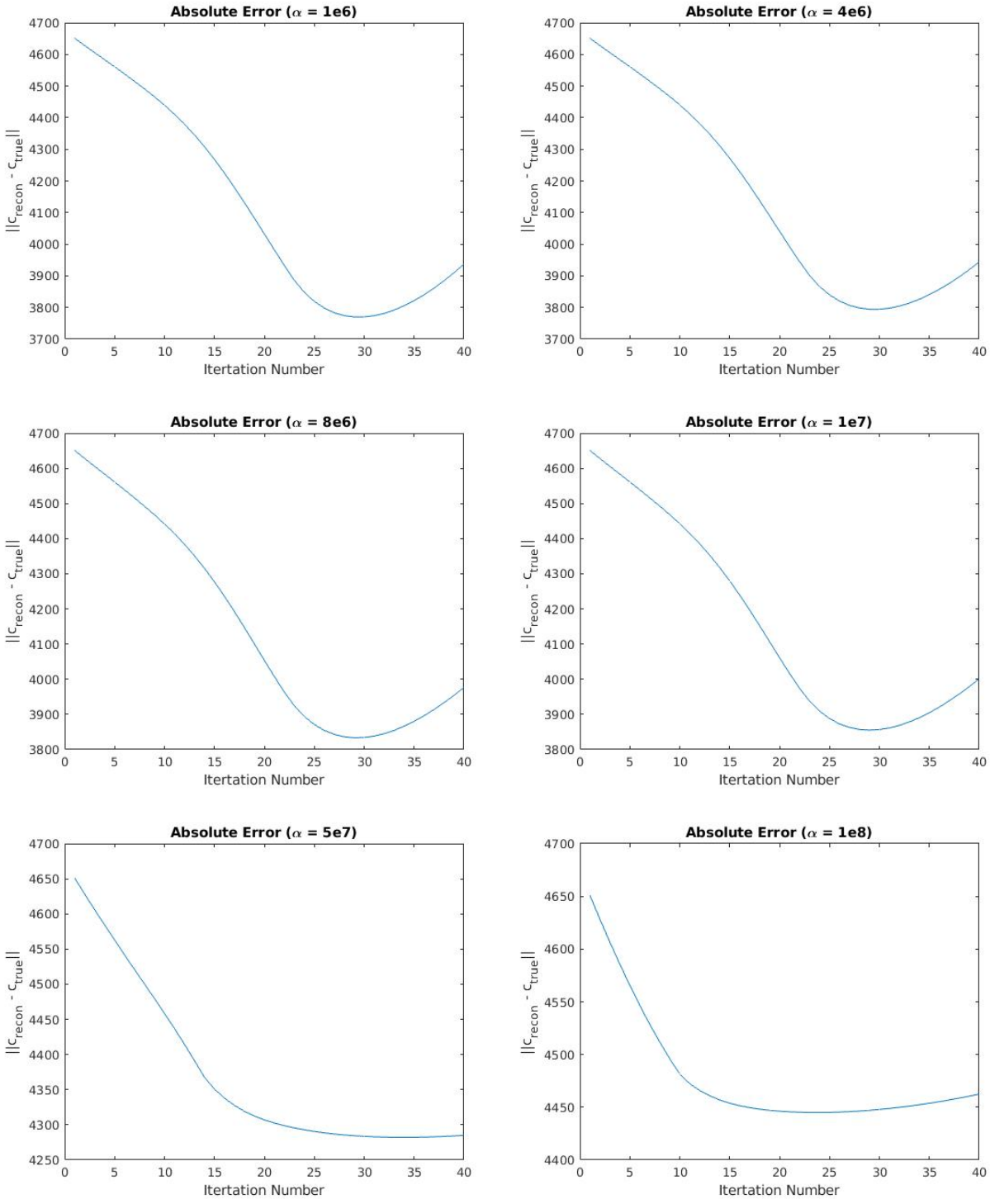
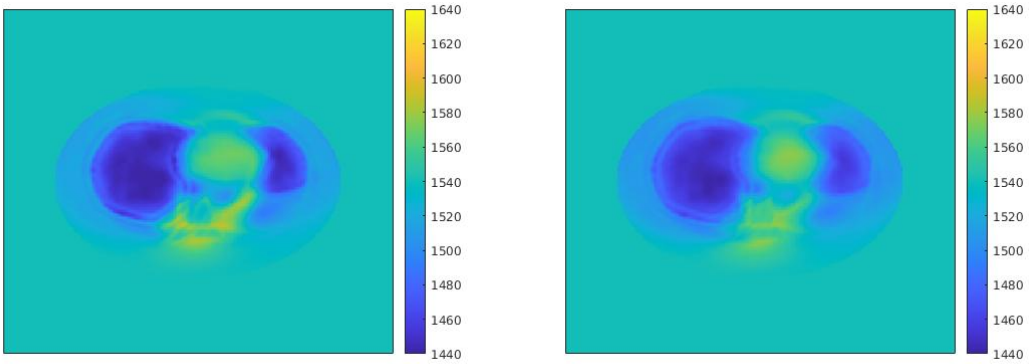


Figure 3.37: The error from ground truth $\|c_{\text{recon}} - c_{\text{true}}\|_2$ for six different TV regularization parameters α.

Figures 3.42, 3.43, and 3.44 display the functions $E(c)$, $R_{\text{tikh}}(c)$ and $R_{\text{ssim}}(c)$ at each iteration. Interestingly, the SSIM-EIT regularization errors in figure 3.44 change very little throughout the algorithm, despite the notable affect they have qualitatively on the reconstructions. Also of note is that the error terms in figures 3.42a and 3.42f decrease faster than in the case of only the Tikhonov regularization term with $\alpha = 10^5$ shown in figure 3.30c. This is exciting, as it suggests that (when properly tuned) the SSIM-EIT regularization term can decrease the error term $E(c)$ more quickly when compared to only Tikhonov regularization.

Figure 3.45 shows the norm of the difference between the sound speed reconstruction at each iteration and the ground truth. The absolute error plots when $\alpha = 10^5$ and when using the scaled α show a greater decrease in the absolute error when compared with Tikhonov-only reconstructions. The fact that figure 3.45f also shows that the absolute error is still decreasing after 40 iterations suggests that an adaptive regularization parameter for the SSIM-EIT prior holds great promise.

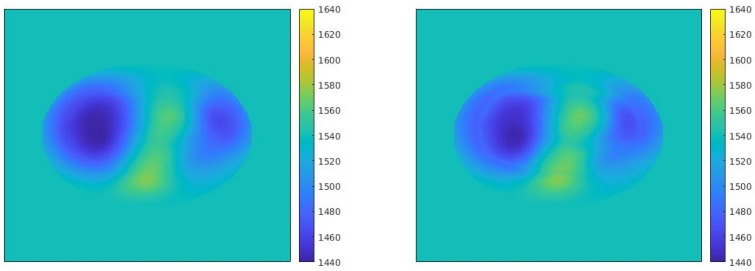


(a) $\alpha_s = 10^5$ **(b)** $\alpha_s = 4 \cdot 10^5$

(c) $\alpha_s = 8 \cdot 10^5$ **(d)** $\alpha_s = 10^6$

(e) $\alpha_s = 10^7$ **(f)** scaled α_s

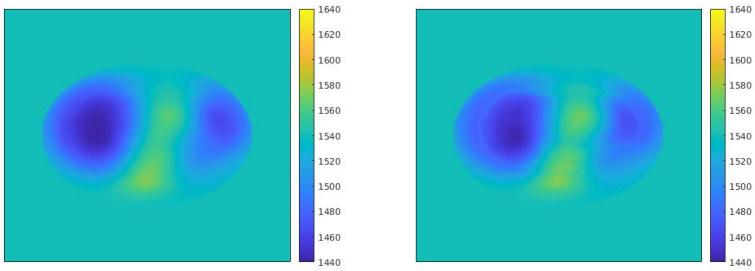
Figure 3.38: Reconstructions from noisy data after 40 iterations for six different SSIM-EIT regularization parameters α_s with $\alpha_t = 10^5$ and a starting sound speed c_{init} obtained from a TOF algorithm.



(a) $\alpha_s = 10^5$, Iteration 7 **(b)** $\alpha_s = 10^5$, Iteration 14

(c) $\alpha_s = 10^5$, Iteration 21 **(d)** $\alpha_s = 10^5$, Iteration 32

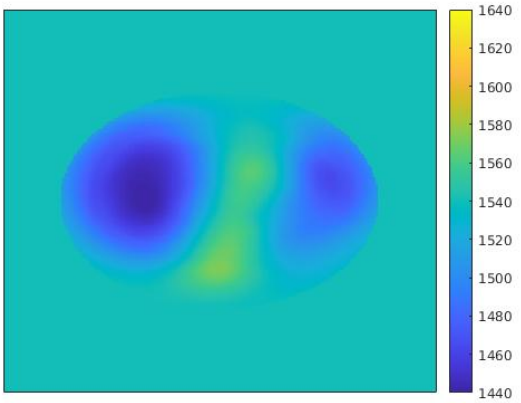
Figure 3.39: Intermediate reconstructions at iterations 7, 14, 21, and 32 for $\alpha_s = 10^5$.



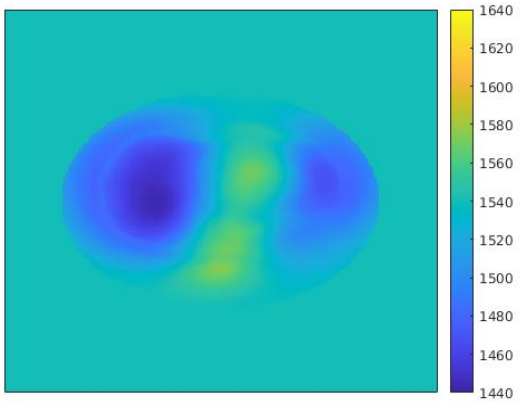
(a) $\alpha_s = 8 \cdot 10^5$, Iteration 7 **(b)** $\alpha_s = 8 \cdot 10^5$, Iteration 14

(c) $\alpha_s = 8 \cdot 10^5$, Iteration 21 **(d)** $\alpha_s = 8 \cdot 10^5$, Iteration 32

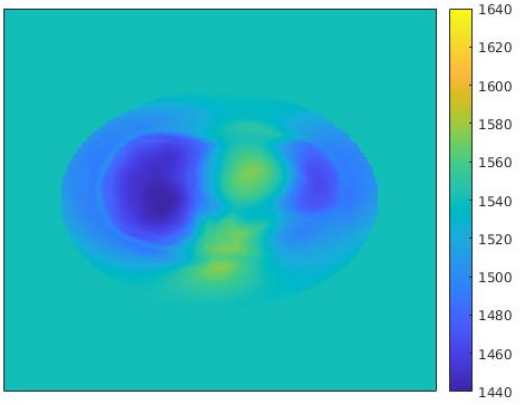
Figure 3.40: Intermediate reconstructions at iterations 7, 14, 21, and 32 for $\alpha_s = 8 \cdot 10^5$.



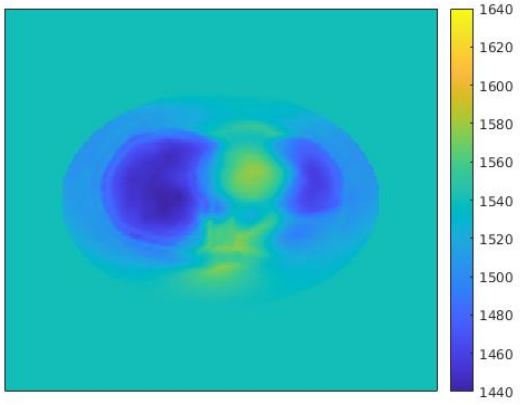
(a) Scaled α_s, Iteration 7



(b) Scaled α_s, Iteration 14



(c) Scaled α_s, Iteration 21



(d) Scaled α_s, Iteration 32

Figure 3.41: Intermediate reconstructions at iterations 7, 14, 21, and 32 for scaled α_s.

