NOVEL ALGORITHMS FOR MAGNETIC INDUCTION TOMOGRAPHY WITH APPLICATIONS IN SECURITY SCREENING

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

2020

Alexander J Hiles

School of Natural Sciences, Department of Mathematics

Contents

Word Count: 30000

List of Tables

List of Figures

List of Algorithms

Abstract

This thesis proposes novel reconstruction schemes for low frequency near-field electromagnetic imaging of high-contrast conductivity distributions enclosed inside shielded regions, using Maxwell's equations in 3D. Our focus lies in the estimation of conductivity value or shape enclosed in shielded regions from electromagnetic data measured externally, for one low frequency. We are interested in regions which are roughly equivalent in size to small rooms or medium-sized containers, though the reconstruction schemes proposed here can easily be adjusted to imaging situations at larger or smaller scale.

The novel regularization techniques proposed here are based on a sparsity promoting regularization scheme on the one hand, and level set based shape evolution techniques on the other. For estimating the conductivity profile enclosed inside these shielded regions, we introduce a sparsity regularization scheme and compare its result to the shape-based schemes developed here and a traditional $\mathcal{L}^2-$based approach. In the shape-based regime, we introduce single and color level set regularization schemes which are designed to reconstruct binary and multi-phase material respectively. Alongside color level set regularization, we introduce a topological perturbation scheme which is designed to avoid a certain type of local minima that is characteristic to simultaneous multi-value shape-based reconstruction. In each reconstruction scheme, Landweber-Kaczmarz iterations are employed for the optimization process, with suitable tailor-made line search techniques designed accordingly.

In our numerical simulations, we perform 3D reconstructions from noisy simulated data and compare the results with those obtained from a standard $\mathcal{L}^2-$based approach. Our results suggest, in the applications considered here, that the proposed novel schemes are able to yield significantly improved reconstructions when compared with traditional techniques. We end with using convolutional neural networks to classify electromagnetic images that result from the reconstruction schemes.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

i. Ministry of Defence ©British Crown Owned Copyright 2020/AWE. The University of Manchester has been granted certain rights to use such Copyright including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=2442` 0), in any relevant Thesis restriction declarations deposited in the University Library, The

University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations/)` and in The University's policy on Presentation of Theses

Acknowledgements

I would firstly like to give my sincere gratitude to my PhD supervisor Dr. Oliver Dorn for his continuous support during my PhD research, for his patience in difficult moments, his motivation for the project and his mentoring. His guidance during the research has been invaluable.

I would like to thank my PhD industrial sponsor Atomic Weapons Establishment (AWE) for their financial support and encouragement during the PhD project. I thank Maxim Joseph, Neil Gaspar and Robert Ward for their useful discussions during the project. It was a pleasure to visit the site in April 2018 to see the physical experiments in the flesh.

I would like to thank everyone I have met in the Alan Turing Building. Thanks for being welcoming, friendly and supportive. I have met some great people. Office 1.220 has been great and I will miss you guys! We are now masters of the countries sporcle quiz.

I would also like to thank my friends Kris, Howard, Ashley and the rest of the WhatsApp group. They have always provided laughs, support and love throughout my life. I am deeply grateful.

I would like to express my deepest gratitude and love to my mother Samantha, my grandad Jack, my brother Dan and the rest of my family for their unconditional support and love throughout.

Most importantly, I would like to thank Katy. She has provided unconditional support and love, and has always been there to listen and give advice in difficult moments - every step of the way. She has been amazing.

This page is intentionally left blank

This page is intentionally left blank

CHAPTER I

Introduction

MAGNETIC Induction Tomography (MIT) is a technique which is used to image the electromagnetic properties of an object, using the eddy current effect. Passing an alternating current through an excitation coil generates a primary magnetic field, which induces an electric field that is detectable by a receiver coil. This allows us to infer information about the object from electromagnetic quantities measured at the receiver coil. Using this imaging modality, we design schemes which recover information about objects inside boxes or containers from electromagnetic data measured at locations external to the boxes or containers under inspection (non-destructive testing).

MIT has not received much attention when compared to alternative electromagnetic techniques. Some of these alternatives include: ground-penetrating radar, synthetic aperture radar, x-ray tomography, microwave imaging, amongst others [90]. Admittedly, designing tailor-made algorithms to infer material properties from electromagnetic data is a difficult task in MIT. Typically, one has access to limited data and, frequently, a partial view of the imaging domain. Moreover, the corresponding imaging task usually requires solutions to Maxwell's equations which for a long time was severely limited by the capabilities of standard computer systems. In applications of security screening, where the imaging domain is typically surrounded by a electromagnetic cage (e.g. cargo container), further difficulties arise. The cages act as electromagnetic shields, meaning high frequency electromagnetic waves have difficulty penetrating. To avoid these problems, low frequency electromagnetic waves are employed to avoid the skin effect phenomenon (a reduction of electromagnetic wave penetration depth inside the metallic cage). In other words, at higher frequencies, information about the contents does not reach the receivers. This results in poor reconstruction of the imaging domain as it cannot be probed at high frequencies. The employment of low frequency electromagnetic waves comes with a penalty; it reduces resolution capabilities of algorithms designed to reconstruct objects inside these cages.

Despite the difficulties described above, MIT has been used in many applications, including: nondestructive testing [58], geophysics [36], brain imaging [116, 117], amongst others. The motivation of our imaging task, which will be introduced in section 1.2, is from experimental research carried out in [29, 30, 133]. In those works, it is demonstrated that extracting information about objects lying beneath electromagnetic shields is possible. In particular, information about the location of objects is found using difference imaging and low frequency electromagnetic waves. This is encouraging, as these results can be improved by developing algorithms which combine both electromagnetic data and mathematical models of electromagnetism.

The use of relatively low frequencies in such near-field electromagnetic imaging applications make it more difficult to obtain high resolution reconstructions, and the design of efficient antennas is challenging. However, over the years a number of interesting and practical applications have evolved where the use of relatively low frequencies comes with great advantages, such as: increased penetration depth through materials, a reduced risk of multipathing, and avoidance of certain local minima when solving the underlying optimization problem. These applications include geophysical imaging [1, 35, 36, 50, 104, 137], near field sensing and tracking [106, 129], and the non-destructive testing of materials [29, 80, 133]. The ongoing interest in these and other fields is continuously refuelled by improvements in computer technology and progress in the construction of efficient antenna systems, which addresses some of the above mentioned difficulties. However, it still remains challenging to design efficient tailor-made reconstruction algorithms which are able to convert the measured electromagnetic signals into useful 3D images. Typically, these images show electromagnetic properties of the imaging domain or make use of them for the localization or tracking of objects.

This thesis proposes novel reconstruction algorithms for near-field electromagnetic imaging of high-contrast conductivity distributions enclosed in shielded 3D regions. We introduce new reconstruction schemes in both pixel-based and shape-based regimes, where ‘pixel-based’ refers to the challenge of recovering a material parameter value and ‘shape-based’ refers to when a material parameter value is *a priori* known but its shape is not. We follow notation from the classical (mainly 2D) literature here, even though in this 3D imaging task voxels are assigned values not pixels. The low probing frequencies considered here, coupled with the conductivity values used for the shielded walls and container contents in the numerical experiments, allow us to consider reconstruction of the conductivity only, with the other usually important material parameters magnetic permeability and electric permittivity assuming their free space values.

Across both regimes, we propose a number of new reconstruction schemes, including one based on sparsity regularization and others based on level set shape evolution, and compare their performances with a more traditional pixel-based reconstruction scheme employing standard Tikhonov-Philips regularization. We will show our novel approaches have the potential to provide significantly improved reconstructions over pixel-based imaging schemes in situations where sharp interfaces and significant contrasts between materials are present in the imaging domain.

Sparsity promoting regularization has received worthy attention in recent years as an alternative to $\mathcal{L}^2-$based minimization schemes. This technique has been applied to both linear inverse problems [31, 45] and also, more recently, to nonlinear inverse problems [8, 11, 100]. It is well-known that standard Tikhonov-Philips regularized $\mathcal{L}^2-$based schemes tend to provide oversmoothed reconstructions from low frequency electromagnetic data, even when the true profiles contain shapes with sharp boundaries. Sparsity promoting regularization schemes have recently been shown to mitigate this smoothing tendency by providing more compact shapes against an approximately known background [68, 69, 100].

Whilst sparsity promoting schemes provide more compact shapes than traditional $\mathcal{L}^2$-based techniques, level set reconstruction schemes are designed explicitly to provide compact shapes with sharp boundaries against the background profile. For more information on level set methods see chapter 2, 4 and [16, 39, 41, 42, 66, 75, 78].

Level set regularization schemes have become a popular methodology when interested in solving shape-based inverse problems. For example, single level set inversion schemes have been designed for a multitude of applications. Some include: diffuse optical tomography [43, 100], non-destructive testing [38, 58], reservoir imaging [48, 127], medical imaging [40], geophysical prospecting [35], amongst others. Its extension, a multi level set regime or so-called *color* level sets, have also proven to be useful as they often depict more realistic scenarios. For example, a structural color level set method was developed in [65] to detect tumours in breast tissue and schemes have been developed in the history matching of petroleum reservoirs [44]. In addition, color level set methods have also been used in electrical capacitance tomography (ECT) to image permittivity [3].

We formulate both sparsity and level set based regularization schemes for a new and highly challenging imaging situation where a 3D Maxwell model is required as a forward solver. Even though the numerical experiments demonstrated here have been chosen to image boxes or medium-sized containers with shielded walls (see section 1.2 and chapter 3), the same techniques can be applied when imaging different sized domains. These include, for example, geophysical applications or the non-destructive testing of small specimens [35, 38]. Moreover, in addition to the well known and established geophysical applications of near-field exploration and monitoring tools [1, 35, 36, 50, 104, 137], we want to highlight here several interesting and promising new experimental verifications of low frequency screening and imaging applications at the scale of boxes or medium-sized containers. Physical experiments at similar scales have been addressed, for example, in [28, 29, 30, 133]. In those works, the idea is to screen objects beneath electromagnetic shields using measured electromagnetic data. The near-field electromagnetic imaging problem in this thesis is motivated by these physical experiments.

Using MIT for security screening applications is challenging when trying to obtain high resolution reconstructions, for reasons already mentioned above. However, it is even more challenging when considering the inherent nature of security screening applications. Regardless of the imaging modality used to probe contents of a package, box or container, the end objective is to determine whether the content poses a hazard or threat. In general, reconstructions which are obtained from various inversion schemes are analysed by an expert's eye. The expert is then typically guided by a data misfit and/or its appearance in comparison to the true phantom. Whilst in many applications qualitative analysis is sufficient or may not be important, in security screening knowing what objects or shapes appear inside is important. Using an expert's eye in such situations is time consuming and subjective, therefore introducing an automaton which performs this process for us seems reasonable. This idea is nothing new and has been used in other imaging modalities. For example, automatons are being used to detect vertebrae fractures from computerized tomography reconstructions of a patient [126]. The underlying problem of an automaton in our application, as well as in detecting vertebrae fracture, is to *classify* a reconstruction into certain groups which are pre defined. Initially, this may seem like a restriction on the automaton, but if certain objects or shapes are sought after and known in advance (e.g. sharp objects in suitcases), then these can be pre defined in the classification scheme. The automaton can be used to determine whether these objects or shapes are present. Appendix A introduces an automaton which is applied to images resulting from shape-based reconstruction schemes (also introduced here).

Shape-based schemes lend themselves well to classification tasks, as they segment the pixel-based images, providing unambiguous shapes which honour the electromagnetic data. The automaton or object/shape detection scheme developed here is independent to the inversion process and only tells us what it observes in images provided to it. The goal here is to provide an introduction into using machine learning techniques for post processing electromagnetic images which result from the reconstruction schemes.

1.1

Maxwell's equations

Maxwell's equations are a mathematical model of how electromagnetism interacts with matter and are fundamental to our imaging problem. Our main objective is to reconstruct material parameters, such as electrical conductivity or magnetic permeability, so that electromagnetic field solutions to Maxwell's equations coincide with electromagnetic quantities measured at specific locations. Inverting this electromagnetic data is the *inverse problem* we are interested in.