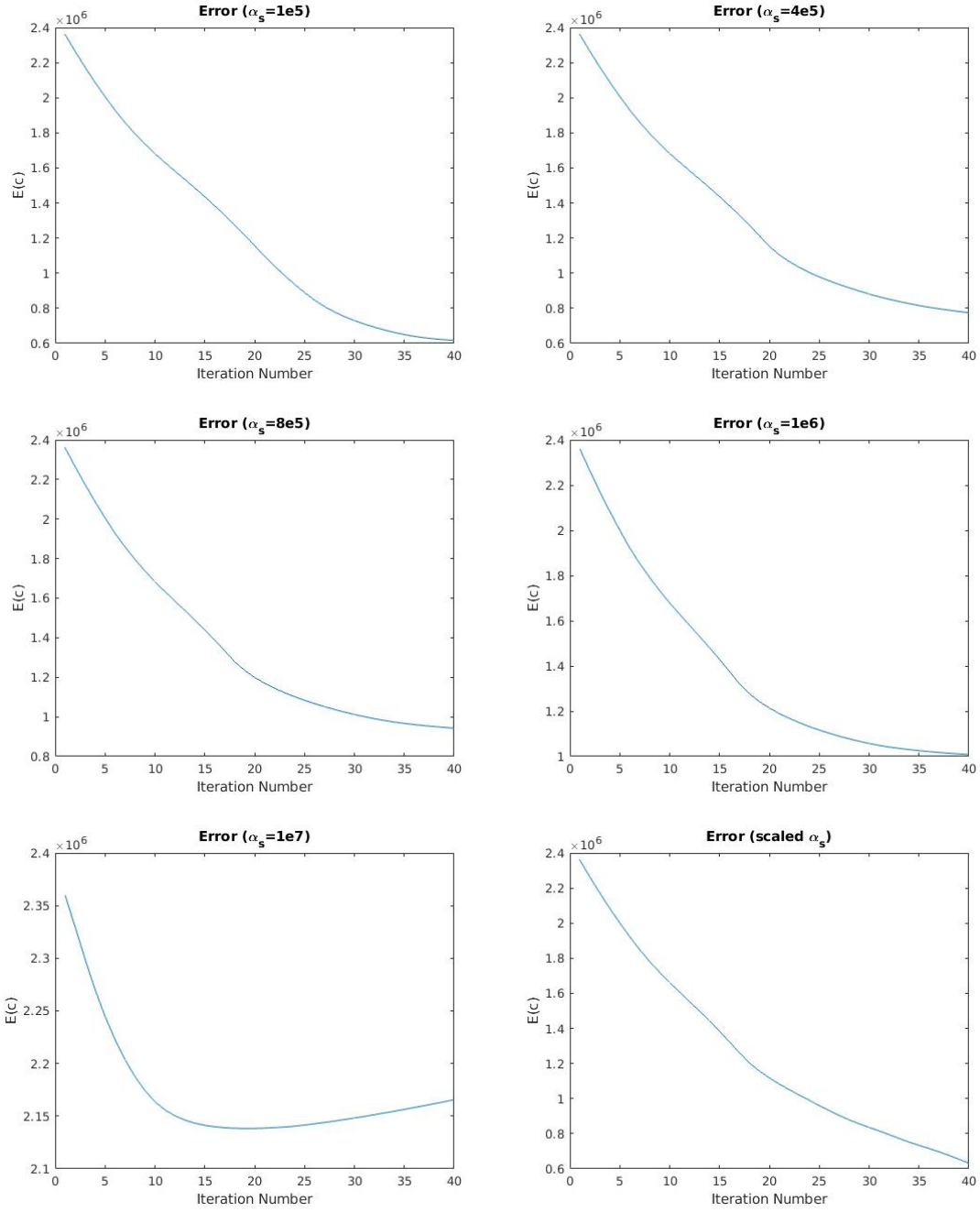


Figure 3.42: Values of the loss function $E(c)$ for six different SSIM regularization parameters α_s.

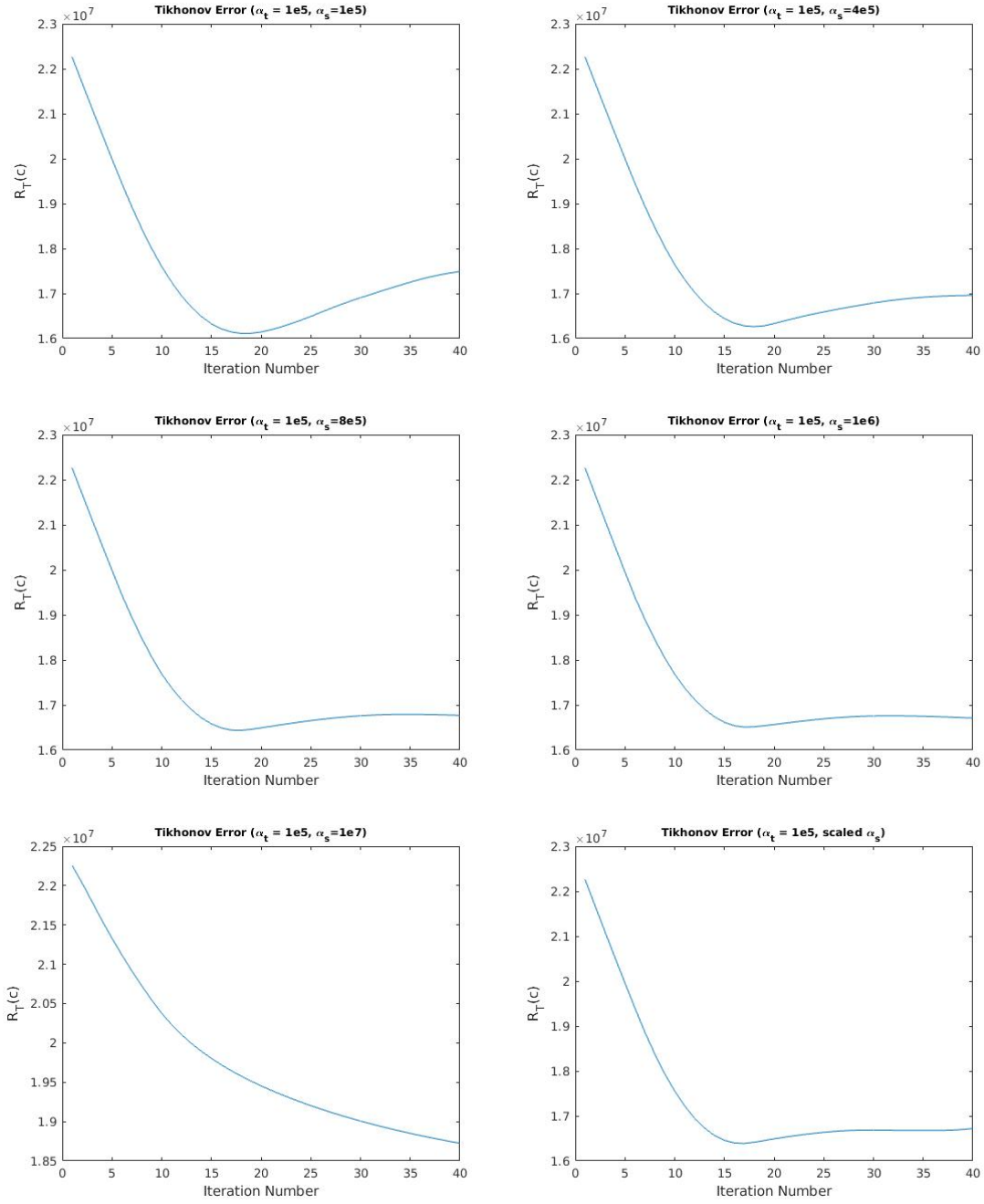


Figure 3.43: Values of the Tikhonov regularization term $R_T(c)$ for six different regularization parameters α_s and $\alpha_t = 10^5$.

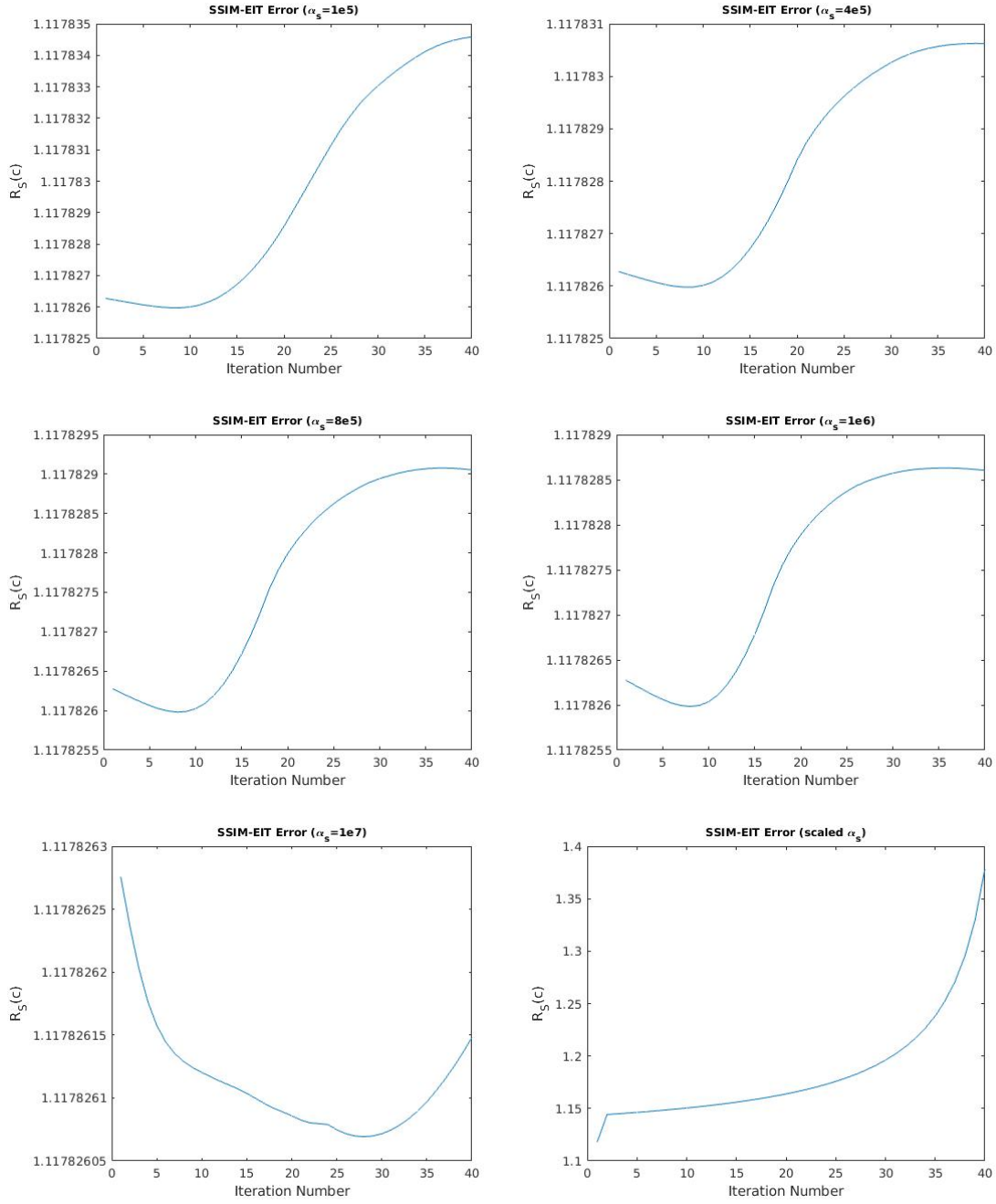


Figure 3.44: Values of the SSIM regularization term $R_S(c)$ for six different regularization parameters α_s and $\alpha_t = 10^5$.

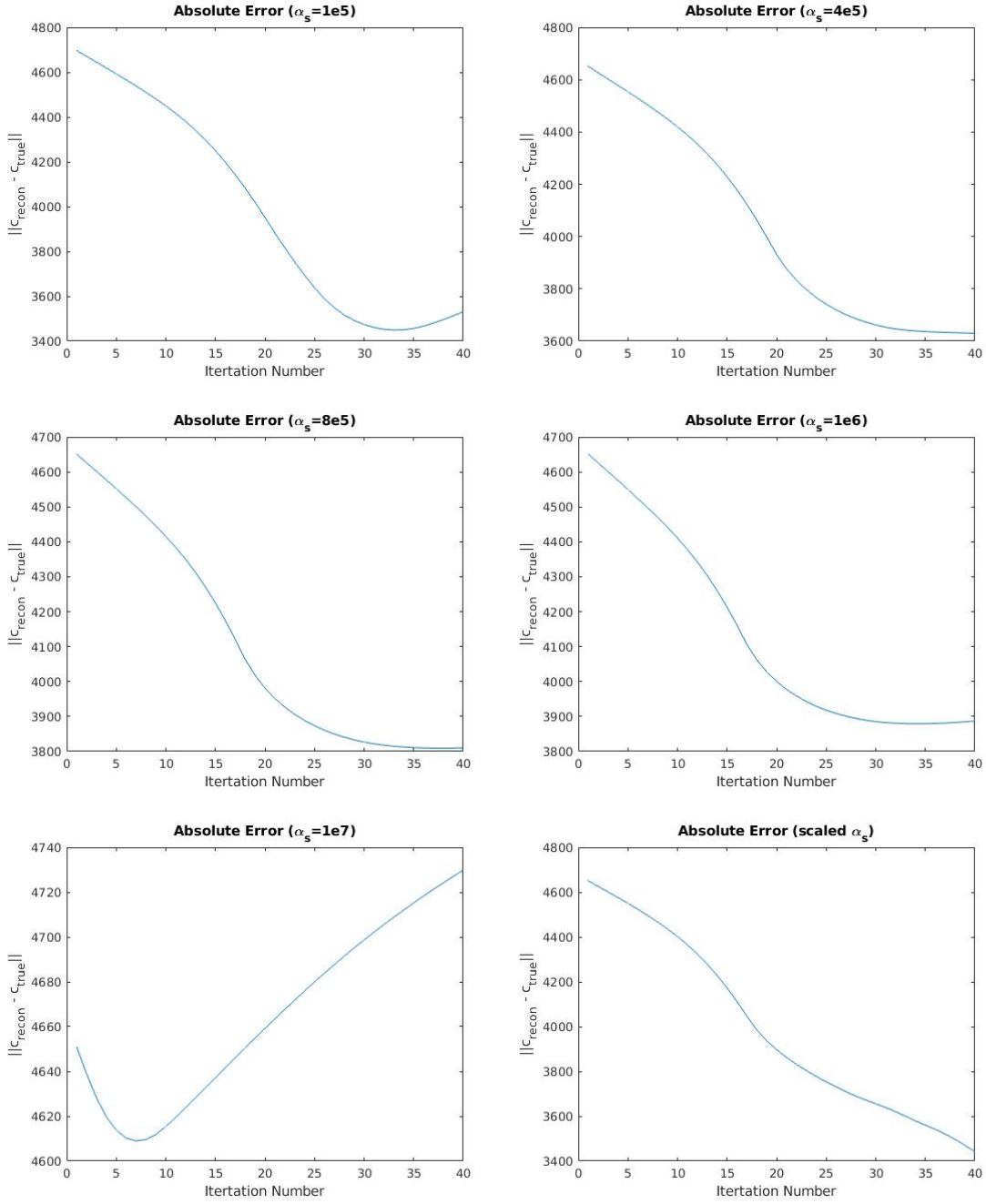


Figure 3.45: The error from ground truth $\|c_{\text{recon}} - c_{\text{true}}\|_{L^2(\Omega)}$ for six different SSIM-EIT regularization parameters α_s and $\alpha_t = 10^5$.

Chapter 4

Bayesian Full Waveform Inversion

We now explore a different approach to FWI and to inverse problems in general. In order to understand how we can use this new perspective to give uncertainty quantification in addition to reconstructions, we will first give some background information on the Bayesian approach to inverse problems.

4.1 Bayesian Inverse Problems

In the traditional approach to inverse problems, one seeks to reconstruct a deterministic parameter of a partial differential equation, ordinary differential equation, stochastic differential equation, or related model. An alternative approach to solving inverse problems is to consider the parameter we wish to reconstruct as non-deterministic, or stochastic. In this case, the solution of the inverse problem is not a deterministic constant or function but rather a probability distribution describing the parameter. Applying this ideology to USCT, this means that we no longer want to reconstruct a single sound speed map c, but instead a probability density function $f(c)$ describing how c is distributed. We can then obtain not only point estimations of the sound speed map but also quantify uncertainty for those point estimations. This technique is called the stochastic or Bayesian approach to inverse problems. While this method is often more computationally expensive and time consuming, we are rewarded with more information concerning our reconstruction.

In addition to providing more knowledge regarding our reconstruction, there is a school of thought which posits that the Bayesian approach to inverse problems is the philosophically “correct" way to approach the problem. This mindset is supported by the idea that randomness can never be removed from a problem. There is stochasticity introduced in the measurement devices used in any data collection and there is also randomness introduced through the modeling of the problem itself.

Because of this, one can never hope to achieve a perfectly accurate point reconstruction. This suggests that quantifying uncertainty regarding a reconstruction will always be necessary. While we do not necessarily subscribe to this argument, we take a Bayesian approach to solving the problem of FWI in an effort to learn more about the types of uncertainty estimations we could produce in the setting of USCT.

4.1.1 Bayes' Theorem

The workhorse in the Bayesian approach to inverse problems is Bayes' Theorem, which states that the posterior probability distribution of a parameter c (we choose c to be consistent with our sound speed map, but the theorem is true for any parameter) given data p_{meas} (again chosen to be consistent with our pressure data $p_{\text{meas}}(\boldsymbol{x}_m, t)$) is proportional to the product of the prior probability distribution of c with the likelihood function of p_{meas} given c. That is,

$$f_{\text{post}}(c|p_{\text{meas}}) \propto f_{\text{prior}}(c) f_{\text{like}}(p_{\text{meas}}|c) \tag{4.1}$$

where $f(*)$ denotes a probability density function (PDF). The formal definition of a PDF and much more on the measure theoretic description of a PDF can be found in [93]. While the posterior distribution $f_{\text{post}}(c|p_{\text{meas}})$ and the prior distribution $f_{\text{prior}}(c)$ are both true PDFs, the likelihood function $f_{\text{like}}(p_{\text{meas}}|c)$ is not a PDF because it may not integrate to 1 over the parameter space. It does, however, have the property of always being non-negative. Although the likelihood function is not a probability distribution, we follow the literature and express it using the notation $f(\cdot)$ which is typically reserved for true probability distributions.

Note that equation (4.1) is written as a proportionality instead of equality. In order to achieve equality we must include a term called the Bayesian evidence [94], which is a constant integral over parameter space. With equality, Bayes' Theorem takes the following form:

$$f_{\text{post}}(c|p_{\text{meas}}) = \frac{f_{\text{prior}}(c) f_{\text{like}}(p_{\text{meas}}|c)}{\int f_{\text{prior}}(c) f_{\text{like}}(p_{\text{meas}}|c) dc}. \tag{4.2}$$

In most practical applications the parameter space is large or otherwise impossible to integrate over given $f_{\text{prior}}(c)f_{\text{like}}(p_{\text{meas}}|c)$, meaning the calculation of the integral in equation (4.2) is infeasible. For this reason, we typically work with Bayes' Theorem as written in (4.1). We will see in section 4.1.2 that having Bayes' Theorem in proportionality is not a hindrance to us so long as we can evaluate $f_{\text{prior}}(c)f_{\text{like}}(p_{\text{meas}}|c)$.

The task of finding the the posterior probability distribution for c relies on evaluating the likelihood function and the prior probability distribution for c. We must thus have a way of determining these two functions. There is extensive literature on how to properly define the likelihood and prior distributions [95–97]. Typically, the likelihood function should be data driven, meaning there is little information in the likelihood function that the data does not supply. This is contrasted with the prior distribution, which should typically encode all information known about the parameter c prior to solving the inverse problem. If we know a great deal about the parameter before beginning our inversion, we can put that knowledge into the prior distribution in some way. This is known as choosing an informative prior. If we do not know much about the parameter c before beginning the inversion, we can inscribe this lack of knowledge into the prior by making it uninformative. The balancing act of choosing a prior distribution with the proper amount of information while still letting data play a role in the reconstruction is not straightforward and there is ample research space to explore how different prior distributions perform in the solution of the inverse problem.

We define the likelihood function as

$$f_{\text{like}}(p_{\text{meas}}|c) \propto \exp\left(-\frac{\lambda}{2}\sum_{l=1}^{K}\sum_{k=1}^{K}\|F_{l,k}(c) - p_{\text{meas,l}}(\boldsymbol{x}_k, t)\|^2_{L^2([0,T])}\right) \tag{4.3}$$

where λ is called the precision and gives a measurement of the uncertainty in the likelihood. A larger value of λ represents great certainty in the likelihood (which corresponds to great certainty in both the collected data and simulated pressures) while a small value of λ corresponds to low confidence in the likelihood. Thus the precision is equivalent to the inverse of the variance both in definition and intuition. We will explain the selection of λ in section 4.4.

Note that this likelihood function is data driven, meaning the closer we are to a sound speed c which produces simulated pressures on the transducers close to the measured pressures $p_{\text{meas,l}}(\boldsymbol{x}_k, t)$, the larger the value of the likelihood function. One could also note that $f_{\text{like}}(p_{\text{meas}}|c) \propto \exp(-\lambda E(c))$ with $E(c)$ defined in (3.27). This is intentional, as we would like to be able to use some of the techniques for FWI presented in chapter 3 within the Bayesian framework.

As noted above, picking a prior distribution is a somewhat subjective undertaking and should be handled with care. A reasonable choice for the prior distribution could be

$$f_{\text{prior}}(c) \propto \exp\left(-\frac{\alpha}{2}\left\|c - c_0\right\|^2_{L^2(\Omega)}\right) \tag{4.4}$$

where $c_0 = c_0(\boldsymbol{x})$ is a homogeneous sound speed and α is the precision in the prior distribution. This choice of prior is a Gaussian centered on the sound speed c_0 with variance equal to $1/\alpha$. Thus, if α is large then we have small variance around the sound speed c_0 and therefore great certainty that the random variable c does not differ much from c_0. A small value of α represents less certainty in c being similar to c_0.

The choice of prior is dependent on the problem and what type of restrictions we want to impose on the sound speed. An alternative prior could be obtained by replacing the L^2 norm of $c - c_0$ in (4.4) with the L^1 norm of ∇c, giving

$$f_{\text{prior}}(c) \propto \exp\left(-\alpha \int_\Omega \left(|\nabla c|^2 + \epsilon^2\right)^{1/2} d\boldsymbol{x}\right) \tag{4.5}$$

This is perfectly valid, and tends to lead to reconstructions with sharper edges as opposed to the smooth reconstructions that result from using the L^2 norm. Experimenting with the norm in the posterior and prior distribution is a worthwhile endeavor, and is similar to the task of picking a regularization term in the deterministic setting. However, the implementation of the prior in the Bayesian structure typically allows for more control and flexibility as compared to the inclusion of known information in the regularization term. We hope to explore more prior distributions than only the ones presented in (4.4) and (4.5) in the future.

Using (4.1), (4.3), and (4.5) we can write the posterior distribution as

$$f_{\text{post}}(c|p_{\text{meas}}) \propto \exp\Bigg(-\frac{\lambda}{2}\sum_{l=1}^{K}\sum_{k=1}^{K}\|F_{l,k}(c) - p_{\text{meas,l}}(\boldsymbol{x}_k,t)\|^2_{L^2([0,T])} - \alpha\int_\Omega \left(|\nabla c|^2+\epsilon^2\right)^{1/2} d\boldsymbol{x}\Bigg). \tag{4.6}$$

If we were to find the minimum of the negative logarithm of the expression in (4.6), we would get

$$c = \text{argmin}_{\tilde{c}}\left\{\frac{\lambda}{2}\sum_{l=1}^{K}\sum_{k=1}^{K}\|F_{l,k}(\tilde{c}(\boldsymbol{x})) - p_{\text{meas,l}}(\boldsymbol{x}_k,t)\|^2_{L^2([0,T])} + \alpha\int_\Omega \left(|\nabla \tilde{c}|^2+\epsilon^2\right)^{1/2} d\boldsymbol{x}\right\}.$$

Attentive readers will see that if $\lambda = 1$, this is the exact minimization problem that characterized the TV regularized inverse problem in section 3.1. The minimum of the negative logarithm of the posterior distribution is called the maximum a posteriori (MAP) estimator, and is completely equivalent to the solution of the inverse problem in the deterministic setting. However, the formulation of the problem in the Bayesian framework allows us to obtain more information about our reconstruction in the form of alternative point estimations and error quantification. For example, if we knew the posterior distribution exactly, we could use the posterior mean as a point estimation and calculate the posterior variance as a measurement of the uncertainty of that estimate. We could also generate 95% confidence intervals about that mean as an additional indicator of reliability.

In order to fully realize this error quantification we must have a good understanding of the posterior distribution $f_{\text{post}}(c|p_{\text{meas}})$. For any type of realistic model, we cannot hope to have an explicit form for this posterior distribution. That is certainly the case in (4.6), as there is no explicit way to express $F_{l,k}(c)$. We thus need a way to computationally learn about prominent features of the posterior distribution including (potentially) the mean, variance, mode, confidence intervals, or any other desired statistic. The tool we will use for this is a Markov Chain Monte Carlo (MCMC) method.

4.1.2 Markov Chain Monte Carlo Methods and the Metropolis-Hastings Algorithm

Our aim is to learn as much as possible about the posterior distribution $f_{\text{post}}(c|p_{\text{meas}})$. To accomplish this, we first need to define the two components of an MCMC method. In the following we refer to a random variable by a capital letter C and a realization of that random variable by the lower case c.

Definition 2. Markov Chain

A *Discrete Markov chain* is a sequence of random variables $C_1, C_2, ...$ satisfying the Markov property:

$$\mathbb{E}(C_n|\sigma(C_{n-1}, C_{n-2}, ..., C_1)) = \mathbb{E}(C_n|\sigma(C_{n-1})) \tag{4.7}$$

for all $n \in \mathbb{N}$ where $\sigma(C)$ is the sigma algebra generated by the random variable C.

In words, a Markov chain is a sequence of random variables such that knowing only the previous step of the chain is equivalent to knowing the entire history of the chain. A Monte Carlo method is simply a technique which involves using sampling to compute a numerical result, typically an integral. Thus an MCMC method is one which uses repeated sampling to compute numerical results in which the samples form a Markov chain. We ensure that the samples form a Markov chain because Markov chains have well-known convergence properties. Additionally, the ability to generate samples using only the previous sample often lends itself to computationally efficient algorithms. Much more can be found on the convergence properties of MCMC methods in the following sources: [98–103].

In the case of USCT, we would like to form a Markov Chain in which the samples drawn represent sound speeds, such that the samples converge in distribution to the posterior distribution $f_{\text{post}}(c|p_{\text{meas}})$. We note that convergence in distribution is a relatively weak form of convergence for random variables (as compared to convergence in probability), but convergence in distribution is

sufficient for analyzing the posterior distribution. The most widely used method for building such a Markov Chain is the Metropolis-Hastings (MH) algorithm.