Our approach involves solving Maxwell's equations in the frequency domain, since this form provides a convenient way of picking out certain low frequency contributions to the time signal[1]. Solutions to Maxwell's equations in the frequency domain are used as measurements at specific locations, in order to generate electromagnetic data for when (1) the true object is present and (2) an estimated object is present. This is known as the *forward problem*. There are many methods available to solve this forward problem, therefore we must choose one which is appropriate for the inverse problem considered here. An extensive literature survey on this topic, including solving Maxwell's equations in the time domain, is carried out in section 3.1. Let us begin by introducing Maxwell's equations in the time domain:

$$\begin{aligned}
\nabla \times \mathcal{E}(\mathbf{x},t) &= -\partial_t \mathcal{B}(\mathbf{x},t); && (1.1a)\\
\nabla \times \mathcal{H}(\mathbf{x},t) - \partial_t \mathcal{D}(\mathbf{x},t) &= \mathcal{J}(\mathbf{x},t); && (1.1b)\\
\nabla . \mathcal{D}(\mathbf{x},t) &= \rho(\mathbf{x},t); && (1.1c)\\
\nabla . \mathcal{B}(\mathbf{x},t) &= 0, && (1.1d)
\end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^3$, $\mathcal{E}$ is the electric field intensity, $\mathcal{H}$ is the magnetic field intensity, $\mathcal{B}$ is the

---

[1] Note that we could also solve an equivalent problem in the time domain by discarding higher frequency contributions to the time signal.

magnetic flux density, $\mathcal{D}$ is the electric flux density, $\mathcal{J}$ is the electric current density and ρ is the electric charge density. This system governs electromagnetism, and is described by the four fields $\mathcal{E}, H, \mathcal{B}$ and $\mathcal{D}$. Note that (1.1d) is a consequence of applying the divergence operator to (1.1a). Hence, we have three linearly independent equations but four unknowns, i.e. an under determined system. We impose an assumption of linearity between fields to alleviate this problem. Namely:

$$\begin{aligned} \mathcal{D}(\mathbf{x},t) &= \varepsilon(\mathbf{x})\mathcal{E}(\mathbf{x},t); & (1.2a) \\ \mathcal{B}(\mathbf{x},t) &= \mu(\mathbf{x})\mathcal{H}(\mathbf{x},\mathbf{t}); & (1.2b) \\ \mathcal{J}(\mathbf{x},t) &= \sigma(\mathbf{x})\mathcal{E}(\mathbf{x},t) + \mathcal{J}_e(\mathbf{x},t), & (1.2c) \end{aligned}$$

where $\varepsilon(\mathbf{x})$ is the dielectric permittivity, $\mu(\mathbf{x})$ is the magnetic permeability, $\sigma(\mathbf{x})$ is the electric conductivity and $\mathcal{J}_e(\mathbf{x},t)$ is the applied current density. The first component of $\mathcal{J}$ is known as Ohm's Law and describes an induced current within a conductor. These relationships above are commonly called the constitutive relations, and describe the electromagnetic fields at the macroscopic level.

By substituting the constitutive relations defined in (1.2) into (1.1), we obtain

$$\begin{aligned} \nabla \times \mathcal{E}(\mathbf{x},t) + \partial_t(\mu(\mathbf{x})\mathcal{H}(\mathbf{x},t)) &= \mathbf{0}; & (1.3a) \\ \nabla \times \mathcal{H}(\mathbf{x},t) - \partial_t(\varepsilon(\mathbf{x})\mathcal{E}(\mathbf{x},t)) &= \sigma(\mathbf{x})\mathcal{E}(\mathbf{x},t) + \mathcal{J}_e(\mathbf{x},t); & (1.3b) \\ \nabla.(\varepsilon(\mathbf{x})\mathcal{E}(\mathbf{x},t)) &= \rho(\mathbf{x},t); & (1.3c) \\ \nabla.(\mu(\mathbf{x})\mathcal{H}(\mathbf{x},t)) &= 0. & (1.3d) \end{aligned}$$

This is a linear, first order hyperbolic partial differential equation (PDE) system with unknowns $\mathcal{E}(\mathbf{x},t)$ and $\mathcal{H}(\mathbf{x},t)$. The constitutive relations are commonly used when modelling Maxwell's equations, as they stop the Maxwell system from being under determined. Therefore, they form the basis of most electromagnetic models [21, 35, 36, 37, 52].

Many different variations of the system in (1.3) can be derived, subject to application. Two, for example, are the electrostatic and magnetostatic systems which are derived by removing the time derivative for the electric and magnetic fields respectively. We consider the frequency domain alternative of that in (1.3).

By the inverse Fourier transform, general solutions to Maxwell's equations can be obtained as a linear combination of single frequency contributions. The forward and inverse

Fourier transform for a time dependent electromagnetic field are:

$$\mathfrak{F}[\mathcal{Y}(\mathbf{x},t)] := \int_{-\infty}^{\infty} \mathcal{Y}(\mathbf{x},t)e^{-i\omega t}d\omega; \tag{1.4a}$$

$$\mathfrak{F}^{-1}[\mathbf{Y}(\mathbf{x},\omega)] := \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{Y}(\mathbf{x},\omega)e^{i\omega t}dt, \tag{1.4b}$$

respectively, where $\mathfrak{F}$ is the Fourier transform operator, $\omega = 2\pi f$ ($\omega \in \mathbb{R}$), and f is the frequency. The fields are assumed to admit a time dependence $e^{-i\omega t}$:

$$\mathcal{Y}(\mathbf{x},t) = \mathbf{Y}(\mathbf{x})e^{-i\omega t},$$

though $e^{i\omega t}$ is also possible. We use this dependency because it is standard and consistent with the literature. This effectively yields the mapping

$$\partial_t \underset{\text{TD}}{\overset{\text{FD}}{\rightleftarrows}} -i\omega, \tag{1.5}$$

where FD and TD are acronyms for frequency domain and time domain respectively. Applying the mapping in (1.5) from left to right brings the system in (1.3) to the form

$$\nabla \times \mathbf{E}(\mathbf{x}) - i\omega\mu(\mathbf{x})\mathbf{H}(\mathbf{x}) = \mathbf{M}(\mathbf{x}); \tag{1.6a}$$

$$\nabla \times \mathbf{H}(\mathbf{x}) - (\sigma(\mathbf{x}) - i\omega\varepsilon(\mathbf{x}))\mathbf{E}(\mathbf{x}) = \mathbf{J}(\mathbf{x}); \tag{1.6b}$$

$$\nabla.(\varepsilon(\mathbf{x})\mathbf{E}(\mathbf{x})) = \rho(\mathbf{x}); \tag{1.6c}$$

$$\nabla.(\mu(\mathbf{x})\mathbf{H}(\mathbf{x})) = 0, \tag{1.6d}$$

where $\mathbf{M}(\mathbf{x})$ is an artificial addition, due to the symmetry of generating the electromagnetic fields. The electric flux quantity $\rho(\mathbf{x})$ is assumed to be zero in our case.

1.2

The near-field electromagnetic imaging problem

Following the considerations in section 1.1, we model the propagation of electromagnetic fields by Maxwell's equations in the frequency domain. We recast the system in (1.6) as:

$$\nabla \times \mathbf{E}_j(\mathbf{x}) - a(\mathbf{x})\mathbf{H}_j(\mathbf{x}) = \mathbf{M}_j(\mathbf{x}); \tag{1.7a}$$

$$\nabla \times \mathbf{H}_j(\mathbf{x}) - b(\mathbf{x})\mathbf{E}_j(\mathbf{x}) = \mathbf{J}_j(\mathbf{x}), \tag{1.7b}$$

where **E** denotes the electric field, **H** the magnetic field, **M** the magnetic source and **J** the electric source. The material parameters are

$$a(\mathbf{x}) = i\omega\mu(\mathbf{x}) \qquad b(\mathbf{x}) = \sigma(\mathbf{x}) - i\omega\varepsilon(\mathbf{x}),$$

where $\sigma(\mathbf{x})$ is the electrical conductivity, $\mu(\mathbf{x})$ is the magnetic permeability and $\varepsilon(\mathbf{x})$ is the electrical permittivity. The subscript j denotes that the jth source distribution is considered, where $j = 1, \ldots, n_s$, and n_s is the total number of different applied sources. These are modelled as rectangular wire loops, and give rise to probing fields $\mathbf{E}_j$ and $\mathbf{H}_j$ in (1.7) which are then measured at receiver locations (also modelled as rectangular wire loops).

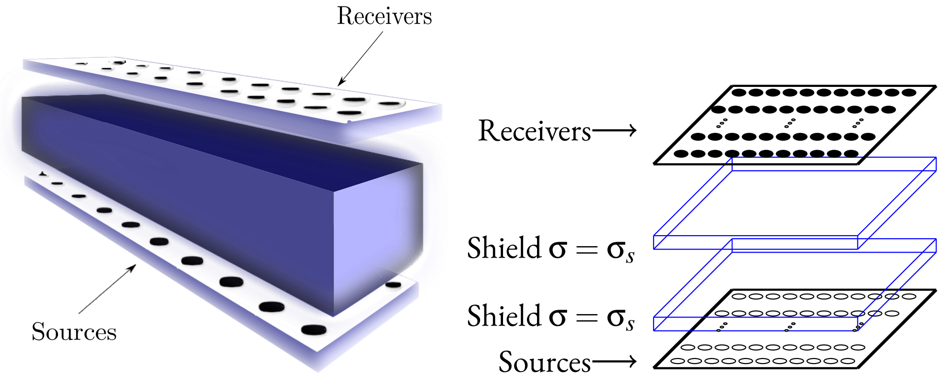


Figure 1.1: Opposing arrays of sources and receivers with shielded container in between. The sources and receivers are only located on top and bottom of the box, but other setups are also possible. Left: Complete set up of imaging the contents of a shielded cage. Right: Schematic of shielding in z direction. The domain of interest is between these shields in each direction.

A possible experimental setup of how the sources and receivers could be distributed is shown in figure 1.1. The domain of interest is represented by a cuboid-shaped domain $\Omega \subset \mathbb{R}^3$ surrounded by shielded walls. Sources and receivers are only located at the two opposing in the xy plane (i.e. facing each other in the z direction), but other choices are possible. Here, we consider no sources or receivers at the other four sides of the box. This particular setup has interesting applications in the surveillance of activities behind walls, the screening of boxes or containers at ports or airports, and the monitoring of processes inside regions which might contain hazardous materials.

The imaging task is to invert electromagnetic data measured at receiver locations when a box or container is present with unknown contents. We focus on estimating the pixel value or shape of the conductivity $\sigma(\mathbf{x})$ inside the domain of interest (box or container contents) from electromagnetic data measured outside of it. In our proof of concept style approach, we assume that both electric permittivity and magnetic permeability remain constant and take their respective values of free space (i.e. $\mu(\mathbf{x}) = \mu_0$ and $\varepsilon(\mathbf{x}) = \varepsilon_0$). At low probing frequencies, these quantities are expected not to have significant impact on the electromagnetic data measurements. Nevertheless, a joint inversion of these parameters with the electrical conductivity would be desirable.

In addition to the assumptions above, we assume some prior knowledge on the background electrical conductivity profile. In particular, that the shielded walls surround a region of interest filled with air. Embedded in this region of interest are inclusion(s) with unknown contrast, topology and shapes. These inclusions need to be reconstructed from a limited amount of electromagnetic data at one single frequency. We assign a low value $\sigma \approx 10^{-8}$ to the background representing the conductivity of air. The shielded walls, on the other hand, are assumed to have a known topology and some moderate constant conductivity value of $\sigma = \sigma_s$. It is possible to relax this assumption and also recover the shielded walls, but we do not consider this. The conductivity values of the inclusions are assumed to be higher than the background, but the values are unknown in the inverse problem and need to be reconstructed together with their shapes and topology. We consider multiple scenarios where the embedded inclusions admit various shapes, sizes and conductivity values. These are considered in chapter 4.