The MH algorithm has been used widely for many years, so there is ample reading material to be found on the subject [95, 104, 105], but we will provide a short summary. The MH algorithm works by starting at an initial point $c_{\text{init}} \in S$, where S is a state space (for us, $\mathbb{R}^{Nx \cdot Ny}$ after discretization), and then generating a new proposal point c_1 using what is called a transition kernel. We denote this transition kernel by $\mathcal{Q}(c_{n+1}|c_n)$ and define it as follows.

Definition 3. Markov Transition Kernel

Let B be a measurable set in S, then $\mathcal{Q}(c, B) = \mathbb{P}\{C_n \in B | C_{n-1} = c\}$ is a *Markov transition kernel* if

- for each fixed $c \in S$, the function $B \mapsto \mathcal{Q}(c, B)$ is a probability measure, and
- for each fixed B, the function $c \mapsto \mathcal{Q}(c, B)$ is a measurable function.

Formal definitions of a measurable set, probability measure, and measurable function are given in [93]. However, for the purpose of intuition, we can think of the transition kernel $\mathcal{Q}$ as being a function which describes the probability of C_n taking a certain value given that we know the realization of the random variable C_{n-1}. If we were to generate a sequence of points using $\mathcal{Q}(c, B)$, this kernel ensures that the sequence we generate forms a Markov chain. Thus the first step in deriving the MH algorithm is determining how to set this Markov transition kernel.

There are two conditions that we would like our Markov chain to satisfy in order to ensure that the chain uniquely converges to $f_{\text{post}}(c|p_{\text{meas}})$. The first is ergodicity, a formal definition and more thorough description of which can be found in [106, 107]. Ergodicity will not ensure that our Markov chain converges but it will ensure that if it does converge then the distribution it converges to (the stationary distribution) is unique. Intuitively, a chain being ergodic means the chain has a nonzero probability of hitting every point in the support of f_{post} and a nonzero probability of returning to any point already visited.

These are desirable properties to have, since if we were unable to realize every point in the support of f_{post} we could not be sure that we did not miss a prominent feature of our distribution. Similarly, if we could not return to a point already visited we are unlikely to correctly identify regions of high probability.

The second condition we require, *detailed balance*, ensures that our Markov chain converges in distribution to a stationary distribution. This is again advantageous, as even an ergodic chain exploring the wrong probability distribution is useless. The detailed balance condition for the Markov kernel $\mathcal{Q}$ and the posterior distribution $f_{\text{post}}(c_n|p_{\text{meas}})$ (in the context of the MH algorithm the posterior distribution would be called the target distribution) is written as

$$Q(c_{n+1}|c_n)f_{\text{post}}(c_n|p_{\text{meas}}) = Q(c_n|c_{n+1})f_{\text{post}}(c_{n+1}|p_{\text{meas}}). \tag{4.8}$$

It can be shown [106] that under suitable conditions, if a sequence $\{c_n\}_{n\in\mathbb{N}}$ satisfies detailed balance, it converges in distribution to the target distribution. Since detailed balance is sufficient to ensure our Markov chain converges to $f_{\text{post}}(c_n|p_{\text{meas}})$, that is the starting point for the MH algorithm.

To better understand the MH algorithm, let's assume that our transition kernel can be written as $\mathcal{Q}(c_{n+1}|c_n) = r(c_{n+1}, c_n)g(c_{n+1}|c_n)$ where $r(c_{n+1}, c_n)$ is known as the acceptance probability (or acceptance ratio) and $g(c_{n+1}|c_n)$ is known as the proposal distribution. With this, we write equation (4.8) as

$$
\begin{aligned}
Q(c_{n+1}|c_n) f_{\text{post}}(c_n|p_{\text{meas}}) &= Q(c_n|c_{n+1}) f_{\text{post}}(c_{n+1}|p_{\text{meas}}) \\
\Rightarrow \frac{Q(c_{n+1}|c_n)}{Q(c_n|c_{n+1})} &= \frac{f_{\text{post}}(c_{n+1}|p_{\text{meas}})}{f_{\text{post}}(c_n|p_{\text{meas}})} \\
\Rightarrow \frac{r(c_{n+1},c_n) g(c_{n+1}|c_n)}{r(c_n,c_{n+1}) g(c_n|c_{n+1})} &= \frac{f_{\text{post}}(c_{n+1}|p_{\text{meas}})}{f_{\text{post}}(c_n|p_{\text{meas}})} \\
\Rightarrow \frac{r(c_{n+1},c_n)}{r(c_n,c_{n+1})} &= \frac{f_{\text{post}}(c_{n+1}|p_{\text{meas}}) g(c_n|c_{n+1})}{f_{\text{post}}(c_n|p_{\text{meas}}) g(c_{n+1}|c_n)}.
\end{aligned}
$$

At this point, the goal is to choose an acceptance probability such that (4.8) is guaranteed to hold. Note that the choice

$$
r(c_{n+1},c_n) = \min\left(1, \frac{f_{\text{post}}(c_{n+1}|p_{\text{meas}}) g(c_n|c_{n+1})}{f_{\text{post}}(c_n|p_{\text{meas}}) g(c_{n+1}|c_n)}\right) \tag{4.9}
$$

guarantees that (4.8) holds. Equation (4.9) is known as the Metropolis acceptance probability.

The idea behind the algorithm is now as follows: pick a proposal distribution $g(c_{n+1}|c_n)$ and use it to generate a candidate or proposal point c^*. Then, pass that proposal point through the Metropolis acceptance probability in (4.9). This will guarantee that at every point in the Markov chain the condition of detailed balance is satisfied and thus produces a sequence of points $\{c_n\}$ which converges in distribution to $f_{\text{post}}(c_n|p_{\text{meas}})$. These steps are laid out in the following algorithm.

Algorithm 2. (MH Algorithm)

Given the current step c_n and proposal distribution $g(c_{n+1}|c_n)$, repeat the following steps:

1. Draw candidate point $c^* \sim g(c^*|c_n)$.
2. Calculate $r(c^*,c_n) = \min\left(1, \frac{f_{\text{post}}(c^*|p_{\text{meas}})}{f_{\text{post}}(c_n|p_{\text{meas}})} \frac{g(c_n|c^*)}{g(c^*|c_n)}\right)$.
3. Draw $u \sim U([0,1])$. If $u \leq r(c^*,c_n)$, let $c_{n+1} = c^*$. Else, let $c_{n+1} = c_n$.

The last step of the above algorithm is known as the accept/reject phase. If the candidate point c^* is in a region of higher probability as compared to c_n, the point will be accepted as the next point in the chain. However, if c^* is in a region of slightly lower probability as compared to c_n, it may still be accepted by the algorithm due to the draw $u \sim U([0,1])$. This is an important step. Without it, our algorithm would proceed to the region of highest probability space near the starting point and do nothing else. With it, we allow the chain to fully explore the parameter space allowing for more accurate and complete error analysis. It can be shown [98] that under suitable conditions on $g(c_{n+1}|c_n)$ and $f_{\text{post}}(c|p_{\text{meas}})$, Algorithm 2 produces an ergodic chain which converges in distribution to $f_{\text{post}}(c|p_{\text{meas}})$.

With the algorithm laid out, the most important step left to consider is the choice of $g(c_{n+1}|c_n)$. There has been extensive research on the best choice for the proposal distribution [108–110], and in the next section we will outline our choice of proposal. However, it will benefit us to describe the algorithm when the proposal distribution is set to

$$g(c_{n+1}|c_n) = \mathcal{N}(c_{n+1}|\mu = c_n, \sigma^2),$$

where σ is a chosen parameter. With this choice, Algorithm 2 becomes the well-known random-walk MH algorithm. Note that if σ is small, then nearly every proposal will be accepted since (for a smooth target distribution) the new proposal will not differ much in probability compared to the previous point. However, a small value of σ will cause the algorithm to take many steps in order to reach convergence. If σ is large, then the proposal will be far from the previous value, meaning there is the possibility of exploring the probability distribution very quickly. However, many more proposals are likely to be rejected if σ is large since there is no guarantee our proposal will be in a higher probability region. There is a good deal of work involved in determining the correct value for the parameter σ, and we leave a discussion of these issues to the following: [111, 112].

The random-walk MH algorithm is a beautifully simple algorithm, as the proposal distribution is cheap and easy to sample from. However, it breaks down very quickly in high dimensional

parameter space [113, 114] since the odds of randomly selecting points in regions of higher probability decrease exponentially as the dimension increases. Since our parameter space will typically be of quite large dimension after discretization, we must consider an alternative way of proposing points c^*.

4.2 The Metropolis-Adjusted Langevin (MALA) Algorithm

In order to more efficiently compute proposals in high-dimensional parameter space, we will employ an algorithm which uses information about the geometry of our posterior distribution. There are multiple MCMC methods that endeavor to do this (i.e., the Hamiltonian Monte Carlo algorithm [113, 114]), but we make use of the Metropolis-Adjusted Langevin (MALA) Algorithm [115, 116]. MALA uses a discretization of a Langevin diffusion to generate proposals and then passes those proposals through the same accept/reject mechanism found in the MH algorithm. We begin our exploration of the MALA algorithm by more carefully analyzing the Langevin diffusion. This requires some definitions.

Definition 4. A *Wiener process* (or *Brownian motion*) is a random variable $W(t)$, depending continuously on $t \in [0, T]$, satisfying the following properties:

- $W(0) = 0$,
- for $0 \leq s < t \leq T$, $W(t) - W(s) \sim \sqrt{t-s}\,\mathcal{N}(0,1)$, where $\sim$ means "distributed as" and $\mathcal{N}(0,1)$ is a Gaussian distribution with mean 0 and variance 1,
- for $0 \leq s < t < u < v \leq T$, $W(t) - W(s)$ and $W(v) - W(u)$ are independent.

Definition 5. A *diffusion* is a continuous time stochastic process $X_t,\ t \geq 0$ which can be written as a solution of a stochastic differential equation (SDE) of the form

$$dX_t = \boldsymbol{b}(X_t)dt + \sigma(X_t)dW_t,$$

where $\boldsymbol{b}$ is known as a drift vector, σ is a volatility matrix, and W_t is a Wiener process.

The Langevin diffusion is the solution to the SDE

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)dt + dW_t,\ X_0 = x_0, \tag{4.10}$$

where $\pi(\cdot)$ is any probability distribution. We cannot in general solve the Langevin diffusion explicitly, so we would like to obtain a numerical solution to (4.10). We can obtain a realization of the Langevin diffusion by using the Euler-Maruyama numerical method. Given an initial value $X_0 = x_0$ and a step-size h, Euler-Maruyama simulates the Langevin diffusion with the procedure

$$x_{n+1} = x_n + \frac{h}{2}\nabla \log(\pi(x_n)) + \epsilon, \tag{4.11}$$

where $\epsilon \sim \mathcal{N}_d(\mathbf{0}, h\boldsymbol{I}_d)$ and $\boldsymbol{I}_d$ is the identity matrix in $\mathbb{R}^d$. Those familiar with the well-known Euler method for traditional ordinary differential equations will see that Euler-Maruyama is simply the stochastic analog of this method.

We would like to generate proposals c^* in our state space $S = L^2(\Omega)$ which converge to the posterior distribution $f_{\text{post}}(c|p_{\text{meas}})$. Thus in (4.10), we let the probability distribution $\pi(\cdot)$ be the posterior distribution $f_{\text{post}}(c|p_{\text{meas}})$ defined in (4.6). With these values, a realization of the solution to the Langevin diffusion in (4.11) using Euler-Maruyama is

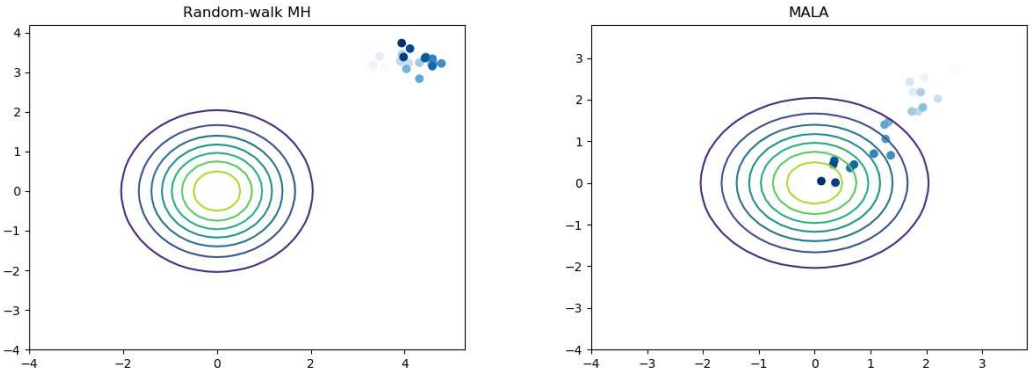
$$c^* = c_n + \frac{h}{2}\nabla_c \log(f_{\text{post}}(c|p_{\text{meas}})) + \epsilon. \tag{4.12}$$

We use the notation c^* in (4.12) to suggest that the result of the Euler-Maruyama method will be used as a proposal point within the MALA algorithm.

With (4.12), we see an intuitive interpretation of the proposal generated by the Langevin diffusion. Starting at the point c_n, we move in the direction of the negative gradient of $E(c_n)$, with a slight perturbation ϵ. This means that with an appropriate step length each proposal moves to a value with a lower output of the error $E(\cdot)$, meaning a region of higher probability in $f_{\text{post}}(c|p_{\text{meas}})$. The perturbation ϵ allows the algorithm to stray a bit and occasionally propose points in a region of lower probability. As we note in section 4.1.2, proposing points in regions of lower probability intermittently is necessary in order to full explore the probability space.

In addition to providing intuition, thinking of the Langevin diffusion as stepping in the direction of the negative derivative of $E(c)$ also offers computational benefits. This is because we have already computed the exact derivative of $E(c)$ using the adjoint method in equation (3.47). Therefore the adjoint method used to calculate a functional derivative in that section is completely applicable to finding the derivative of the negative gradient of the log of the posterior distribution.

We highlight the difference in the MALA proposal mechanism as compared with the random-walk MH algorithm in figure 4.1. Figure 4.1 displays a comparison between proposal points for the random-walk MH algorithm and the MALA algorithm for a 2D Gaussian probability with mean 0 and variance 1. Note that this is not the actual Markov chain produced by these algorithms, but only the proposal points. Both algorithms were started at a value of $x = 3,\ y = 3$, and 20 proposals were generated. For each algorithm, the random value ϵ was drawn from the distribution $\mathcal{N}(\mu = 0, \sigma^2 = 0.3)$ and the step-length h for the MALA algorithm was set to 0.2. The proposals for the random-walk MH algorithm are grouped around the starting value of $(x, y) = (3, 3)$, and even if we increased the value of σ^2 we could not expect to achieve more informed proposals. We note that adding an accept/reject phase would help the algorithm move towards the origin (the area of highest probability), but this would incur a computational cost due to the discarding of proposals. As seen in figure 4.1b, the proposals for the MALA algorithm reach the high probability region in only 20 steps. This is because the MALA algorithm uses information about the probability



(a) 20 proposals for the random-walk MH algorithm. **(b)** 20 proposals for the MALA algorithm.

Figure 4.1: A comparison of proposals for the random-walk MH algorithm and MALA algorithm on a simple 2D probability distribution.

space to inform its proposals. We can already see this advantage in the 2D example presented, and for higher dimensional spaces (such as in the problem of USCT) this advantage is even more pronounced [115].

With the proposal mechanism in place, we must determine how to express the proposal distribution $g(c^*|c_n)$ in the Metropolis acceptance probability $r(c^*, c_n)$. This proposal is not symmetric since there is a much higher probability of moving to c^* given c_n as opposed to moving from c^* to c_n. We therefore must set $g(c^*|c_n)$ to be

$$g(c^*|c_n) = \mathcal{N}(c_n + \frac{h}{2}\nabla_c \log(f_{\text{post}}(c_n|p_{\text{meas}})), h). \tag{4.13}$$

This definition for $g(c^*|c_n)$ matches that given in the literature [117] and ensures that the resulting sequence of points forms a Markov chain which satisfies the detailed balance condition.

We are now ready to present the complete MALA algorithm.

Algorithm 3. The Metropolis Adjusted Langevin Algorithm

Given the current step c_n, step length parameter h, and parameter ϵ, repeat the following steps:

1. Draw candidate point $c^* \sim \mathcal{N}(c_n + \frac{h}{2}\nabla_c \log(f_{\text{post}}(c_n|p_{\text{meas}})), \epsilon)$.
2. Calculate $r(c^*, c_n) = \min\left(1, \frac{f_{\text{post}}(c^*|p_{\text{meas}})}{f_{\text{post}}(c_n|p_{\text{meas}})} \frac{g(c_n|c^*)}{g(c^*|c_n)}\right)$ with $g(\cdot)$ given by (4.13).
3. Draw $u \sim \mathcal{U}([0,1])$. If $u \leq r(c^*, c_n)$, let $c_{n+1} = c^*$. Else, let $c_{n+1} = c_n$.

The MALA algorithm has the added advantage of well-informed proposals while maintaining the perks of the MH algorithm. Step 2 of Algorithm 3 still only requires evaluation of the posterior distribution up to a constant. However, Algorithm 3 has a more sophisticated proposal mechanism arising from the evaluation of the gradient of the log of the posterior distribution.

After probing the posterior distribution using the MALA algorithm, we would like to obtain a point estimation of the sound speed map and uncertainty information regarding that estimation. With the Markov chain in hand this is straightforward, as we can simply calculate the posterior mean, variance, maximum posteriori (MAP) estimator, or any other desired statistic using the samples from the chain.

With all necessary background established, we are now ready to present our full, novel algorithm for USCT in the Bayesian framework.

4.3 A MALA FWI Algorithm for USCT

Algorithm 4. Given the current step c_n, step length parameter h, and parameter ϵ, repeat the following steps:

1. Compute the derivative $\nabla_c \log(f_{\text{post}}(c_n|p_{\text{meas}}))$ using (3.47).
2. Propose a sound speed c^* using (4.12).
3. Compute $r(c^*, c_n) = \min\left(1, \frac{f_{\text{post}}(c^*|p_{\text{meas}})}{f_{\text{post}}(c_n|p_{\text{meas}})} \frac{g(c_n|c^*)}{g(c^*|c_n)}\right)$ with $g(\cdot)$ given by (4.13).

4. Draw $u \sim U([0,1])$. If $u \leq r(c^*, c_n)$, let $c_{n+1} = c^*$. Else, let $c_{n+1} = c_n$.

The above is a succinct summary of the algorithm, but there are many variations that could be adopted. In section 4.5 we will explore some of these variations that allow us to obtain samples from the posterior distribution with less computational effort. Before examining these variations, we specify how relevant parameters in Algorithm 4 are chosen.

4.4 Parameter Selection

While the majority of tools necessary for the algorithm have been described in the previous sections, there is still the matter of selecting the parameters ϵ, h, λ, and α that appear (either explicitly or implicitly) in Algorithm 4. Selecting appropriate values for these parameters is necessary, as poor selections will lead to substandard results. If the parameter ϵ is too large, then the proposal c^* will be over-influenced by random noise and thus lose the information provided by the derivative $\nabla_c \log(f_{\text{post}}(c_n|p_{\text{meas}}))$. If ϵ is too small, then the algorithm will simply move toward a high probability region and fail to properly capture the structure of the posterior distribution. There are equally disastrous results if the step length parameter h and precision parameters λ and α are not set to sensible values. We therefore need a way of selecting our parameters in a manner which does not add excessive computational burden to the algorithm.

Since we would like ϵ to be somewhat small compared to the derivative $\nabla_c \log(f_{\text{post}}(c_n|p_{\text{meas}}))$, we will let

$$\epsilon \sim \mathcal{N}(\mathbf{0}, 0.01\mathbb{I}_d) \tag{4.14}$$

That is, ϵ is drawn from a d-dimensional Gaussian distribution centered at the origin with variance equal to 0.01 m/s. At first glance, this variance seems too small compared to the derivative, but theoretically once the algorithm converges to the posterior distribution the value of the derivative should start to shrink substantially. This should allow for quicker convergence to the posterior distribution followed by random samples from that distribution.

Step 2 of Algorithm 4 involves selecting the step length parameter h. Since we have described the proposal mechanism as being a step in the direction of the negative gradient of $E(c)$ with a bit of randomness added, it makes sense to let this step length be equal to the step length in equation (3.50) which was used successfully in chapter 2. In this case, however, we let $s_{\max}$ be an adaptive term which begins at 3 and shrinks successively over the first 100 iterations to a value of 0.25, where it settles for the remainder of the algorithm. This choice was made with the hope that the Markov chains would reach a region of higher probability space quickly and then sample more conservatively in that area.

Choosing the precision values λ and α is an interesting and delicate undertaking. We have already said that these precision values represent our certainty in the likelihood and prior distribution, respectively. It can be difficult, however, to quantify our certainty in the likelihood and prior without introducing biasing into the results. If we were seeking only the MAP estimator, we could just set $\lambda = 1$ and treat α as a regularization parameter. This would allow us to use the L-curve method or Morozov discrepancy principle [118] or any other regularization parameter selection technique. Yet, our goal in moving to the Bayesian framework is to view the problem through the lens of random variables. With this in mind, we set the precision values by attempting to describe the error in measurements and/or modeling through λ and our certainty (or lack thereof) of the prior distribution through α. This is a bit of a subjective approach, and is the cause of some criticism in the Bayesian approach to inverse problems. However, it allows more flexibility in choosing parameters and allows the user to input more information into the problem as a whole. A discussion of techniques for finding precisions (or variances) in the Bayesian approach to inverse problems can be found in [119]. When the accuracy of the measurement device can be determined, knowledge of the measurement error can be directly included in the precision parameter λ. Since we are working with simulated data, this is a more difficult endeavor. In place of this rigorous approach to determining the precision parameters we instead consider the model error between the program which simulated the data and the one used to solve the inverse problem and use that knowledge to coarsely select λ. The prior precision α will be set by considering our confidence in

the selected prior as informed through the deterministic reconstructions in section 3.5.3. The exact values for λ and α in our reconstruction are given in section 4.6.2.

4.5 Computational Improvements

Algorithm 4 builds a Markov chain which can sample from the posterior distribution, but it does not do so in an efficient manner. In this chapter we consider computational improvements to the MALA FWI algorithm which will allow us to obtain more samples from the posterior distribution in less time.

4.5.1 Faster Derivative Calculation

In (3.47) we presented the derivative of the log of the posterior distribution with the assumption that this derivative is calculated using every transducer. As stated in section 3.3, this necessitates K solutions to the forward problem and adjoint problem for each calculation of the derivative, where K is the number of transducers. Thus, a single calculation of the derivative constitutes the bulk of the numerical cost in Algorithm 4. One way to reduce the number of forward and adjoint problems which need to be solved in the adjoint method is to reduce the number of transducers used to calculate the derivative. That is, instead of summing over K transducers in (3.47), we instead sum over a subset of the transducers. This has the effect of making the calculation of the derivative more efficient at the expense of a less accurate representation of the functional derivative $\nabla_c E(c)$. However, we can select the transducers which are used to calculate the derivative in a strategic way so that we preserve the image quality while also allowing for more samples to be drawn from the posterior distribution in a shorter period.

Let the set of indices for all transducers be called $\mathcal{K}$. Assume that we may partition $\mathcal{K}$ into J disjoint subsets $\mathcal{K}_1, \mathcal{K}_2, ..., \mathcal{K}_J$ of equal cardinality such that

$$\bigcup_{j=1}^{J} \mathcal{K}_j = \mathcal{K}.$$

We assign the physical location of the transducer $\boldsymbol{x}_k$ by giving it the coordinates $\left(r_t\cos\left(\frac{2k\pi}{K}\right), r_t\sin\left(\frac{2k\pi}{K}\right)\right)$, where r_t is the radius of the ring array. The elements of each set $\mathcal{K}_j$ are set as

$$\mathcal{K}_j = \{j + nJ,\ n = 0, 1, 2, 3, ... \mid j + nJ \leq K\}.$$

As an example, if $K = 120$ and $J = 4$, then $\mathcal{K}_1 = \{1, 5, 9, 13, ..., 117\}$, $\mathcal{K}_2 = \{2, 6, 10, 14, ..., 118\}$, $\mathcal{K}_3 = \{3, 7, 11, 15, ..., 119\}$, and $\mathcal{K}_4 = \{4, 8, 12, 16, ..., 120\}$. If we were to plot the physical locations of the transducers corresponding to each $\mathcal{K}_j$ in this example, they would appear as 30 equally spaced points on a ring array surrounding the region of interest instead of the original 120 points in $\mathcal{K}$.

Now we can revise equation (3.47) to be calculated using the transducers in one of the subsets $\mathcal{K}_j$. At iteration i of Algorithm 4, the derivative may be calculated as

$$\nabla_c E(c) = -\frac{2}{c^3} \sum_{\{l \in \mathcal{K}_{i(\text{mod } J)}\}} \frac{|\mathcal{K}|}{|\mathcal{K}_i|} \int_0^T \frac{\partial p(\boldsymbol{x}, t|\boldsymbol{x}_l)}{\partial t} \frac{\partial \tilde{p}(\boldsymbol{x}, t|\boldsymbol{x}_l)}{\partial t} dt. \tag{4.15}$$

With the new derivative calculation in (4.15), we are still ensuring that our region is illuminated from many different angles while using only $1/J$ the number of transducers compared to the calculation of the derivative in (3.47). This means that step 1 of Algorithm 4 will be computed J times more quickly, which constitutes a significant improvement.

4.5.2 Parallel Chains

In addition to the performance enhancement provided by the faster derivative calculation, we also note that Algorithm 4 lends itself to being run in parallel using multiple Markov chains. As long as each of these Markov chains converges to the posterior distribution, they will each obtain independent samples from the posterior distribution. In theory, this means that if M different Markov chains are simulated in parallel using Algorithm 4, and each one obtains N independent

samples from the posterior distribution $f_{\text{post}}(c|p_{\text{meas}})$, then we will have actually generated $M * N$ samples in the same time it would have originally taken to generate N.