1.3
Publications and conferences

This thesis contains works found in papers that have either been published or submitted. These are:

1. A.J. Hiles and O. Dorn. Sparsity and level set regularization for near-field electromagnetic imaging in 3D. Inverse Problems, 36(2):025012, 2020.

2. A.J. Hiles and O. Dorn. Colour level set regularization for the electromagnetic imaging of highly discontinuous parameters in 3D. Submitted to Inverse Problems in Science and Engineering, 2020.

Conferences that I have attended and given talks at during my PhD:

1. Electromagnetic Non-Destructive Evaluation XXI (2017) Saclay, France

2. Invited talk at Atomic Weapons Establishment (AWE) Blacknest (2018) Aldermaston, UK
3. Applied Inverse Problems (2019) Grenoble, France
4. 2nd IMA Conference On Inverse Problems (2019) London, UK

Chapters 2 and 4 present work which appear in the two articles mentioned above.

1.4

Thesis structure

Chapter 2 begins with a general introduction to the inverse problem under investigation. Following that, novel reconstruction schemes and new line search criteria are developed in section 2.1. The schemes are based on two main approaches: sparsity promoting and level set shape-based evolution. In the latter approach, we introduce a stochastic seeding method. It is used to try and avoid certain local minima, which have been observed in numerical experiments when using these methods.

Chapter 3 provides a review of existing methods to solve Maxwell's equations both analytically and numerically. In order to provide numerical experiments involving the reconstruction schemes developed in chapter 2, we choose to use the finite volume method on a vector potential formulation of Maxwell's equations. We briefly outline the method in section 3.2. The numerical implementation of the Maxwell solver, along with the novel reconstruction schemes, are implemented in the Python programming language. It is numerically validated in section 3.5. The software package, which we have named "lsMax", will be available on GitHub soon.

Chapter 4 carries out numerical experiments for the near-field electromagnetic imaging problem described in section 1.2. They involve comparisons between the new reconstruction schemes and more traditional approaches. Subsets of these results are also described in the publications listed in section 1.3.

Chapter 5 provides concluding remarks from the work and possible directions of future research.

Appendix A introduces an object detection scheme using convolutional neural networks. The scheme detects shapes in images which result from the new reconstruction schemes described in chapter 2.

CHAPTER 2

Inverse Problem

Inverse Problems for Maxwell's equations have been investigated for a long time in a variety of applications and settings. For recent overviews of available theoretical and computational results we refer to [23, 25, 137] and the references provided there. The important questions of uniqueness and differentiability for related inverse problems are addressed for example in [15, 20, 56, 96, 119]. For derivation of the algorithms in this thesis, we assume differentiability (in properly chosen function spaces) of all forward mappings that arise here, allowing us to adopt well-established expressions for the relevant derivatives (or sensitivities) with respect to the unknown medium parameters [36, 37, 82, 118, 137]. Alternative techniques for solving the inverse problem, which are not considered in this thesis, are polarization and monotonicity based methods, amongst others. See [4, 54, 74, 122, 123, 124] for more details on those.

In this chapter, we formulate various reconstruction schemes which are concerned with estimating the value or shape of conductivity $\sigma(\mathbf{x})$. For notational purposes, we reduce Maxwell's equations described in (1.7a) and (1.7b) to the form

$$\Lambda(a_0, b)\mathbf{u}_j = \mathbf{q}_j, \tag{2.1}$$

Definition 2.1: Electrical Conductivity Inverse Problem

Assume that we have n_s source distributions $\mathbf{q}_j$ ($j = 1, 2, \ldots, n_s$) located at $\mathbf{s}_j$ such that radiation from the source is propagated by the Maxwell model in (2.1). In addition, let there be n_r receivers, where each of those are located at $\mathbf{r}_{jk}(k = 1, 2, \ldots, n_r)$. At these receiver locations, measure the electromagnetic field $\mathbf{E}$ when an unknown object is present in our domain $\boldsymbol{\Omega}$. Hence, let

$$D_{j,k} = \mathbf{E}_j(\mathbf{r}_k). \;\; k = 1, 2, \ldots, n_{s,j}$$

Then, the inverse problem is stated as follows: Find a parameter $b^* \in P$ such that the solution $\hat{\mathbf{E}}_j$ of

$$\Lambda(a_0, b^*)\hat{\mathbf{u}}_j = \mathbf{q}_j$$

coincides with $D_{j,k}$, where $\Lambda(\cdot,\cdot)$ is the Maxwell operator defined in (2.2).

where

$$\Lambda(a_0, b) = \begin{pmatrix} -b & \nabla\times \\ \nabla\times & a_0 \end{pmatrix}, \tag{2.2}$$

a_0 denotes that we take the free space value of $\mu(\mathbf{x})$, $\mathbf{u}_j = (\mathbf{E}_j \; \mathbf{H}_j)^T$ and $\mathbf{q}_j = (\mathbf{J}_j \; \mathbf{M}_j)^T$.

When the conductivity is everywhere positive, i.e. $\mathrm{Re}(b(\mathbf{x})) > 0$, (2.1) has a unique solution $\mathbf{u} \in U$ for each source $\mathbf{q}$, meaning it is well posed [37]. This notation allows us to separate operators and quantities in the Maxwell PDE system, which is useful when formulating the inverse problem. Definition 2.1 describes the inverse problem which this thesis addresses. We develop novel schemes to solve this problem for the near-field electromagnetic imaging problem described in section 1.2.

Let us denote P as the space of parameters $\mathbf{b}$ (which are assumed here for simplicity to be isotropic together with μ_0 and ε_0), Z_j the space of measurements D_j and noiseless residuals $\mathcal{R}_j$, U the spaces of states $\mathbf{u}$ and Y the space of sources. We assume these function spaces are represented by Hilbert spaces consisting of $\mathcal{L}^2$- functions which carry a special physical significance. All these function spaces are then equipped with standard inner products in

$\mathcal{L}^2$. They are defined as:

$$\begin{aligned}
P &:= \mathcal{L}^2(\Omega); \\
Z_j &:= \left\{ \begin{pmatrix} \eta \\ \xi \end{pmatrix}, \eta, \xi \in (\mathbb{C}^3)^{n_{s,j}} \right\}; \\
U &:= \left\{ \begin{pmatrix} \mathbf{E}(\mathbf{x}) \\ \mathbf{H}(\mathbf{x}) \end{pmatrix}, \mathbf{E}, \mathbf{H} \in \left[\mathcal{L}^2(\Omega) \right]^3 \right\}; \\
Y &:= \left\{ \begin{pmatrix} \mathbf{J}(\mathbf{x}) \\ \mathbf{M}(\mathbf{x}) \end{pmatrix}, \mathbf{J}, \mathbf{M} \in \left[\mathcal{L}^2(\Omega) \right]^3 \right\}.
\end{aligned}$$

The inner products on these spaces are:

$$\begin{aligned}
\left\langle \begin{pmatrix} \mathbf{E}_1(\mathbf{x}) \\ \mathbf{H}_1(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} \mathbf{E}_2(\mathbf{x}) \\ \mathbf{H}_2(\mathbf{x}) \end{pmatrix} \right\rangle_U &= \int_\Omega \mathcal{D}_E^2 \mathbf{E}_1 \overline{\mathbf{E}_2}\, d\mathbf{x} + \int_\Omega \mathcal{D}_H^2 \mathbf{H}_1 \overline{\mathbf{H}_2}\, d\mathbf{x}; \\
\left\langle \begin{pmatrix} \mathbf{J}_1(\mathbf{x}) \\ \mathbf{M}_1(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} \mathbf{J}_2(\mathbf{x}) \\ \mathbf{M}_2(\mathbf{x}) \end{pmatrix} \right\rangle_Y &= \int_\Omega \mathcal{D}_J^2 \mathbf{J}_1 \overline{\mathbf{J}_2}\, d\mathbf{x} + \int_\Omega \mathcal{D}_M^2 \mathbf{M}_1 \overline{\mathbf{M}_2}\, d\mathbf{x}; \\
\langle m_1, m_2 \rangle_Z &= \sum_{n=1}^{k} \sum_{i=x,y,z} \mathcal{D}_M^2 m_{1,n}^{i} \overline{m_{2,n}^{(i)}}; \\
< \mathbf{diag}(b_1^{(x)}, b_1^{(y)}, b_1^{(z)}), \mathbf{diag}(b_2^{(x)}, b_2^{(y)}, b_2^{(z)}) >_P &= \int_\Omega \mathcal{D}_b^2 \left(\sum_{i=x,y,z} b_1^{(i)} \overline{b_2^{(i)}} \right) d\mathbf{x}; \\
\langle f, g \rangle_{\mathcal{L}^2(\Omega)} &= \int_\Omega f g\, d\mathbf{x},
\end{aligned}$$

where $\mathcal{D}_\alpha = [\alpha]^{-1}$ denotes the dimension of a physical quantity. The functions spaces and their corresponding inner products are useful when insisting that solutions of the inverse problem belong to certain function spaces.

In the following discussion, we assume additional regularity of the electromagnetic fields **E** and **H** such that point-wise evaluation is defined. This condition is for purposes of point-wise measurement discussion only (which follows after introduction of the measurement operator below), as in this thesis we actually use a different form of the measurement operator which does not require this additional regularity condition.

Let us now define the *measurement operator* $\mathcal{M}_j : U \to Z_j$:

$$\mathcal{M}_j \mathbf{u}_j(\mathbf{x}) := \left\{ \int_\Omega \mathbf{E}_j(\mathbf{x}) \delta(\mathbf{x} - r_{j,k})\, d\mathbf{x} \right\}_{k=1,2,\ldots,n_r}. \tag{2.4}$$

The form in (2.4) is a mathematical representation of measurement at a receiver location. The *measurement operator* allows us to create a metric between measurements which are implicitly generated by different conductivities and those which are generated by the true content. As mentioned above, we assume here that the electric and magnetic fields have sufficient regularity to justify such use of Dirac delta functions. In case of uncertainty, the Dirac delta function can be replaced by a more realistic measurement functional that is tailored to represent specific receiver profiles. In fact, we use a different form of the measurement operator for the reconstruction schemes developed in this thesis - it amounts to taking wire loop measurements rather than point-wise ones. Overall, the results of this thesis do not depend significantly on the choice of the measurement operator. In addition, the smooth assumption does not hinder us in using $\mathcal{L}^2$-spaces (or related Sobolev spaces) to design inversion algorithms, given those quantities are also well defined. For more details on how to integrate different function spaces in the shape reconstruction framework of low frequency electromagnetics we refer to, for example, the work in [59], which uses the eddy current approximation for this purpose or [96] which uses the full Maxwell system for a similar setup.

This brings us to define the *residual operator* $\mathcal{R}_j : P \to Z_j$:

$$\mathcal{R}_j[b] = \mathcal{M}_j \mathbf{u}_j(\mathbf{x}; b) - \mathbf{d}_j, \tag{2.5}$$

where $\mathbf{u}(\mathbf{x}; b)$ is a solution of (2.1) with parameter b and $\mathbf{d}_j$ are data sets. The data sets $\mathbf{d}_j$ are generated when a box or container has unknown contents and are typically measured at the same locations for all j, where their source patterns are distinct. The residual operator gives us a quantitative measure in data space Z_j for the distance between an electromagnetic field generated by a parameter distribution b and one which is generated by a value in a small neighbourhood of the unknown parameter distribution b^* (due to noise in the physical measurement). The relation in (2.5) describes the data mismatch with respect to the jth source, whereby a sum over j results in the total data mismatch $\mathcal{R}[b]$. The goal is to minimize this mismatch by some metric, such that we find the conductivity profile that generated the true electromagnetic data $\mathbf{d}_j$. Let us now introduce a cost functional in terms of the residual operator described in (2.5):

$$\mathcal{J}_j[b] = \frac{1}{2}||\mathcal{R}_j[b]||^2_{\mathcal{L}^2(\Omega)}. \tag{2.6}$$

Introduction of the residual operator and the cost functional in (2.6) allows us to rewrite Definition 2.1 in this new framework.