While each of the chains could have the same starting sound speed c_{init} (they are guaranteed to behave differently due to the random variable ϵ in equation (4.12)), it would be more beneficial to have each chain begin at a different sound speed. This would allow each chain to take a different path through parameter space while ideally moving to the same region of higher probability, thus providing more detailed information concerning the shape of the posterior distribution. For the size of parameter space that we are dealing with (the dimension of our parameter space is on the order of 50,000), we would like to have many thousands of Markov chains with unique starting points run in parallel. In practice, at least for this author, that is a fools' wish. However, we should not discount the added savings in computational time achieved by running even a few chains in parallel.

4.6 Test Phantom

We now investigate how Algorithm 4 performs on data simulated from a simplified breast phantom. This phantom mimics the size and sound speeds of the breast phantom in section 3.5, but is in theory a bit easier to image due to the lack of the fatty layer within the boundary of the region.

4.6.1 Data Generation

Just as in sections 3.5.1 and 3.6.1, we simulated data using k-Wave. We have already described the relevant fields that need to be set in order to generate time dependent pressure waves on the transducers in k-Wave, so here we simply state all of the relevant fields while pointing out changes in the setup for this problem as compared to the configuration in section 3.5.

kgrid: The simplified breast phantom has an overall domain size of $0.1m \times 0.1m$ with a grid size of $Nx = Ny = 302$. This means our discretization is of size $dx = dy = 3.3113 \times 10^{-4}$.

medium: The mass density, nonlinear parameter BonA, power law absorption coefficient, and power law absorption exponent were set identically to the breast phantom in section 3.5.1. The sound speed parameter, however, is the primary change for the test phantom. The true sound speed map is shown in figure 4.2, and a table of sound speed values is given in table 4.1. The absence of an outer layer of tissue and complex interior structure simplifies the problem and eases the computational burden of the algorithm.

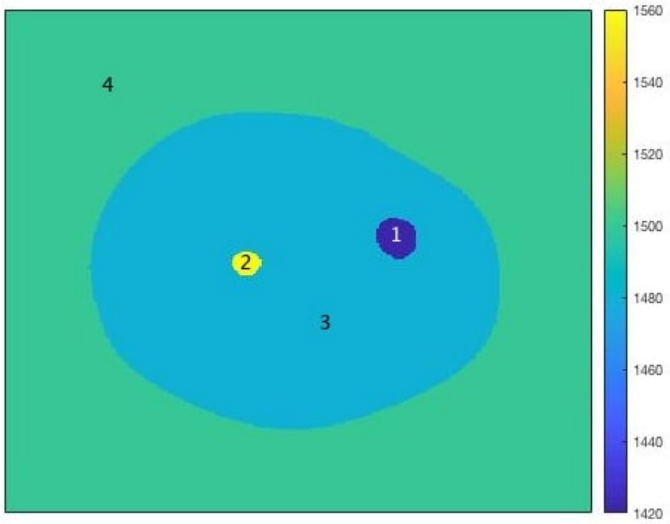


Figure 4.2: Test phantom used to generate data.

Table 4.1: Parameters of the numerical test phantom.

Structure Index	**Tissue Type**	**Sound Speed (m/s)**
1	Fatty Tissue	1420
2	Malignant Tumor	1560
3	Parenchyma	1480
4	Background	1500

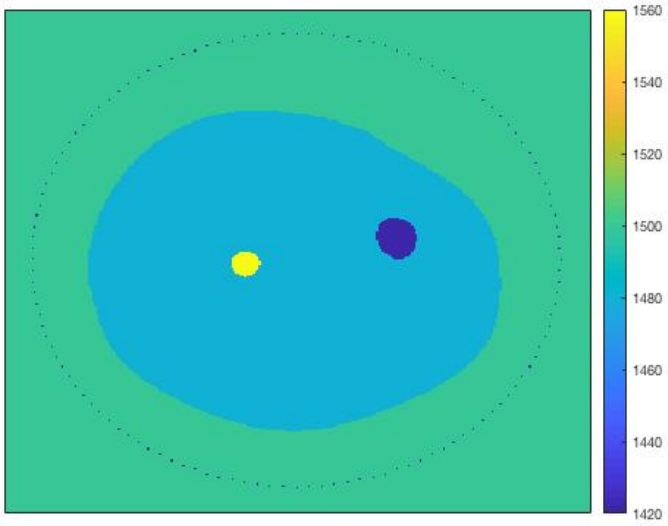


Figure 4.3: Transducer locations for K=128 surrounding the simplified breast phantom.

sensor, source and signal: The sensor, source and signal for the generation of data in this section are identical to the those used to generate data in section 3.5.1, so we do not linger on their description. The location of the transducers surrounding the phantom is shown in figure 4.3.

Each source emitted a wave packet similar to the one in figure 3.5, again with a center frequency of 1 MHz and a strength of 1000 Pa.

4.6.2 Inverse Problem Setup

In order to test Algorithm 4, Gaussian noise η was added to each time dependent signal of our data array. The noise term η was again drawn from the distribution $\mathcal{N}(\mathbf{0}, \sigma\mathbb{I}_{Nt})$ with σ equal to 0.01. The inverse problem was solved on a grid of size 300×300, ensuring that no inverse crime was committed. The data calibration step outlined in section 3.5.2 was also applied to this data set.

Six Markov chains were run in parallel, each with a slightly different starting point. These starting sound speeds were generated by beginning with a homogeneous background of sound speed 1500 m/s and then adding an inclusion of the same size and shape as the boundary of the object in figure 4.2. The sound speed of that insertion was set to 1477 m/s, 1478 m/s, 1479 m/s, 1481 m/s, 1482 m/s, and 1483 m/s (recall the true value of the background of the breast is 1480

m/s) for the six different chains in order to ensure that each chain took different paths through parameter space. This also mimics a real-life scenario in which we might roughly know the sound speed of the boundary of our region but allow for some variance in this sound speed at the start.

The derivative $\nabla_c \log(f_{\text{post}}(c_n|p_{\text{meas}}))$ was calculated using (4.15) with $J = 4$. The precision parameters λ and α were set by running multiple small reconstructions and considering the role of modeling error in addition to noise level. Since the inverse problem was run on a coarser grid than that used to generate the data, the position of the transducers with regards to the region of interest is slightly mismodeled. This, along with the noise η, implies that λ should be slightly less than 1 in order to convey our lack of certainty in the measured and simulated data. Thus some experimentation led us to set λ equal to 0.95.

The prior precision parameter α reflects our relative certainty in the edge sharpness prior given in equation (4.5). Some preliminary testing with this parameter led us to select $\alpha = 5 \cdot 10^7$. We point out that if the uncertainty in modeling and measurement devices were known more precisely, a less heuristic method could be used to set λ and α. See [119] for reference.

4.6.3 Results

The problem of determining the number of samples needed to accurately compute statistics is non-trivial. Previous work [120] has suggested that an effective sample size of roughly 10,000 is necessary for accurate estimates, but other work has suggested this number may be lower. Whatever the case, our computational and temporal resources make obtaining this number of samples unreasonable. Therefore results from the following section should be seen as preliminary, and represent an attempt to determine the viability of Algorithm 4 in a scenario in which time and computational power are abundant.

We were able to obtain 202 samples from each Markov chain, not including the first 100 iterations of each Markov chain which were thrown out as burn-in. That led to a total pool of 1212 samples, which may or may not be independent draws from the posterior distribution. With these samples, we calculated the MAP estimator, posterior mean, posterior variance, and 95 percent con-

fidence intervals. Those results are shown in figure 4.4. Note that the image of the posterior mean also includes the acceptance ratio (number of accepted proposals over number of total samples). This acceptance ratio does not take into account the first 100 iterations of the Markov chains, for which the acceptance ratio was significantly higher. These results suggest that the phantom was too simplistic, as our point estimators are nearly identical to the true sound speed and the posterior variance is quite small. However, these results also suggest that Algorithm 4 excelled at providing a detailed reconstruction of the phantom. In particular, the derivative calculation in equation (4.15) did not result in a degradation of the sound speed or any unusual artifacts.

The plots of the posterior variance and element-wise width of the 95 percent confident interval in figures 4.4c and 4.4d show that the algorithm was most uncertain about the inhomogeneity representing the fatty tissue and the outer boundary of the phantom. The uncertainty at the edge of the phantom is likely due to the prior being edge-sensitive and affecting the boundary slightly. This is valuable information that we would not get from a deterministic reconstruction. It is also evidence that our Markov chains did not converge, and suggests that we need more samples in order to achieve better point estimations.

With the samples from each Markov chain in hand, it is typically also good practice to determine if each Markov chain has converged to the target distribution. One tool for this analysis is a calculation of the Geweke p-value for each chain at a particular pixel [121]. The Geweke test begins with the hypothesis that the mean of the first ten percent of the samples is equal to the mean of the final fifty percent of the samples. A t-value can be calculated for this hypothesis and used to produce a p-value in order to help determine the viability of the null hypothesis. We selected three pixels and produced a trace plot of each Markov chain at that location. One of the pixels is located in the background of the breast, one is on the inhomogeneity of larger sound speed, and the last is on the inhomogeneity of smaller sound speed. These plots and the corresponding Geweke p-values for each are shown in figures 4.5, 4.6, and 4.7.

The Geweke p-values for these plots are generally very close to 1, which strongly suggests that the null hypothesis for the Geweke test is true. Indeed by visual inspection the values within

each of the pixels in the above images change very little throughout the chain. This is a good sign that our chain is indeed in equilibrium, and is also a sign that perhaps the variance of 0.01 within equation (4.14) was too small. More evidence that the variance was too small can be found in the similar shape of the trace plots in figures 4.5, 4.6, and 4.7. While the trace plots have slightly different scales on the y-axis, the overall shape of each Markov chain at the specified pixels is remarkably similar. This is an indication that each Markov chain took an almost identical path through parameter space, and suggests that the variance was too small and the starting sound speeds were too similar. For a more involved breast phantom, this would likely result in much worse results than those seen in figure 4.4.

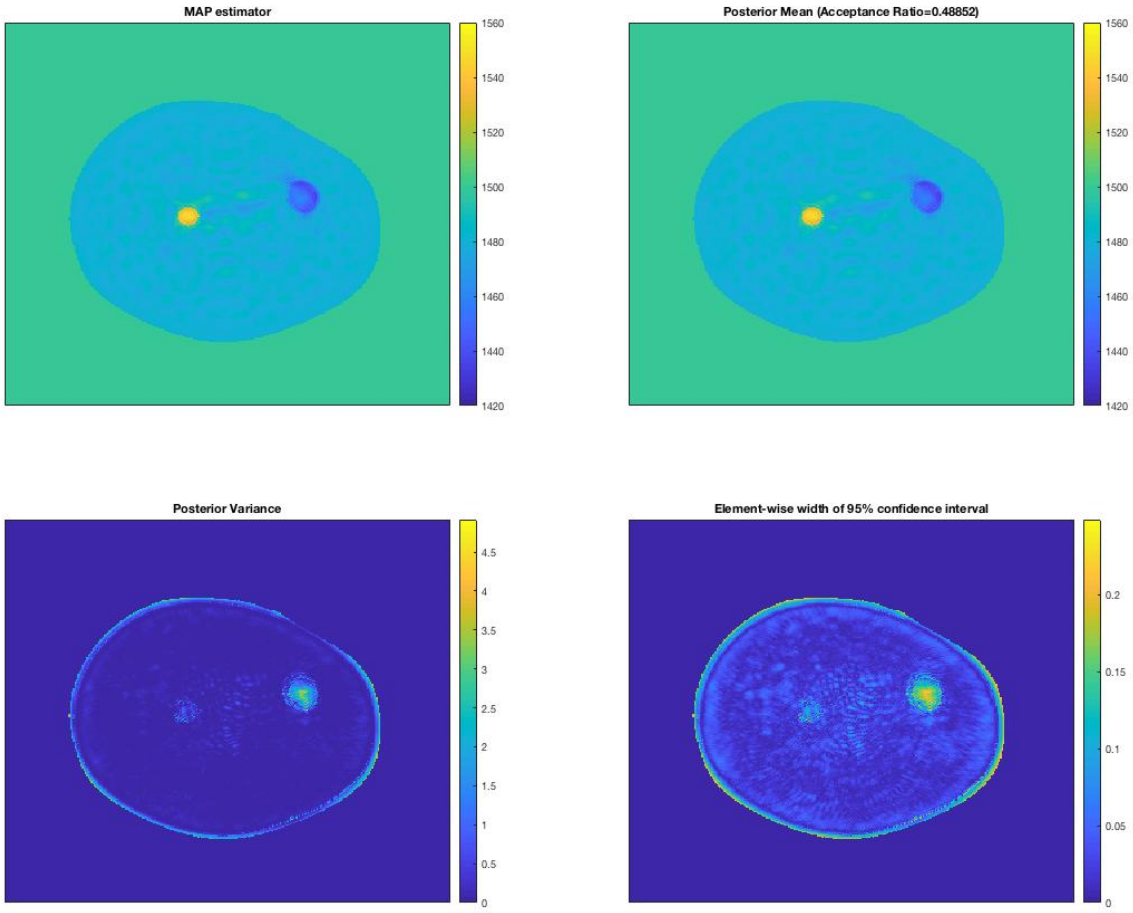


Figure 4.4: The MAP estimator, posterior mean, posterior variance, and confidence intervals from the 1212 samples drawn from the posterior distribution.

Figure 4.5: Plots of the Markov Chain at a pixel in the background of the breast.

Figure 4.6: Plots of the Markov Chain at a pixel in the inhomogeneity of larger sound speed within the breast.

Figure 4.7: Plots of the Markov Chain at a pixel in the inhomogeneity of smaller sound speed within the breast.

Chapter 5

Summary and Future Work

This work focused on full waveform inversion and its application to USCT in the field of medical imaging. After considering current approaches to the problem in chapter 2, we presented a FWI algorithm in the deterministic framework in chapter 3. Within that chapter, we investigated the role and performance of different regularization techniques and also introduced a novel SSIM-EIT regularization term which penalized sound speed maps that differed in structure from an EIT reconstruction. Chapter 3 also presented reconstructions of a lung phantom using USCT at low frequencies, which is a new application of FWI techniques. In chapter 4 we shifted mindsets and analyzed the inverse problem from the Bayesian perspective. This led to a novel MALA-FWI algorithm that makes use of FWI techniques within the Bayesian framework.

The results presented in chapters 3 and 4 show the ability of FWI to accurately reconstruct small inhomogeneities within a medium. The position and shape of the inhomogeneities in the breast phantom, lung phantom, and test phantom were identified with great accuracy, and the sound speeds of those inhomogeneities were determined with good accuracy. In addition, the SSIM-EIT regularization term led to a slight increase in the speed of convergence for the algorithm, which is beneficial for a computationally demanding algorithm such as FWI. The images produced of the lung with three different regularization terms are the first using FWI, and provide an excellent starting point for imaging a cross section of the human thorax with high resolution using USCT.

Qualitatively, the best breast phantom result from noisy data would appear to be the Tikhonov reconstruction when $\alpha = 2 \cdot 10^7$. This sound speed map identifies all of the major features of the phantom while controlling the speckling that dominates the reconstruction at $\alpha = 10^6$. Interestingly, this is not the best performing reconstruction with regards to the absolute error, as the value of $||c_{\text{recon}} - c_{\text{true}}||$ hits a minimum of around 1260 when $\alpha = 2 \cdot 10^7$ compared with roughly 1175 when $\alpha = 10^6$. This suggests that the absolute error term may not be the best indicator of a clean image since it does not take into account the speckling that occurs for low regularization

parameters. The best lung reconstruction qualitatively is a bit tougher to pick out, as many of the reconstructions do an excellent job. However, it would seem that the TV sound speed map for $\alpha = 10^7$ and the SSIM+EIT sound speed map for $\alpha = 4 \cdot 10^5$ do the best job concerning organ placement and sound speed values. The SSIM-EIT reconstruction achieves a lower minimum in terms of the absolute error at roughly 3600 while the TV reconstruction only hits a minimum of approximately 3800. There is some subjectivity in picking the ideal reconstruction, and realistically many of the lung phantom results could be considered as optimal.

In the Bayesian framework, even with fewer than optimal samples, the statistics of the Markov chain representing the test phantom showed promising results. In order to continue the success of the reconstructions presented in this dissertation, there are a few changes we would like to make in the future as well as a new avenue to potentially explore.

There are three primary changes or extensions that could be implemented in Algorithms 1 and 4. The first is a more careful approach to selecting the regularization parameter α in Algorithm 1, the precision parameters λ and α in Algorithm 4, and the step length parameter h_n present in both algorithms. We discussed in chapters 3 and 4 less ad hoc methods for selecting these parameters, and future work would almost certainly require an implementation of one of these methods. It would also be beneficial to investigate a more rigorous stopping criterion for Algorithm 1.

The second improvement is more samples for Algorithm 4. As previously stated, the number of samples obtained for the test phantom was adequate to determine that the MALA FWI algorithm holds promise, but it was not enough to have full confidence that the statistics calculated using the Markov chain were accurate. It is possible that further computational improvements could be made to our algorithm in order to produce more samples more quickly, but short of that, obtaining more samples will require computational power and time. This could be helped along by using GPU computing capabilities to more efficiently parallelize the code.

The final extension to this work would be testing its capabilities on real data. Inevitably, the results obtained from using real data are much worse than results obtained using simulated data, and using real data introduces a host of new issues to contend with. The phantoms used in this

work did not include physical parameters such as elasticity, density variations, attentuation, or reflectivity. These parameters would provide more challenges to our algorithms and also allow for more realistic numerical phantoms to be explored before moving to real-world data. It would be interesting to see the effect of adding an EIT-SSIM regularization term when confronting real data as well as see the added benefit of uncertainty quantification from MALA-FWI in an actual diagnostic setting.

The above changes and improvements to the work presented in this dissertation are intriguing, and there is another related problem that would provide mathematical and diagnostic value. The reconstruction algorithm in section 4.3 is a Bayesian approach to imaging in a stationary setting. However, there are certain situations (i.e. bedside monitoring) in which obtaining a dynamic image of the lung would be of great merit. We thus want to extend the reconstruction approach outlined in this thesis to the dynamic problem and consider producing a real time (or close to it) image of a breathing lung.

This problem has already been explored with EIT in [122], but has not yet been thoroughly investigated for USCT. It would be interesting to consider different spatial and temporal updates within both the Kalman filter and extended Kalman filter [97, 123, 124] that make use of a lung model or make use of images taken at different states of inflation for the lung. In addition, comparing the performance of the standard Kalman filter, extended Kalman filter, and particle filter methods [125, 126] would be a worthwhile endeavor.

Bibliography

[1] Jago and Whittingham. Experimental Studies in Transmission Ultrasound Computed Tomography. *Physics in Medicine and Biology*, 36(11):1515–1527, 1991.

[2] Bronstein. Reconstruction in Diffraction Ultrasound Tomography Using Nonuniform FFT. *IEEE Transactions on Medical Imaging*, 21(11), November 2002.

[3] Kak and Slaney. *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics, 2001.

[4] Quan and Huang. Sound-speed Tomography Using First-arrival Transmission Ultrasound for a Ring Array. In Emelianov and McAleavey, editors, *Medical Imaging 2007: Ultrasonic Imaging and Signal Processing*, volume 6513, pages 48 – 56. International Society for Optics and Photonics, SPIE, 2007.

[5] Perez-Liva, Herraiz, Udias, Miller, Cox, and Treeby. Time Domain Reconstruction of Sound Speed and Attenuation in Ultrasound Computed Tomography using Full Wave Inversion. *The Journal of the Acoustical Society of America*, 2017.

[6] Watson. *Industrial Tomography*, chapter 9-Ultrasound Tomography. Woodhead Publishing Series in Electronic and Optical Materials, 2015.

[7] Devaney. Geophysical Diffraction Tomography. *IEEE Transactions on Geoscience and Remote Sensing*, GE-22(1):3–13, 1984.

[8] Duric, Littrup, Babkin, Chambers, Azebedo, Kalinin, Pevzner, Tokarev, Holsapple, Rama, and Duncan. Development of Ultrasound Tomography for Breast Imaging: Technical Assessment. *Medical Physics*, 32, 2005.

[9] Duric and Littrup. *Breast Imaging*, chapter Breast Ultrasound Tomography. InTech, 2018.

[10] Duric and Littrup. Clinical Breast Imaging with Ultrasound Tomography: A Description of the SoftVue System. *Journal of the Acoustical Society of America*, 2014.

[11] Guasch, Calderon Agudo, and Tang et al. Full-waveform Inversion Imaging of the Human Brain. *npj Digit. Med.*, 3(28), 2020.

[12] Rueter, Hauber, Droeman, Zabel, and Uhlig. Low-Frequency Ultrasound Permeates the Human Thorax and Lung: a Novel Approach to Non-Invasive Monitoring. *Ultraschall in der Medizin*, August 2009.

[13] Alsaker, Cardenas, Furuie, and Mueller. Complementary Use of Priors for Pulmonary Imaging with Electrical Impedance and Ultrasound Computed Tomography. *Journal of Computational and Applied Mathematics, to appear*, 2021.

[14] Mueller, Cardenas, and Furuie. A Preclinical Simulation Study of Ultrasound Tomography for Pulmonary Bedside Monitoring. In *Proceedings of the Second International Workshop on Medical Ultrasound Tomography (MUSTII)*, 2019.

[15] Li, Duric, and Huang. Clinical Breast Imaging Using Sound-Speed Reconstructions of Ultrasound Tomography Data. In *Ultrasonic Imaging and Signal Processing*, 2008.

[16] O'Flynn, Fromageau, Ledge, Messa, Aquino, Schoemaker, Schmidt, Duric, Swerdlow, and Bamber. Ultrasound Tomography Evaluation of Breast Density. *Investigative Radiology*, 2017.

[17] Matrone, Ramalli, and Tortoli. *Ultrasound B-Mode Imaging: Beamforming and Image Formation Techniques*. MDPI Books, 2019.

[18] Goodenough and Weaver. Overview of Computed Tomography. *IEEE Transactions on Nuclear Science*, NS-26(1), February 1979.

[19] Ranger, Littrup, Duric, Mody, Li, Schmidt, and Lupinacci. Breast Ultrasound Tomography versus MRI for Clinical Display of Anatomy and Tumor Rendering: Preliminary Results. *American Journal of Roentgenology*, 2011.

[20] Duric et al. Using Whole Breast Ultrasound Tomography to Improve Breast Cancer Risk Assessment: A Novel Risk Factor Based on the Quantitative Tissue Property of Sound Speed. *Journal of Clinical Medicine*, 2020.

[21] Virieux and Operto. An Overview of Full-waveform Inversion in Exploration Geophysics. *Geophysics*, 74(6), 2009.

[22] Toivanen, Meaney, Siltanen, and Kolehmainen. Joint Reconstruction in Low Dose Multi-Energy CT. *arXiv:1904.05671v3*, February 2020.

[23] Koponen, Huttunen, Tarvainen, and Kaipio. Bayesian Approximation Error Approach in Full-wave Ultrasound Tomography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 61(10):1627–1637, 2014.

[24] Zhao, Basarab, Kouame, and Tourneret. Restoration of Ultrasound Images using a Hierarchical Bayesian Model with a Generalized Gaussian Prior. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4577–4581, 2014.

[25] Zhu, Li, Fomel, Stadler, and Ghattas. A Bayesian Approach to Estimate Uncertainty for Full-waveform Inversion using a priori Information from Depth Migration. *GEOPHYSICS*, 81(5):R307–R323, 2016.

[26] Gebraad, Boehm, and Fichtner. Bayesian Elastic Full-Waveform Inversion Using Hamiltonian Monte Carlo. *Journal of Geophysical Research: Solid Earth*, 125(3):e2019JB018428, 2020. e2019JB018428 10.1029/2019JB018428.

[27] Greenleaf, Johnson, Lee, Herman, and Wood. Algebraic Reconstruction of Spatial Distributions of Acoustic Absorption Within Tissues from their Two-dimensional Acoustic Projections. *Acoustical Holography*, 5:591–603, 1974.

[28] Ramachandran and Lakshminarayanan. Three-dimensional Reconstruction from Radiographs and Electron Micrographs: Application of Convolutions Instead of Fourier Transforms. *Proceedings of the National Academy of the Sciences, USA*, 1971.

[29] Nebeker, Jakob, and Nelson. Imaging of Sound Speed using Reflection Ultrasound Tomography. *Journal of Ultrasound in Medicine*, 31(9):1389–1404, 2012.

[30] Mueller and Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*, chapter Pratical X-ray Tomography with Limited Data. SIAM, 2012.

[31] Glover and Sharp. Reconstruction of Ultrasound Propagation Speed Distribution in Soft Tissue: Time-of-Flight Tomography. *IEEE TRans. Sonics Ultrason.*, 24(4), 1977.

[32] Ozmen, Dapp, Zapf, Gemmeke, Ruiter, and van Dongen. Comparing Different Ultrasound Imaging Methods for Breast Cancer Detection. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 62(4), 2015.

[33] Janee, Andre, Ysrael, and Martin. *Diffraction Tomography Breast Imaging System: Patient Image Reconstruction and Analysis*, volume 24 of *Acoustical Imaging*, chapter 45. Springer, 2002.

[34] Simonetti, Huang, and Duric. Transmission and Reflection Diffraction Tomography in Breast Imaging. In *IEEE INt. Conf. Biomedical Engineering and Informatics*, 2008.

[35] Colton and Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer, 1993.

[36] Institute of Med Board on Biobehavioral Sciences and Mental Disorders. *Mathematics and Physics of Emerging Biomedical Imaging*. National Academy Press, 1996.

[37] Devaney. A Filtered Backpropagation Algorithm for Diffraction Images. *Ultrasonic Imaging*, 4, 1982.

[38] Kak and Slaney. *Principles of Computerized Tomography*. SIAM, Philadelpia, PA, 2001.

[39] F.T.A., Wajer, Lethmate, van Osch, Graveron-Demilly, Fuderer, and van Ormondt. Interpolation from Arbitrary to Cartesian Sample Positions: Gridding. In *Proc. ProRISC/IEEE Workshop*, 2000.