This thesis is concerned with a single parameter inversion involving the conductivity $\sigma = \mathrm{Re}(b)$. As discussed in section 1.2, the other material parameters μ and ε assume their free space values. Whilst we derive gradients with respect to the complex admittivity b in this chapter, which could in theory be used for a joint inversion involving ε, we only update the real component of b throughout this thesis.

Definition 2.2: Electrical Conductivity Inverse Problem

Find a function b^* such that

$$b^* = \operatorname*{argmin}_{b \in P} \mathcal{J}[b], \tag{2.7}$$

where

$$\mathcal{J}[b] = \sum_{j=1}^{n_s} \mathcal{J}_j[b]. \tag{2.8}$$

For simplifying notation in parts of this thesis, we also use b to denote conductivity. It is clear from the context whether b is the complex admittivity or the conductivity. We use this notation because the imaginary component is fixed and constant throughout this thesis. Therefore, whenever we use a parameter b to minimize some form of the cost functional in Definition 2.2, the imaginary component is assumed to be $\omega\varepsilon_0$.

A standard approach of minimizing the cost functional in (2.8) is to find a search direction δb such that $b + \delta b$ results in a reduction of $\mathcal{J}[b]$. Therefore, consider the mapping $b \mapsto b + \delta b$. This results in:

$$\mathcal{J}[b + \delta b] = \frac{1}{2}||\mathcal{R}[b + \delta b]||^2_{\mathcal{L}^2(\Omega)}. \tag{2.9}$$

By linearizing the nonlinear operator $\mathcal{R}$, assuming $\mathcal{R}'$ exists and is well defined at b, we have that

$$\mathcal{R}[b + \delta b] = \mathcal{R}[b] + \mathcal{R}'[b]\delta b + O(||\delta b||^2_P). \tag{2.10}$$

We make no assumptions on the differentiability of $\mathcal{R}$ (i.e. whether it is Gateaux or Fréchet differentiable). We follow standard approaches when calculating gradients in MIT. For information on differentiability of related problems see [18, 59, 89, 120]. For the correction δb to be a valid perturbation in the parameter space such that $\mathcal{J}$ is minimized, we require that $\mathcal{R}[b+\delta b]=0$. And so,

$$\mathcal{R}'[b]\delta b = -\mathcal{R}[b], \tag{2.11}$$

or component wise

$$\mathcal{R}_j'[b]\delta b = -\mathcal{R}_j[b] \quad j=1,2,\ldots,n_s. \tag{2.12}$$

Solving (2.12) is a highly ill-posed problem, as solutions δb are not necessarily unique [36]. Moreover, (2.12) is typically under determined (i.e. less data than unknowns). Therefore, regularization is typically applied so that certain types of solutions are sought after. For example, a popular choice is the minimal norm solution, i.e. $\delta b^* = \min||\delta b||_{\mathcal{L}^2(\Omega)}$. We will pay closer attention to regularization techniques when addressing the fully nonlinear inverse problem later in this chapter.

Assuming that the system in (2.11) is under determined, which is the most realistic scenario, we have that

$$\delta b^* = -\mathcal{R}'[b]^*\left(\mathcal{R}'[b]\mathcal{R}'[b]^*\right)^{-1}\mathcal{R}[b], \tag{2.13}$$

where $\mathcal{R}'[b]^*$ is the adjoint operator of $\mathcal{R}'[b]$ with respect to P and Z. The argument of the inverse operator in (2.13) requires regularization because it is highly ill-conditioned. Typically, one adds a constant λ to the diagonals of $\mathcal{R}'[b]\mathcal{R}'[b]^*$. For large λ, we obtain the descent direction

$$\delta b = -\mathcal{R}'[b]^*\mathcal{R}[b],$$

which is scaled with respect to λ [36]. Plugging (2.10) into (2.9) results in

$$\mathcal{J}[b+\kappa\delta b] = \mathcal{J}[b] + \kappa \mathrm{Re}\left\langle \mathcal{R}'[b]^*\mathcal{R}[b], \delta b\right\rangle_P + O(||\kappa\delta b||_P^2),$$

which, using the descent direction, reduces to

$$\mathcal{J}[b+\kappa\delta b] = \mathcal{J}[b] - \kappa||\mathcal{R}'[b]^*\mathcal{R}[b]||_P^2 + O(||\kappa\delta b||_P^2),$$

meaning that δb reduces the cost functional because $||\cdot||_P^2 > 0$ and κ is chosen positive. To reduce $\mathcal{J}$ in practice, we must find an explicit form of $\mathcal{R}'[b]^*\mathcal{R}[b]$. In the general setting, we wish to find a representation of $\mathcal{R}'[b]^*\xi$ for some $\xi \in Z$. The cost functional itself is an implicit function of the conductivity through the residual operator, meaning any perturbation in the cost functional results in a perturbation to the underlying Maxwell PDE. With this deduction, we can relate the linearized residual operator to a measure of the solution to a perturbed Maxwell problem. We begin by perturbing the conductivity, which generates secondary electromagnetic fields as a result. A perturbation in parameter space leads to a perturbation in the state space, due to uniqueness of the Maxwell system, given that $\mathrm{Re}(b) > 0$ [36]. Mathematically speaking

$$b \mapsto b + \delta b, \implies \mathbf{E} \mapsto \mathbf{E} + \delta\mathbf{E},\ \mathbf{H} \mapsto \mathbf{H} + \delta\mathbf{H}.$$

Applying these mappings in (2.1) generates

$$\Lambda(a_0, b + \delta b)(\mathbf{u}_j + \delta\mathbf{u}_j) = \mathbf{q}_j. \tag{2.14}$$

We first linearize (2.14) and pick out quantities which are equivalent to the linearized residual term. Without loss of generality, let us assume that a single source $\mathbf{q}$ located at $\mathbf{s}$ is given as

$$\mathbf{q} = \begin{pmatrix} \mathbf{0} \\ \mathbf{m}_s\delta(\mathbf{x} - \mathbf{s}) \end{pmatrix},$$

where $\mathbf{m}_s = \sum_{i=x,y,z} a_i\mathbf{e}_i, a_i \in \mathbb{C}$ and let $\hat{a}_i := D_M a_i$. Then, denote $\mathbf{u}$ to be the solution of the *forward problem*

$$\Lambda(a_0, b)\mathbf{u} = \begin{pmatrix} \mathbf{0} \\ \mathbf{m}_s\delta(\mathbf{x} - \mathbf{s}) \end{pmatrix}. \tag{2.15}$$

Linearizing the system in (2.15) results in

$$\Lambda(a_0, b)\,[\mathbf{u} + \delta\mathbf{u}] = \begin{pmatrix} \mathbf{0} \\ \mathbf{m}_s\delta(\mathbf{x} - \mathbf{s}) \end{pmatrix} + \begin{pmatrix} \delta b\mathbf{E} \\ \mathbf{0} \end{pmatrix}. \tag{2.16}$$

We know how the residual operator in (2.5) relates to the system in (2.1), so how does the perturbed linearized residual operator relate to the expression in (2.16)? We can find out by using Taylor's Theorem to find an expression for the linearized residual operator. From (2.10) and (2.16), we know that

$$\mathcal{R}'[b]\delta b = \mathcal{M}\delta\mathbf{u}, \tag{2.17}$$

where $\delta\mathbf{u}$ solves

$$\Lambda(a_0, b)\delta\mathbf{u} = \begin{pmatrix} \delta b\mathbf{E} \\ \mathbf{0} \end{pmatrix}. \tag{2.18}$$

Before we find a useful form of $\mathcal{R}'[b]\delta b$, we will firstly have some discussion on $\mathcal{M}\delta\mathbf{u}$. By linearity of Maxwell's equations, we can say that

$$\delta\mathbf{u} = \mathbf{u}_t(\mathbf{x}; b+\delta b) - \mathbf{u}(\mathbf{x}; b),$$

where the subscript t denotes the total electromagnetic field. It follows that

$$\mathcal{R}'[b]\delta b = \mathcal{M}(\mathbf{u}^t - \mathbf{u}).$$

By linearity of the measurement operator, we obtain

$$\mathcal{M}(\mathbf{u}^t - \mathbf{u}) = \mathcal{M}\mathbf{u}^{(t)} - \mathcal{M}\mathbf{u}. \tag{2.19}$$

The relation in (2.19) gives us a better notion of the electromagnetic data. In particular, it allows us to interpret the right hand side of (2.17) as a difference measurement. We can interpret $M\mathbf{u}$ as a measurement on a state $\mathbf{u}$ which is generated by a known parameter b. In addition, $M\mathbf{u}^{(t)}$ can be interpreted as the measurement on a total state $\mathbf{u}^{(t)}$ which is generated by an unknown parameter distribution $b+\delta b$, i.e. a domain with an unknown conductivity distribution. It follows then, that the difference between the two is a difference measurement of the electromagnetic fields. To demonstrate this interpretation, and without loss of generality (i.e. multiple sources and receivers), figure 2.1 shows an example setup of one source and receiver.

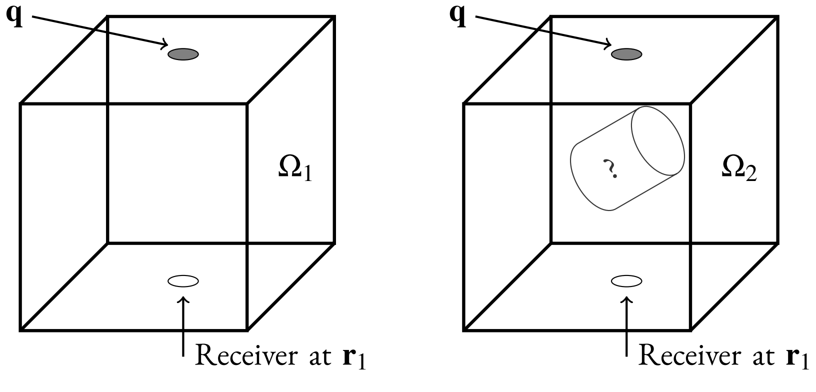


Figure 2.1: A qualitative description of measuring electromagnetic fields on Ω with and without an unknown object present.

In particular, we have the following representation of the parameter distribution b:

$$b(\mathbf{x}) \quad = \quad \begin{cases} b_0(\mathbf{x}) & \mathbf{x} \in \Omega_1, \\ b_0(\mathbf{x}) + \delta b(\mathbf{x}) & \mathbf{x} \in \Omega_2. \end{cases} \tag{2.20}$$

Each value of b described in (2.20) generates an electromagnetic field solution in the Maxwell PDE, whose state can then be realized at a receiver location by applying the measurement operator. For example, we could take a pointwise measurement as follows:

$$\begin{aligned} \mathcal{M}\mathbf{u} &= \int_{\Omega_1} \mathbf{H}_0(\mathbf{x})\delta(\mathbf{x}-\mathbf{r}_1)\, d\mathbf{x}; \\ &= \mathbf{H}_0(\mathbf{r}_1); \qquad (2.21a) \\ \mathcal{M}\mathbf{u}^{(t)} &= \int_{\Omega_2} \mathbf{H}(\mathbf{x})\delta(\mathbf{x}-\mathbf{r}_1)\, d\mathbf{x}; \\ &= \mathbf{H}(\mathbf{r}_1), \qquad (2.21b) \end{aligned}$$

which generalizes to the following:

$$(\mathcal{M}\mathbf{u}^{(t)} - \mathcal{M}\mathbf{u})_k = \left(\mathbf{H}^{(t)}(\mathbf{r}_k) - \mathbf{H}(\mathbf{r}_k)\right) \quad \text{for} \quad k = 1, 2, \ldots, n_r,$$

where quantities $\mathbf{H}^{(t)}(\mathbf{r}_k)$ can be interpreted as a realisation of the solution to (2.14) at $\mathbf{x} = \mathbf{r}_k$, whereas $\mathbf{H}(\mathbf{r}_k)$ can be viewed as a realisation of the solution to (2.15) at $\mathbf{x} = \mathbf{r}_k$. Note that the background conductivity b_0 can take many forms, and is typically a known quantity. For example, in applications such as cargo container screening, one envisages $b = b_{cc} \in \Omega_1$ being the conductivity for an empty cargo container, with $b^* = (b_{cc} + \delta b) \in \Omega_2$ being the conductivity of a cargo container with unknown contents. The measurement operator can take many forms. For the numerical experiments considered in this thesis, we use a different form of $\mathcal{M}\mathbf{u}$. Instead, we use:

$$\left[\mathcal{M}_j \mathbf{u}_j\right]_k = \oint_{S_k} \mathbf{E}_j \cdot \underline{\theta}\, dl, \tag{2.22}$$

for $k = 1, \ldots, n_r$, where $\mathbf{E}_j \cdot \underline{\theta}$ is the tangential component of $\mathbf{E}_j$ with respect to the closed wire loop S_k. The surfaces S_k act as receivers and are modelled as rectangular loops in the discrete setting. The measurement operator as defined in (2.22) represents the act of measuring an electric current. Notice that the additional regularity on $\mathbf{E}_j$ is no longer needed with this measurement.