[40] Mueller, Cardenas, and Furuie. A Preclinical Simulation Study of Ultrasound Tomography for Pulmonary Bedside Monitoring. In *International Workshop on Medical Ultrasound Tomography*, 2019.

[41] Oelze and Lavarello. A Study on the Reconstruction of Moderate Contrast Targets using the Distorted Born Iterative Method. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 2008.

[42] Ruszczynski. *Nonlinear Optimization*. Princeton University Press, 2006.

[43] Kaczmarz. Approximate Solution of Systems of Linear Equations. *International Journal of Control*, 1993.

[44] Tanabe. Projection Method for Solving a Singular System of Linear Equations and its Applications. *Numer. Math.*, 1971.

[45] Bernard, Monteiller, Komatitsch, and Lasaygues. Ultrasonic Computed Tomography Based on Full-waveform Inversion for Bone Quantitative Imaging. *Physics in Medicine and Biology*, 62(17), 2017.

[46] Agudo, Guasch, Huthwaite, and Warner. 3D Imaging of the Breast using Full-Waveform Inversion. In *Proceedings on the International Workshop on Medical Ultrasound Tomography*, 2017.

[47] Wang, Matthews, Anis, Li, Duric, and Anastasio. Waveform Inversion with Source Encoding for Breast Sound Speed Reconstruction in Ultrasound Computed Tomography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 62(3), 2015.

[48] Matthews, Wang, Li, Duric, and Anastasio. Regularized Dual Averaging Image Reconstruction for Full-Wave Ultrasound Computed Tomography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 64(5), 2017.

[49] Warner et al. Anisotropic 3D Full-waveform Inversion. *Geophysics*, 78, 2013.

[50] Routh et al. Impact of High-resolution Full Waveform Inversion in the Western Black Sea: Revealing Overburden and Reservoir Complexity. *Leading Edge*, 36:60–66, 2017.

[51] Wiskin, Malik, Natesan, Borup, Pirshafiey, Lenox, and Klock. Full Wave 3D Inverse Scattering Transmission Ultrasound Tomography: Breast and Whole Body Imaging. In *2019 IEEE International Ultrasonics Symposium (IUS)*, pages 951–958, 2019.

[52] Sandhuand, Li, Roy, Schmidt, and Duric. Frequency Domain Ultrasound Waveform Tomography: Breast Imaging Using a Ring Transducer. *Physics in medicine and biology*, 60:5381–5398, 06 2015.

[53] Engl, Werner, Hanke, and Neubauer. *Regularization of Inverse Problems*, volume 375. Springer Science and Business Media, 1996.

[54] Golub, Howard, and Von Matt. *Tikhonov regularization for large scale problems*. Scientific Computing and Computational Mathematics Program, Computer Science Department, Stanford University, 1997.

[55] Vogel. *Computational Methods for Inverse Problems*. SIAM, 2002.

[56] Rudin, Leonid, Osher, and Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms, volume = 60. *Physica D: nonlinear phenomena*, pages 259–268, 1992.

[57] Estrela, Magalhaes, and Saotome. Total Variation Applications in Computer Vision. *https://arxiv.org/pdf/1603.09599.pdf*, 2016.

[58] Cheney, Isaacson, and Newell. Electrical Impedance Tomography. *SIAM Review*, 41(1), 1999.

[59] Saulnier, Blue, Newell, Isaacson, and Edic. Electrical Impedance Tomography. *IEEE Signal Processing Magazine*, 18(6):31–43, 2001.

[60] Gunther Uhlmann. Electrical Impedance Tomography and Calderón's Problem. *Inverse Problems*, 25(12):123011, 2009.

[61] Hanke and Brühl. Recent Progress in Electrical Impedance Tomography. *Inverse Problems*, 19(6):S65, 2003.

[62] Mueller and Siltanen. The d-bar Method for Electrical Impedance Tomography—Demystified. *Inverse Problems*, 36(9):093001, 2020.

[63] Robert G. Bartle. *The Elements of Integration and Lebesgue Measure*. John Wiley & Sons, New York, 1995.

[64] John Sylvester and Gunther Uhlmann. The Dirichlet to Neumann Map and Applications. *Inverse Problems in Partial Differential Equations*, 42:101, 1990.

[65] Liliana Borcea. Electrical Impedance Tomography. *Inverse Problems*, 18(6):R99, 2002.

[66] Islam and Kiber. Electrical Impedance Tomography Imaging using Gauss-Newton Algorithm. In *2014 International Conference on Informatics, Electronics Vision (ICIEV)*, pages 1–4, 2014.

[67] Nilsson and Akenine-Moller. Understanding SSIM. *arxiv:2006.13846v2*, June 2020.

[68] Zhou, Bovik, Sheikh, and Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[69] Treeby, Cox, and Jaros. *k-Wave: A MATLAB Toolbox for the Time Domain Simulation of Acoustic Wave Fields*, 1.0.1 edition, November 2012.

[70] Pierce. *Acoustics: An Introduction to its Physical Principles and Applications*. New York: Acoustical Society of America, 1989.

[71] Hamilton and Blackstock. *Nonlinear Acoustics*. Melville: Acoustical Society of America, 2008.

[72] Pierce. *Handbook of Acoustics*, chapter Mathematical Theory of Wave Propagation. New York: John Wiley and Sons, 1998.

[73] Treeby, Jaros, Rendell, and Cox. Modeling Nonlinear Ultrasound Propagation in Heterogeneous Media with Power Law Absorption using a k-space Pseudospectral Method. *J. Acoust. Soc. Am*, 131(6), 2012.

[74] Treeby and Cox. A k-space Greens Function Solution for Acoustic Initial Value Problems in Homogeneous Media with Power Law Absorption. *J. Acoust. Soc. Am.*, 129(6), 2011.

[75] Bojarski. The k-space Formulation of the Scattering Problem in the Time Domain. *J. Acoust. Soc. Am.*, 72(2), 1982.

[76] Mast, Souriau, Liu, Tabei, Nachman, and Waag. A k-space Method for Large-scale Models of Wave Propagation in Tissue. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 48(2), 2001.

[77] Tabei, Mast, and Waag. A k-space Method for Coupled First-order Acoustic Propagation Equations. *J. Acoust. Soc. Am.*, 111(1), 2002.

[78] Norton. Iterative Inverse Scattering Algorithms: Methods of Computing Frechet Derivatives. *J. Acoust. Soc. Am.*, 106, 1999.

[79] Treeby, Tumen, and Cox. *Medical Image Computing and Computer-Assisted Intervention, Part I*, volume 6891, chapter Time Domain Simulation of Harmonic Ultrasound Images and Beam Patterns in 3D Using the k-space Pseudospectral Method, pages 363–370. Spring, Heidelberg, 2011.

[80] Barzilai and Borwein. Two-point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 1988.

[81] Burdakov, Huang, and Dai. Stabilized Barzilai-Borwein Method. *arXiv:1907.06409v3*, 2019.

[82] Fletcher. *On the Barzilai-Borwein Method*. Springer, 2005.

[83] Hansen. The L-curve and its Use in the Numerical Treatment of Inverse Problems. 1999.

[84] Liao, Li, and Ng. Selection of Regularization Parameter in Total Variation Image Restoration. *JOSA A*, 26(11):2311–2320, 2009.

[85] Ghosh. *Signals and Systems*. Pearson India, 2005.

[86] Li, Duric, Littrup, and Huang. In Vivo Breast Sound-Speed Imaging with Ultrasound Tomography. *Ultrasound in Medicine and Biology*, 35(10):1615–1628, 2009.

[87] Insana and Brown. *Ultrasonics Scattering in Biological Tissue*, chapter Acoustic Scattering Theory Applied to Soft Biological Tissues, pages 75–124. Boca Raton: CRC Press, 1993.

[88] Chen and Holm. Fractional Laplacian Time-space Models for Linear and Nonlinear Lossy Media Exhibiting Arbitrary Frequency Power-law Dependency. *J. Acoust. Soc. Am.*, 115(4):1424–1430, 2004.

[89] Treeby and Cox. Modeling Power Law Absorption and Dispersion for Acoustic Propagation using the Fractional Laplacian. *J. Acoust. Soc. Am.*, 127(5):2741–2748, 2010.

[90] Medina-Valdes, Perez-Liva, Camacho, Udias, Herraiz, and Gonzalez-Salido. Multi-modal Ultrasound Imaging for Breast Cancer Detection. *Phys. Procedia*, 63:134–140, 2015.

[91] Speed of Sound Tissue Properties. https://itis.swiss/virtual-population/tissue-properties/database/acoustic-properties/speed-of-sound/. Accessed: 2021-10-19.

[92] Kim, Boverman, Newell, Saulnier, and Isaacson. The Complete Electrode Model for EIT in a Mammography Geometry. *Physiological measurement*, 28(7):S57, 2007.

[93] Athreya and Lahiri. *Measure Theory and Probability Theory*. Springer Texts in Statistics, 2006.

[94] Reigh and Ghosh. *Bayesian Statistical Methods*. CRC Press, 1st edition, 2019.

[95] Bardsley. *Computational Uncertainty Quantification for Inverse Problems*. Computational Science and Engineering. SIAM, 2018.

[96] Robert. *The Bayesian Choice*. Springer, 2001.

[97] Kaipio and Somersalo. *Statistical and Computational Inverse Problems*. Springer, 1988.

[98] Chib. Chapter 57 - Markov Chain Monte Carlo Methods: Computation and Inference. In *Handbook of Econometrics*, volume 5, pages 3569 – 3649. Elsevier, 2001.

[99] Neal. *Probabilistic Inference using Markov Chain Monte Carlo Methods*. Canada: Department of Computer Science, 1993.

[100] Murray. *Advances in Markov Chain Monte Carlo Methods*. PhD thesis, University College London, 2002.

[101] Hastings. Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika*, 57(1), April 1970.

[102] Tierney. Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

[103] Tierney and Mira. Some Adaptive Monte Carlo Methods for Bayesian Inference. *Statistics in Medicine*, 18(17-18):2507–2515, 1999.

[104] Hitchcock. A History of the Metropolis-Hastings Algorithm. *The American Statistician*, 57(4):254–257, 2003.

[105] Richey. The Evolution of Markov Chain Monte Carlo Methods. *The American Mathematical Monthly*, 117(5):383–413, 2010.

[106] Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[107] Robert and Gasella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics, 2nd edition edition, 2004.

[108] Stormark. Multiple Proposal Strategies for Markov Chain Monte Carlo. Master's thesis, Department of Mathematical Sciences, Norwegian University of Science and Technology, 2006.

[109] Haario, Saksman, and Tamminen. An Adaptive Metropolis Algorithm. *Bernoulli*, 7(2):223–242, 2001.

[110] Haario, Saksman, and Tamminen. Adaptive Proposal Distribution for Random Walk Metropolis Algorithm. *Computational Statistics*, 1999.

[111] Neal. Taking Bigger Metropolis Steps by Dragging Fast Variables. Technical Report 0411, Department of Statistics, University of Toronto, October 2004.

[112] Gasemyr. On an Adaptive Version of the Metropolis-Hastings Algorithm with Independent Proposal Distribution. *Scandinavian Journal of Statistics*, 30(1):159–173, 2003.

[113] Girolami and Calderhead. Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[114] Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv: stat.ME*, 2018.

[115] Xifara et al. Langevin Diffusions and the Metropolis-adjusted Langevin Algorithm. *Statistics and Probability Letters*, 2014.

[116] Manguobi and Vishnoi. Nonconvex Sampling with the Metropolis-adjusted Langevin Algorithm. *arXiv: stat.MpE*, 2019.

[117] Roberts and Tweedie. Exponential Convergence of Langevin Distributions and their Discrete Approximations. *Bernoulli*, 2(4):341–363, Dec. 1996.

[118] Bonesky. Morozov's Discrepancy Principle and Tikhonov-type Functionals. *Inverse Problems*, 25(1):015015, 2008.

[119] Allmaras, Bangerth, Linhart, Polanco, F. Wang, K. Wang, Webster, and Zedler. Estimating Parameters in Physical Models through Bayesian Inversion: A Complete Example. *siam REVIEW*, 55(1):149–167, 2013.

[120] Kruschke. Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. 2014.

[121] Cowles and Carlin. Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association*, 1996.

[122] Trigo, Gonzalez-Lima, and Amato. Electrical Impedance Tomography using the Extended Kalman Filter. *IEEE Transactions on Biomedical Engineering*, 51(1), 2004.

[123] Ribeiro. Kalman and Extended Kalman Filters: Concept, Derivation and Properties. *Institute for Systems and Robotics*, 2004.

[124] Chui and Chen. *Kalman Filtering with Real-Time Applications*. Springer, fifth edition, 2017.

[125] Carpenter, Clifford, and Fearnhead. Improved Particle Filter for Nonlinear Problems. *IEE Proceedings - Radar, Sonar and Navigation*, 146(1), 1999.

[126] David Gyorgy, Kelemen. Unscented Kalman Filters and Particle Filter Methods for Nonlinear State Estimation. *Procedia Technology*, 12:65–74, 2014.