Whether we are interested in finding b^* for the linearized inverse problem of Definition

2.2 or a full nonlinear inversion, we require an explicit form of the linearized residual operator and its adjoint. This leads us to recall some theoretical results found in [36].

Theorem 1

Let $\begin{pmatrix} \delta\mathbf{E} \\ \delta\mathbf{H} \end{pmatrix}, \begin{pmatrix} \mathcal{E} \\ \mathcal{H} \end{pmatrix} \in U$, then

$$\left\langle \Lambda(a,b) \begin{pmatrix} \delta\mathbf{E} \\ \delta\mathbf{H} \end{pmatrix}, \begin{pmatrix} \mathcal{E} \\ \mathcal{H} \end{pmatrix} \right\rangle_{\mathcal{L}^2(\Omega)} = \left\langle \begin{pmatrix} \delta\mathbf{E} \\ \delta\mathbf{H} \end{pmatrix}, \Lambda^*(a,b) \begin{pmatrix} \mathcal{E} \\ \mathcal{H} \end{pmatrix} \right\rangle_{\mathcal{L}^2(\Omega)},$$

where $\Omega = \mathbb{R}^3$.

Proof. Using Green's formulas for an infinite space, without boundary terms since $\mathrm{Re}(b) > 0$, we have

$$\begin{aligned} \int_\Omega (\nabla \times \delta\mathbf{E}) \cdot \overline{\mathcal{E}}\, d\mathbf{x} &= \int_\Omega \delta\mathbf{E} \cdot \overline{(\nabla \times \mathcal{E})}\, d\mathbf{x}; \\ \int_\Omega (\nabla \times \delta\mathbf{H}) \cdot \overline{\mathcal{H}}\, d\mathbf{x} &= \int_\Omega \delta\mathbf{H} \cdot \overline{(\nabla \times \mathcal{H})}\, d\mathbf{x}. \end{aligned}$$

Then,

$$\begin{aligned} \left\langle \Lambda(a,b) \begin{pmatrix} \delta\mathbf{E} \\ \delta\mathbf{H} \end{pmatrix}, \begin{pmatrix} \mathcal{E} \\ \mathcal{H} \end{pmatrix} \right\rangle_{\mathcal{L}^2(\Omega)} &= \int_\Omega \mathcal{D}_J \mathcal{D}_E (\nabla \times \delta\mathbf{H} - b\delta\mathbf{E}) \cdot \overline{\mathcal{E}}\, d\mathbf{x} \\ &+ \int_\Omega \mathcal{D}_M \mathcal{D}_H (\nabla \times \delta\mathbf{E} + a\delta\mathbf{H}) \cdot \overline{\mathcal{H}}\, d\mathbf{x}; \\ &= \int_\Omega \mathcal{D}_J \mathcal{D}_E \delta\mathbf{E} \cdot \overline{(\nabla \times \mathcal{H} - \overline{b}\mathcal{E})}\, d\mathbf{x} \\ &+ \int_\Omega \mathcal{D}_M \mathcal{D}_H \delta\mathbf{H} \cdot (\overline{\nabla \times \mathcal{E}} + \overline{a}\mathcal{H})\, d\mathbf{x}; \\ &= \left\langle \begin{pmatrix} \delta\mathbf{E} \\ \delta\mathbf{H} \end{pmatrix}, \Lambda^*(a,b) \begin{pmatrix} \mathcal{E} \\ \mathcal{H} \end{pmatrix} \right\rangle_{\mathcal{L}^2(\Omega)}. \blacksquare \end{aligned}$$

From (2.17), we have that

$$\begin{aligned} (\mathcal{R}'[b]\delta b)_k &= \int_\Omega \delta\mathbf{H}(\mathbf{x}) \delta(\mathbf{x} - \mathbf{r}_k)\, d\mathbf{x}; \\ &= \delta\mathbf{H}(\mathbf{r}_k), \end{aligned}$$

with the definition of the measurement operator as in (2.4). Namely, $\mathcal{R}'[b]\delta b \in Z$. By taking an inner product in data space Z, we obtain

$$\begin{aligned}
\left\langle \mathcal{R}'[b]\delta b, \xi \right\rangle_Z &= \mathcal{D}_H^2 \sum_{k=1}^{n_r} \sum_{i=x,y,z} \left(\int_\Omega \delta\mathbf{H}(\mathbf{x})\delta(\mathbf{x}-\mathbf{r}_k)d\mathbf{x} \right) \overline{h_k^{(i)}}\mathbf{e}_i; \\
&= \int_\Omega \mathcal{D}_H \mathcal{D}_M \delta\mathbf{H}(\mathbf{x}) \left[\overline{\mathcal{D}_H \mathcal{D}_M^{-1} \sum_{k=1}^{n_r} \sum_{i=x,y,z} h_k^{(i)} \delta(\mathbf{x}-\mathbf{r}_k)\mathbf{e}_i} \right] d\mathbf{x},
\end{aligned}$$

where $\xi = \left(\mathbf{h}_1, \ldots, \mathbf{h}_{n_r}\right)$ and $\mathbf{h}_k = (h_k^{(x)}, h_k^{(y)}, h_k^{(z)})$, with $\hat{h}_k^{(i)} := \mathcal{D}_H h_k^{(i)}$. The adjoint linearized residual operator is formally defined by the identity

$$\left\langle \mathcal{R}'[b]\delta b, \xi \right\rangle_Z = \left\langle \delta b, \mathcal{R}'[b]^* \xi \right\rangle_P. \tag{2.24}$$

The expression in (2.24) describes the relationship between the data space Z and parameter space P. Theorem 2 shows a decomposition of the quantities appearing in this expression, which can then be readily computed for minimization of the cost functional in (2.8) and regularized versions of it.

Theorem 2

Assume that we have some data $\xi = (\mathbf{h}_1, \ldots, \mathbf{h}_k) \in Z$ where $\mathbf{h}_k = (h_k^{(x)}, h_k^{(y)}, h_k^{(z)}) \in \mathbb{C}^3$. Then we denote the solution $\begin{pmatrix} \mathcal{E} \\ \mathcal{H} \end{pmatrix}$ of

$$\Lambda^*(a,b) \begin{pmatrix} \mathcal{E} \\ \mathcal{H} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathcal{D}_M^{-1} \sum_{k=1}^{n_r} \sum_{i=x,y,z} \hat{h}_k^{(i)} \delta(\mathbf{x}-\mathbf{r}_k) \end{pmatrix}, \tag{2.25}$$

as the adjoint solution, where

$$\Lambda^*(a,b) = \begin{pmatrix} -\overline{b} & \nabla\times \\ \nabla\times & \overline{a} \end{pmatrix}. \tag{2.26}$$

Then $\mathcal{R}'[b]^*\xi \in F$ is given by

$$\mathcal{R}'[b]^*\xi(\mathbf{x}) = \mathcal{D}_b^{-1} \mathbf{diag}\left(\overline{E^{(x)}}\mathcal{E}^{(x)}, \overline{E^{(y)}}\mathcal{E}^{(y)}, \overline{E^{(z)}}\mathcal{E}^{(z)} \right),$$

where $\mathbf{E}$ is a solution of (2.1).

Proof. Given the result in Theorem 1, coupled with the identities in (2.24), we can say that

$$\begin{aligned}&\left\langle \Lambda(a,b)\begin{pmatrix}\delta\mathbf{E}\\ \delta\mathbf{H}\end{pmatrix},\begin{pmatrix}\mathcal{E}\\ \mathcal{H}\end{pmatrix}\right\rangle_{\mathcal{L}^2(\Omega)}+\left\langle \mathcal{R}'[b]\delta b,\xi\right\rangle_Z=\\ &\left\langle \begin{pmatrix}\delta\mathbf{E}\\ \delta\mathbf{H}\end{pmatrix},\Lambda^*(a,b)\begin{pmatrix}\mathcal{E}\\ \mathcal{H}\end{pmatrix}\right\rangle_{\mathcal{L}^2(\Omega)}+\left\langle \mathcal{R}'[b]\delta b,\xi\right\rangle_Z.\end{aligned} \tag{2.27}$$

The second term on the left hand side cancels with the first time on the right in (2.27), given that $\begin{pmatrix}\mathcal{E}\\ \mathcal{H}\end{pmatrix}$ solves (2.25). Therefore, we are left with

$$\left\langle \Lambda(a,b)\begin{pmatrix}\delta\mathbf{E}\\ \delta\mathbf{H}\end{pmatrix},\begin{pmatrix}\mathcal{E}\\ \mathcal{H}\end{pmatrix}\right\rangle_{\mathcal{L}^2(\Omega)}=\left\langle \mathcal{R}'[b]\delta b,\xi\right\rangle_Z.$$

Given $\mathbf{q}\in Y$, $b\in P$ and $\mathbf{u}\in U$, let $\mathbf{u}=\begin{pmatrix}\mathbf{E}\\ \mathbf{H}\end{pmatrix}$ be a solution to the problem in (2.1), ignoring the j subscript. Similarly, let $\delta\mathbf{u}=\begin{pmatrix}\delta\mathbf{E}\\ \delta\mathbf{H}\end{pmatrix}$ be a solution to the perturbed problem in (2.18). Hence, with (2.17) and (2.18), (2.27) becomes

$$\begin{aligned}\left\langle \mathcal{R}'[b]\delta b,\xi\right\rangle_Z &= \int_\Omega \mathcal{D}_b\mathcal{D}_E^2\left(\delta b^{(x)}E^{(x)},\delta b^{(y)},\delta b^{(z)}E^{(z)}\right)\cdot\overline{\mathcal{E}}\,d\mathbf{x};\\ &= \int_\Omega \mathcal{D}_b\mathcal{D}_E^2\left[\sum_{i=x,y,z}\delta b^{(i)}\left(E^{(i)}\overline{\mathcal{E}^{(i)}}\right)\right]d\mathbf{x};\\ &= \left\langle \delta b,\mathcal{D}_b^{-1}\mathbf{diag}\left(\overline{E^{(x)}}\mathcal{E}^{(x)},\overline{E^{(y)}}\mathcal{E}^{(y)},\overline{E^{(z)}}\mathcal{E}^{(z)}\right)\right\rangle_P.\end{aligned} \tag{2.28}$$

Using the adjoint linearized operator identity defined in (2.24) results in

$$\mathcal{R}'[b]^*\xi(\mathbf{x})=\mathcal{D}_b^{-1}\mathbf{diag}\left(\overline{E^{(x)}}\mathcal{E}^{(x)},\overline{E^{(y)}}\mathcal{E}^{(y)},\overline{E^{(z)}}\mathcal{E}^{(z)}\right).\ [36]\blacksquare \tag{2.29}$$

We can also consider the general form for our application:

$$\left\langle \begin{pmatrix} \delta b \mathbf{E}_{j,m} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathcal{E}_{k,l} \\ \mathcal{H}_{k,l} \end{pmatrix} \right\rangle_{\mathcal{L}^2(\Omega)} = \left\langle \begin{pmatrix} \delta \mathbf{E}_{j,m} \\ \delta \mathbf{H}_{j,m} \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \mathcal{D}_M^{-1} e_l \delta(\mathbf{x} - \mathbf{r}_k) \end{pmatrix} \right\rangle_{\mathcal{L}^2(\Omega)} \tag{2.30}$$

whose quantities solve the following problems:

$$\begin{aligned} \Lambda(a_0, b) \begin{pmatrix} \delta \mathbf{E}_{j,m} \\ \delta \mathbf{H}_{j,m} \end{pmatrix} &= \begin{pmatrix} \delta b \mathbf{E}_{j,m} \\ \mathbf{0} \end{pmatrix}; \\ \Lambda^*(a_0, b) \begin{pmatrix} \mathcal{E}_{k,l} \\ \mathcal{H}_{k,l} \end{pmatrix} &= \begin{pmatrix} \mathbf{0} \\ \mathcal{D}_M^{-1} \mathbf{e}_l \delta(\mathbf{x} - \mathbf{r}_k) \end{pmatrix}; \\ \Lambda(a_0, b) \begin{pmatrix} \mathbf{E}_{j,m} \\ \mathbf{H}_{j,m} \end{pmatrix} &= \begin{pmatrix} \mathbf{0} \\ \mathcal{D}_M^{-1} \mathbf{e}_m \delta(\mathbf{x} - \mathbf{s}_j) \end{pmatrix}. \end{aligned}$$

Then, (2.30) has the explicit form

$$\int_\Omega \mathcal{D}_b \mathcal{D}_E^2 \mathbf{E}_{j,m}(\mathbf{x}) \delta b(\mathbf{x}) \overline{\mathcal{E}_{k,l}(\mathbf{x})} \, d\mathbf{x} = \int_\Omega \mathcal{D}_H \mathcal{D}_M \delta \mathbf{H}_{j,m}(\mathbf{x}) \cdot \mathcal{D}_M^{-1} \mathbf{e}_l \delta(\mathbf{x} - \mathbf{r}_k) \, d\mathbf{x}. \tag{2.32}$$

We know that

$$\delta \mathbf{H}_j(\mathbf{r}_k) = \sum_{m=x,y,z} \hat{a}_m \delta \mathbf{H}_{j,m}(\mathbf{r}_k). \tag{2.33}$$

Integrating the right hand side of (2.32) results in

$$\mathcal{D}_H \sum_{m=x,y,z} \hat{a}_m \delta \mathbf{H}_{j,m}(\mathbf{r}_k) \cdot \mathbf{e}_l = \mathcal{D}_b \mathcal{D}_E^2 \sum_{m=x,y,z} \hat{a}_m \int_\Omega \mathbf{E}_{j,m}(\mathbf{x}) \delta b(\mathbf{x}) \overline{\mathcal{E}_{k,l}(\mathbf{x})} \, d\mathbf{x}.$$

And so, by (2.33), we have that

$$\delta \mathbf{H}_m(\mathbf{r}_k) = \mathcal{D}_H^{-1} \mathcal{D}_b \mathcal{D}_E^2 \sum_{l=x,y,z} \left(\sum_{m=x,y,z} \hat{a}_m \int_\Omega \mathbf{E}_{j,m}(\mathbf{x}) \delta b(\mathbf{x}) \overline{\mathcal{E}_{k,l}(\mathbf{x})} \, d\mathbf{x} \right) \mathbf{e}_l.$$

With (2.17), we conclude that

$$(\mathcal{R}'[b]\delta b)_k = \mathcal{D}_H^{-1} \mathcal{D}_b \mathcal{D}_E^2 \sum_{l=x,y,z} \left(\sum_{m=x,y,z} \hat{a}_m \int_\Omega \mathbf{E}_{j,m}(\mathbf{x}) \delta b(\mathbf{x}) \overline{\mathcal{E}_{k,l}(\mathbf{x})} \, d\mathbf{x} \right) \mathbf{e}_l. \tag{2.34}$$

The relationship in (2.34) yields the decomposition of the linearized residual operator $\mathcal{R}'[b]\delta b$. With the linearized residual operator explicitly defined in (2.34), we now formulate the corresponding linearized inverse problem.

Consider a domain $\Omega \subset \mathbb{R}^3$ with n_s sources and n_r receivers such that position $\mathbf{s}_j$ is the location of the jth source and $\mathbf{r}_k$ is the location of the kth receiver, for $j = 1,2,\ldots,n_s$ and $k = 1,2,\ldots,n_r$. Then, ignoring dimension quantities, (2.34) takes the form

$$\sum_{j=1}^{n_s}\sum_{k=1}^{n_r}\sum_{m=x,y,z}\sum_{l=x,y,z}\sum_{n=1}^{n_v}\mathbf{E}_{j,m}(\mathbf{x}_n)\overline{\mathcal{E}_{k,l}(\mathbf{x}_n)}\delta b(\mathbf{x}_n) \quad = \quad \delta\mathbf{H}(\mathbf{r}_k),$$

where n_v is the number of voxels in the discretized domain. This summation can be written as $G\delta b = \delta\mathbf{H}(\mathbf{r}_k)$ where $G \in \mathbb{C}^{9n_sn_r\times n_v}$, $\delta b \in \mathbb{C}^{n_v}$ and $\delta\mathbf{H}(\mathbf{r}_k) \in \mathbb{C}^{9n_sn_r}$:

$$\begin{bmatrix} \sum_{n=1}^{n_v}\mathbf{E}_{x,1}(\mathbf{x_n})\overline{\mathcal{E}_{1,x}(\mathbf{x}_n)} \\ \sum_{n=1}^{n_v}\mathbf{E}_{x,1}(\mathbf{x_n})\overline{\mathcal{E}_{1,y}(\mathbf{x}_n)} \\ \sum_{n=1}^{n_v}\mathbf{E}_{x,1}(\mathbf{x_n})\overline{\mathcal{E}_{1,z}(\mathbf{x}_n)} \\ \sum_{n=1}^{n_v}\mathbf{E}_{y,1}(\mathbf{x_n})\overline{\mathcal{E}_{1,x}(\mathbf{x}_n)} \\ \vdots \\ \sum_{n=1}^{n_v}\mathbf{E}_{x,1}(\mathbf{x_n})\overline{\mathcal{E}_{2,x}(\mathbf{x}_n)} \\ \vdots \\ \sum_{n=1}^{n_v}\mathbf{E}_{z,1}(\mathbf{x_n})\overline{\mathcal{E}_{n_r,z}(\mathbf{x}_n)} \\ \sum_{n=1}^{n_v}\mathbf{E}_{z,2}(\mathbf{x_n})\overline{\mathcal{E}_{1,z}(\mathbf{x}_n)} \\ \vdots \\ \sum_{n=1}^{n_v}\mathbf{E}_{z,n_s}(\mathbf{x_n})\overline{\mathcal{E}_{n_r,z}(\mathbf{x}_n)} \end{bmatrix} \begin{bmatrix} \delta b(\mathbf{x}_1) \\ \vdots \\ \\ \\ \vdots \\ \delta b(\mathbf{x}_p) \\ \vdots \\ \\ \\ \vdots \\ \delta b(\mathbf{x}_{n_v}) \end{bmatrix} = \begin{bmatrix} \delta H^{(x)}{}_{x,x,1}(\mathbf{r}_1) \\ \delta H^{(y)}{}_{x,y,1}(\mathbf{r}_1) \\ \delta H^{(z)}{}_{x,z,1}(\mathbf{r}_1) \\ \delta H^{(x)}{}_{y,x,1}(\mathbf{r}_1) \\ \vdots \\ \delta H^{(x)}{}_{x,x,1}(\mathbf{r}_2) \\ \vdots \\ \delta H^{(z)}{}_{z,z,1}(\mathbf{r}_{n_r}) \\ \delta H^{(x)}{}_{x,x,2}(\mathbf{r}_1) \\ \vdots \\ \delta H^{(z)}{}_{z,z,n_s}(\mathbf{r}_{n_r}) \end{bmatrix}.$$

A schematic of the matrix equation $G\delta b = \delta\mathbf{H}(\mathbf{r}_k)$ shown above is for a discrete form of the linearized inverse problem, where each source and receiver can be directed in all three directions x, y, z. Size of the sensitivity matrix G is dictated by orientation and amount of sources and receivers, as well as the number of pixels employed in the discrete domain.

The sensitivity matrix can be created using electromagnetic field solutions which correspond to specific geometries or to the general problem as defined in this chapter. Some analytical methods, for example, are found by assuming specific knowledge of the conductivity in the domain. Solutions to these methods can be used to create the sensitivity matrix in the linearized inverse problem, though in an iterative nonlinear process where the conductivity is updated more than once, these methods break down as assumptions to generate the solutions are no longer valid.

The physical experiments described in [29, 30, 133] can be viewed mathematically as taking the sensitivity matrix to be the Identity matrix, in the act of difference imaging. In theory, using analytical methods which break down after one step, or numerical methods (which we will consider), should improve the results as they incorporate known physics of the problem. Whilst solving the linearized inverse problem could produce interesting results, we are more interested with the fully nonlinear inverse problem. Provided we have access to a robust Maxwell solver with respect to the conductivity and other material parameters if possible, we can iteratively solve the linearized inverse problem. This results in an nonlinear scheme which we build reconstruction algorithms for in this thesis. Both analytical and numerical methods for solving Maxwell's equations in the frequency domain will be discussed in chapter 3.

2.1

Regularization schemes

The problem in Definition 2.2 is the traditional $\mathcal{L}^2$-based conductivity inverse problem in MIT and has been solved for various applications. One, for example, is geophysical prospecting, see [36, 37] for more detail. Minimizers b^* of this traditional problem are $\mathcal{L}^2$, due to the conductivity being composed of a summation of $\mathcal{L}^2$ gradients. The inverse problem is highly ill-posed, meaning suitable regularization schemes need to be designed in order to find stable solutions to the problem. This section investigates and proposes various regularization schemes which address the challenging near-field electromagnetic imaging problem described in section 1.2. In particular, we consider variants of the traditional $\mathcal{L}^2$-based conductivity inverse problem which arise from applying certain regularization strategies. Whilst additional terms in the cost functional or implicit constraints on the conductivity are used to regularize the problem to try and capture desired properties or incorporate *a priori* knowledge into a minimizer, the original cost functional is also a form of regularization as we are assuming the conductivity lives in the $\mathcal{L}^2$ space. This assumption is valid though, as many physical phenomena have been observed to behave as $\mathcal{L}^2$ functions. The additional regularization schemes considered here can be classified into two distinct groups: 'pixel-based' and 'shape-based', as mentioned in chapter 1. These translate as explicit and implicit regularization terms on the conductivity respectively.

2.1.1
Sobolev smoothing

One smoothing tool, which we apply to all gradients arising from regularization schemes considered in this thesis, is the use of tailor-made function spaces. Recall that the formulation described in the adjoint problem maps into standard $\mathcal{L}^2$ spaces. This is sufficient and convenient in general, but for the schemes considered here, it will be beneficial to obtain gradient directions in a subset of smoother functions. In particular, such smooth gradient directions provide greater stability of the reconstruction schemes. Without loss of generality, we will derive the smooth gradient with respect to parameter b. The same methodology can be applied for gradients which result from different inversion schemes. Let us begin by introducing the following space:

$$W_1(\Omega) = \left\{ f : f \in L_2(\Omega), \nabla f \in L_2(\Omega), \frac{\partial f}{\partial n} = 0 \text{ on } \partial\Omega \right\},$$

with the equipped $W_1(\Omega)$ weighted inner product

$$\langle v, w \rangle_{W_1(\Omega)} = \alpha \langle v, w \rangle_{L_2(\Omega)} + \beta \langle \nabla v . \nabla w \rangle_{L_2(\Omega)} . \tag{2.35}$$

Here $\alpha \geq 1$ and $\beta > 0$ are suitably chosen regularization parameters. In the following we will derive a convenient mapping between the adjoint operator $\mathcal{R}_j'[b]^*$ and a new adjoint operator which is related to the space $W_1(\Omega)$. For this purpose, let $\xi_j \in Z_j$ and δb be a perturbation of $b \in W_1(\Omega)$. Then we have the following adjoint property:

$$\left\langle \mathcal{R}_j'[b]\delta b, \xi_j \right\rangle_{Z_j} = \left\langle \delta b, \mathcal{R}_j'[b]^* \xi_j \right\rangle_P . \tag{2.36}$$

To map $\mathcal{R}_j'[b]^*$ to a smoother space, we define a new adjoint $\widehat{\mathcal{R}_j'[b]}$ by

$$\left\langle \delta b, \mathcal{R}_j'[b]^* \xi_j \right\rangle_P = \left\langle \delta b, \widehat{\mathcal{R}_j'[b]} \xi_j \right\rangle_{W_1(\Omega)} . \tag{2.37}$$

We can expand the right hand side of (2.37) by using the weighted inner product defined in (2.35), resulting in

$$\left\langle \delta b, \widehat{\mathcal{R}_j'[b]} \xi_j \right\rangle_{W_1(\Omega)} = \alpha \left\langle \delta b, \mathcal{R}_j'[b]^* \xi_j \right\rangle_P + \beta \left\langle \nabla \delta b, \nabla \mathcal{R}_j'[b]^* \xi_j \right\rangle_P .$$

By Green's Identity, coupled with the Neumann boundary condition for $W_1(\Omega)$ functions,

$$\left\langle \nabla\delta b, \nabla\mathcal{R}_j'[b]^*\xi_j \right\rangle_P = -\left\langle \delta b, \Delta\mathcal{R}_j'[b]^*\xi_j \right\rangle_P.$$

Therefore,

$$\begin{aligned}\left\langle \delta b, \widehat{\mathcal{R}_j'[b]\xi_j} \right\rangle_{W_1(\Omega)} &= \alpha\left\langle \delta b, \mathcal{R}_j'[b]^*\xi_j \right\rangle_P - \beta\left\langle \delta b, \Delta\mathcal{R}_j'[b]^*\xi_j \right\rangle_P; \\ &= \left\langle \delta b, (\alpha I - \beta\Delta)\mathcal{R}_j'[b]^*\xi_j \right\rangle_P.\end{aligned}$$

Choosing $\xi_j = \mathcal{R}_j[b] \in Z_j$ gives us the formal gradient of the cost functional defined in (2.6), obtaining

$$(\alpha I - \beta\Delta)\widehat{\nabla\mathcal{J}_j[b]} = \nabla\mathcal{J}_j[b] \;\text{ in }\; \Omega, \tag{2.38}$$

where I and Δ are the Identity and the Laplacian operator respectively, and the hat above $\nabla\mathcal{J}_j[b]$ denotes that it has been smoothed. The relationship in (2.38) allows us to compute a smoothed gradient by a simple post-processing step after obtaining $\nabla\mathcal{J}_j[b]$. This smoothed gradient is also often referred to as a Sobolev gradient in the literature [91]. For more information on practical ways of calculating this Sobolev gradient, we refer the reader to a discussion in [48]. Across all regularization schemes, we project $\mathcal{L}^2$ gradients from the various cost functionals considered here, towards the smoother space $W_1(\Omega)$.

2.1.2

Pixel-based regularization

In this section, we develop two novel pixel-based reconstruction algorithms for the near-field electromagnetic imaging problem described in section 1.2. One is a modification of the traditional $\mathcal{L}^2$-based conductivity reconstruction scheme, and the other is an explicit modification of the original cost functional to encourage sparsity in the conductivity profile. In general, the pixel-based conductivity inverse problem can be written as:

$$b^* = \underset{b\in P}{\operatorname{argmin}} \left\{ \mathcal{J}[b] + \frac{1}{2}\sum_{j=1}^{n} \alpha_j M_j(b) \right\}, \tag{2.39}$$

where $\alpha \geq 0$ and $M_j(\cdot)$ are regularization terms. The terms $M_j(b)$ are specific penalties on the conductivity, whereas the first term is the data misfit term. The scalars α_j dictate how much weighting is given to a particular regularization term. Care must be taken upon choosing this, since too little may still lead to similar problems as the original scheme without regularization whereas too much may be solving an entirely different problem. Some popular choices of these regularization terms include total variation, sparsity, multi-bang, Mumford-Shah, Tikhonov, amongst many others. For more information on those regularization techniques, see [60, 62, 87, 125].

2.1.2.1

Traditional $\mathcal{L}^2$-based regularization

Recall from Definition 2.2 the traditional conductivity inverse problem:

$$b^* = \underset{b \in P}{\operatorname{argmin}}\, \mathcal{J}[b]. \tag{2.40}$$

This section derives an algorithm to solve (2.40). Recall that

$$\mathcal{J}[b] = \sum_{j=1}^{n_s} \mathcal{J}_j[b]; \quad \mathcal{J}_j[b] := \frac{1}{2}||\mathcal{R}_j[b]||^2_{\mathcal{L}^2(\Omega)}, \tag{2.41}$$

where we now redefine $\mathcal{R}_j[b]$ as:

$$\mathcal{R}_j[b] = \mathcal{F}_j[b] - \mathbf{d}_j,$$

where $\mathcal{F}_j$ are defined as forward operators

$$\mathcal{F}_j[b] = \mathcal{M}_j \mathbf{u}_j(\mathbf{x}; b).$$

The cost functional $\mathcal{J}[b]$ is useful for monitoring progress of any iterative estimation technique and for the design of suitable reconstruction schemes. Many popular minimization schemes use the formal gradient of (2.41):

$$\nabla \mathcal{J}[b] = \sum_{j=1}^{n_s} \nabla \mathcal{J}_j[b], \tag{2.42}$$

which can be calculated efficiently by the adjoint scheme described at the start of this chapter. Assuming sufficient regularity of $\mathcal{R}_j[b]$ in the chosen function spaces P and Z_j, let us denote the linearized residual operator by $\mathcal{R}_j'[b]$ and its adjoint operator with respect to these function spaces by $\mathcal{R}_j'[b]^*$. Let us choose some data $\xi_j = \mathcal{R}_j[b] \in Z_j$, $b \in P$ and let $(\mathcal{E}_j, \mathcal{H}_j)^T$ be the solution of the adjoint Maxwell problem:

$$\begin{pmatrix} -b & \nabla\times \\ \nabla\times & a \end{pmatrix} \begin{pmatrix} \mathcal{E}_j \\ \mathcal{H}_j \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathcal{M}_j^* \overline{\mathcal{R}_j[b]} \end{pmatrix},$$

where the overline denotes *complex conjugate* and $\mathcal{M}_j^*$ is the formal adjoint of the measurement operator. Its application amounts to putting the residuals as artificial adjoint sources at the receiver locations. Then,

$$\nabla \mathcal{J}_j[b] = \mathcal{R}_j'[b]^* \xi_j = \mathcal{R}_j'[b]^* \mathcal{R}_j[b] \tag{2.43}$$

is given by

$$\left[\mathcal{R}_j'[b]^* \mathcal{R}_j[b]\right](\mathbf{x}) = \overline{\mathbf{E}_j(\mathbf{x}) \cdot \mathcal{E}_j(\mathbf{x})}, \tag{2.44}$$

as analogously derived in (2.29). The interpretation of $\mathcal{R}_j[b]$ as an artificial adjoint source, upon which $\mathcal{R}_j'[b]^*$ is acting on in (2.44), gives rise to the interpretation of *back-projecting* the residual data to their (secondary) source.

The nonlinear inverse problem is of large scale in 3D. Numerically speaking, iterative techniques are employed to find solutions, requiring repeated calculation of descent directions to $\mathcal{J}[b]$ in (2.41). Most standard schemes require calculation of the full gradient $\nabla \mathcal{J}[b]$ in (2.42). Depending on the available forward solver this might consume considerable resources and processing time in each iteration. Therefore, alternative techniques have been developed in order to speed up the inversion process. One of those is the nonlinear Landweber-Kaczmarz (LK) scheme, which we employ here. The technique cycles over individual source positions in some order and only requires to calculate gradients of partial data sets in each step, which can be done efficiently with the adjoint method as outlined in this chapter. One complete cycle of the gradient set is called a sweep. With this scheme it is no longer the primary goal to directly reduce the data misfit to $\mathcal{J}[b]$, shown in (2.41), in each individual step. Instead, in each iteration, we reduce the data misfit $\mathcal{J}_j[b]$ with respect to the jth source only. In more detail, we use the single-step update formula

$$b_{k+1} = b_k - \tau \mathrm{Re}\left\{\widehat{\nabla \mathcal{J}_{[k]}[b_k]}\right\}, \tag{2.45}$$

where we follow, for simplicity, a sequential rule $[k] := (k \text{ modulo } n_s) \in \{0, 1, \ldots, n_s - 1\}$ and the hat above the gradient term denotes that the original $\mathcal{L}^2$ gradient has been projected into $W_1(\Omega)$ using (2.38). Alternative approaches, such as using a random selection of sources, are possible as well. Note that similar 'Stochastic Gradient Methods' have become popular in several large scale data sciences applications, which have attracted a lot of interest recently [13, 49]. We do not deal with random selection of gradient updates in this thesis, partly because we have small gradient sets (lowers the appeal of stochastic methods).

2.1.2.2

Pixel-based line search strategy for Kaczmarc iterations

This section introduces a new line search criteria which stops the iteration formula in (2.45) from over fitting to cost functionals $\mathcal{J}_j[b]$. Whereas in standard gradient based optimization schemes a line search is computed following well established rules and criteria [94], the task of finding an optimal step size in each step of Kaczmarz type approaches is not yet well understood. The reason being is that only part of the data is involved, such that large step sizes in this partial gradient direction might severely deteriorate the data misfit which has been achieved by previous updates from different parts of the data set. Therefore, one must be careful when choosing τ as this parameter dictates how much we reduce the cost functional with respect to one source only. Some standard choices include:

$$\begin{aligned} \tau &= c\frac{||b_k||_2^2}{||\nabla \mathcal{J}_{[k]}[b_k]||_2^2}; \\ \tau &= c, \end{aligned}$$

where $c \in \mathbb{R}^+$.

In this thesis, we insist that $\tau = \tau_k$, i.e. a function of the iteration number. Here, we propose a novel line search strategy for the pixel-based regularization schemes. The criteria is split into two parts; the first is used for estimating how the initial step size should be chosen, and the second is used for how the step size should be updated in subsequent iterations.

To start with, we consider a preprocessing step (involving a sufficiently large subset of the entire data set) which gives us a rough estimate of how τ_0, denoted as the initial step size, should be chosen. Then, we dynamically update τ_k in the following sweeps using a different criteria. For this purpose, we choose to calculate n_0 gradients when $b = b_0$, n_0 sufficiently large ($1 \leq n_0 \leq n_s$), and do a partial line search to find τ_0. Calculating these gradients involves solving Maxwell's equations $2n_0$ times (one forward and one adjoint solve per gradient). This is the computational cost for computing $\nabla \mathcal{J}^{n_0}[b_0]$, which is defined as:

$$\nabla \mathcal{J}^{(n_0)}[b_0] = \sum_{j \in J = \left\{j_1, j_2, \ldots, j_{n_0}\right\}} \nabla \mathcal{J}_j[b_0].$$

Whereas the succeeding Kaczmarz type approach in (2.45) usually only uses information from one single source position in one update, the first step in our scheme uses a subset of n_0 sources. The idea is to combine information from a significant subset of source positions to calculate a more reliable estimate of the gradient $\nabla \mathcal{J}[b_0]$, resulting in a more reliable initial step size τ_0. We usually would choose a quarter or half of all available sources, but we could use more or less if it seems reasonable. The choice of source indices can follow a random rule (as it is done here and is often done in stochastic gradient schemes [49]) or alternatively we can select source indices following a more systematic way. Once we have calculated $\nabla \mathcal{J}^{(n_0)}[b_0]$, we devise a method for computing τ_0 as follows. Instead of considering the cost functional that incorporates the entire data set, we consider in the initial step a reduced cost functional S_0 which only considers the chosen subset of n_0 source positions. In our line search we monitor behaviour of the cost with respect to various choices of τ. The cost function is defined as:

$$\mathcal{S}_0(\tau) = ||\mathcal{R}^{(n_0)}[b(\tau)]||^2_{\mathcal{L}^2(\Omega)}, \tag{2.46}$$

where

$$\mathcal{R}^{(n_0)}[b(\tau)] = \sum_{j\in J} \mathcal{R}_j[b(\tau)] \quad \text{and} \quad b(\tau) = b_0 - \tau \mathrm{Re}\left\{ \widehat{\nabla \mathcal{J}_{n_0}[b_0]} \right\}.$$

We now follow a variant of the so-called Brent method [99], which we adjust to the Kaczmarz type scheme. Let us define $I_\tau = [\tau_{min}, \tau_{max}]$ and denote $\tau^* \in I_\tau$ the minimizer of $\mathcal{S}_0(\tau)$ (assuming that there is only one in the given interval). We choose here

$$\tau_{max} = c_1 \frac{||b_0||_2^2}{||\nabla \mathcal{J}_{n_0}[b_0]||_2^2}, \ \tau_{min} = c_1 \tau_{max}, \tag{2.47}$$

where $c_1 \in (0, a), a \in \mathbb{R}^+$, but other choices are possible. Typically, a is chosen to be much less than unity so that we don't promote large τ (as this might cause problems in a Kaczmarz type scheme). Note that choosing $n = n_s$ retains our original cost functional and performs a traditional line search in this first step. In practice, a small fraction of n_s sources are sufficient for choosing τ_0. With the two step size quantities in (2.47) and an additional term τ_{mid} (defined as the middle point of the interval I_τ), we evaluate (2.46) for each value of τ and choose which to use in the initial inversion step by the criterion in (2.48):

$$\tau_0 = \begin{cases} \tau_{min} & \text{if } S_0(\tau_{min}) < S_0(\tau_{mid}) \text{ and } S_0(\tau_{max}). \\ \tau^q_{mid} & \text{if } S_0(\tau_{mid}) < S_0(\tau_{min}) \text{ and } S_0(\tau_{max}) \text{ where} \\ & \tau^q_{mid} \text{ is the minimizer of a quadratic polynomial} \\ & \phi^q(\tau) = a\tau^2 + b\tau + c \text{ fitted to } \tau_{min}, \tau_{mid}, \tau_{max}. \\ \tau_{max} & \text{if } S_0(\tau_{max}) < S_0(\tau_{min}) \text{ and } S_0(\tau_{mid}). \end{cases} \tag{2.48}$$

Once τ_0 has been chosen, we proceed with the iteration formula and update the step size dynamically after each sweep. It is updated in the following way:

$$\tau_s = \begin{cases} \tau_{s-1} & if \mathcal{J}[b_{s-1}] - \mathcal{J}[b_{s-2}] < 0; \\ (1-c_1)\tau_{s-1} & if \mathcal{J}[b_{s-1}] - \mathcal{J}[b_{s-2}] > 0, \end{cases} \tag{2.49}$$

where in this case τ_s denotes the step size to be used for sweep s, such that $s \geq 2$, $\tau_1 = \tau_0$ and $c_1 \in [0.5, 1)$. For example, with this notation, $\mathcal{J}[b_1]$ represents the cost quantity which is calculated after updating the conductivity using each gradient once. This dynamic correction helps the algorithm stabilize as it tries to stop the reconstruction scheme from overestimating. Practically, we approximate the cost; as will be discussed in chapter 4. Algorithm 1 presents the complete $\mathcal{L}^2$-based conductivity reconstruction scheme with the novel line search criteria and Sobolev smoothing. We will demonstrate this scheme in practice when we compute numerical experiments in chapter 4.

Algorithm 1 Pixel-based LK reconstruction with dynamic line search and Sobolev smoothing

```
procedure INITIALIZATION
    Choose $n_s, n_r, f, S$, source vectors $\mathbf{q}_j$ and $b_0$
    Collect $\nabla\mathcal{J}^{(n_0)}[b_0] = \sum_{j\in\{j_1,j_2,\dots,j_{n_0}\}} \nabla\mathcal{J}_j[b_0]$
    Initial Line Search Scheme as outlined in section 2.1.2.2:
        Calculate $\tau_{min}, \tau_{mid}, \tau_{max}$ using (2.47)
        Compute $b_1(\tau)$ for $\tau = \tau_{min}, \tau_{mid}, \tau_{max}$ using (2.45)
        Choose $\tau_0$ using (2.48)
    End of Initial Line Search
procedure LK RECONSTRUCTION
    for $k = 0 : Sn_s - 1$ do
        Calculate $\nabla\mathcal{J}_{[k]}[b_k]$ using (2.43)
        Compute smoothed gradient $\widehat{\nabla\mathcal{J}_{[k]}[b_k]}$ by solving (2.38)
        Update $b_{k+1} \leftarrow b_k - \tau_k \mathrm{Re}\left\{\widehat{\nabla\mathcal{J}_{[k]}[b_k]}\right\}$
        If applicable, update step size for next sweep using criteria in (2.49)
```

2.1.2.3

Sparsity promoting regularization

The traditional $\mathcal{L}^2$-based scheme shown in algorithm 1 assumes the minimizer belongs in $\mathcal{L}^2$ and also employs implicit regularization through Sobolev smoothing, such that the minimizer is projected towards $W_1(\Omega)$. This results in a solution to the corresponding optimization problem which is smoother, since updating the conductivity profile involves subtracting some multiple of these smoothed gradients. Whereas this helps stabilizing the inversion process, it also tends to deliver quite oversmoothed conductivity profiles. This oversmoothing property of pixel-based inversion schemes has also been observed when not using Sobolev gradients, but standard $\mathcal{L}^2$ gradients for the inversion [36, 37]. Therefore, in situations where it is known or expected that the unknown part of the conductivity distribution has compact support, the traditional $\mathcal{L}^2$-based approach does not seem to be optimal. Instead, one can try incorporate *a priori* knowledge on expected discontinuities of medium parameters into the inversion model. Recently, a sparsity promoting regularization scheme has been suggested for similar situations using other imaging modalities such as electrical impedance tomography or optical tomography, see for example [67, 68, 100]. We will develop a sparsity promoting scheme for the challenging near-field electromagnetic imaging problem described in section 1.2.

In the sparsity promoting regularization scheme we modify the standard cost functional from (2.41) by adding an additional term in the form of an l^1 penalty, such that contributions in the parameter space, which are considered small in a predefined sense, are penalized. Sparsity can be considered in several ways, with respect to a variety of basis systems or over-complete dictionaries. We will consider sparsity with respect to the standard voxel basis, which favours reconstructions with compact support. Such a process can be computed practically by using so-called shrinkage operators [31].

Let us decompose $b = b_0 + \varepsilon_b$ where b_0 denotes an approximately known background and ε_b the unknown inclusions. The sparsity inverse problem can then be written as follows:

$$b^* = \underset{b}{\operatorname{argmin}}\, \hat{\mathcal{J}}[b],$$

where

$$\hat{\mathcal{J}}[b] = \mathcal{J}[b] + \frac{\alpha}{2}||\varepsilon_b||_1 \tag{2.50}$$

with an assumption that

$$||\varepsilon_b||_1 = \sum_k |\langle \varepsilon_b, \psi_k \rangle|,$$

and $\{\psi_k\}$ is assumed to be an orthonormal basis, but in other applications it could be an over-complete dictionary. Due to the $||\varepsilon_b||_1$ term, the cost functional in (2.50) is no longer differentiable. Following suggestions in [31, 67, 68] and results shown in [100], we perform the following steps to promote sparsity in the conductivity reconstruction:

1. Compute gradient $\nabla \mathcal{J}[b_k]$;
2. Using step 1 and (2.38) to calculate $\widehat{\nabla \mathcal{J}[b_k]}$;
3. Apply shrinkage on $\varepsilon_{b_{k+1}} = \varepsilon_{b_k} - \tau \mathrm{Re}\left\{\widehat{\nabla \mathcal{J}[b_k]}\right\}$.

These three steps can be combined to write

$$\varepsilon_{b_{k+1}} = S_{\tau_k \gamma}\left(\varepsilon_{b_k} - \tau \mathrm{Re}\left\{\widehat{\nabla \mathcal{J}[b_k]}\right)\right\},$$

where

$$S_{\tau\gamma}(x)_i = \begin{cases} (|x_i| - \tau\gamma)\mathrm{sign}(x_i) & \text{if } |x_i| > \tau\gamma \\ 0 & \text{otherwise.} \end{cases} \tag{2.51}$$

The sparsity promotion is therefore embedded in the shrinkage operator, as this influences how the gradient descent behaves in the next iteration. Whilst we could follow steps (1-3) above with access to the full gradient, we instead consider sparsity with respect to each entry of the full gradient; such that it is in line with the LK scheme. When initializing τ, we perform the same line search technique as that in section 2.1.2.2. In contrast to the traditional $\mathcal{L}^2$-based inversion, however, we now incorporate the thresholding parameter $\Lambda = \tau\gamma$, which is defined in (2.51). This parameter controls the intended sparsity level of the conductivity reconstruction. Here, we choose

$$\Lambda = d_1 \max(|\varepsilon_{b_1}|), \tag{2.52}$$

although other choices such as $\Lambda = \Lambda(k)$ are possible. Note that

$$\varepsilon_{b_1} = \varepsilon_{b_0} - \tau_0 \mathrm{Re}\left\{\widehat{\nabla \mathcal{J}_0[b_0]}\right\} \tag{2.53}$$

and $d_1 \in (0, y_2)$, where y_2 is typically unity, but other choices are possible. Our assumption on the thresholding term is that the maximum value of ε_{b_1} has a similar order of magnitude as ε_{b_k}, for $k > 1$. Furthermore, d_1 can be interpreted as a parameter that chooses the level of zero filling in the reconstruction.

Algorithm 2 Pixel-based LK-Sparsity reconstruction with dynamic line search and Sobolev smoothing

```
procedure INITIALIZATION
    Choose $n_s, n_r, f, S$, source vectors $\mathbf{q}_j$ and $b_0$
    Choose $\tau_0$ using (2.48)
procedure LK-SPARSITY RECONSTRUCTION
    for $k = 0 : Sn_s - 1$ do
        Calculate $\nabla \mathcal{J}_{[k]}[b_k]$ using (2.43)
        Compute smoothed gradient $\widehat{\nabla \mathcal{J}_{[k]}[b_k]}$ by solving (2.38)
        if $k = 0$ then
            Compute $\varepsilon_{b_1}$ using (2.53)
            Choose $\Lambda$ using (2.52) and use for $k > 1$
        Compute $\varepsilon_{b_{k+1}} = S_{\tau_0\Lambda}\left(\varepsilon_{b_k} - \tau_0 \mathrm{Re}\left\{\widehat{\nabla \mathcal{J}_{[k]}[b_k]}\right\}\right)$
        Update $b_{k+1} \leftarrow b_k + \varepsilon_{b_{k+1}}$
```

The complete sparsity reconstruction scheme using the LK optimizer with Sobolev smoothing and the novel line search criteria is presented in algorithm 2. We demonstrate this scheme in practice when computing numerical experiments in chapter 4.

2.1.3

Shape-based regularization

The sparsity promoting scheme proposed in the previous section is useful when inclusions with compact support or specific shape embedded in some approximately known background are sought after. A similar effect has been observed in a variety of applications using a level set based approach, which directly uses a shape-based model for the unknown inclusions. Originally, level set methods were developed by Osher and Sethian as a means for tracking propagation of interfaces in image processing and computational physics [97, 107]. This led to Santosa applying their ideas in the arena of Inverse Problems [105]. Since then, level set methods for nonlinear inverse problems have been applied across a whole range of imaging modalities, see [17, 39, 41] for recent reviews.

The key difference between the level set shape-based approach and the pixel-based schemes described in algorithms 1 and 2 is that some estimate of the true conductivity value is assumed known *a priori* in the level set shape-based approach. Whilst it is true that the sparsity regularizer looks for sparse solutions, it does not have information on conductivities present when the true electromagnetic data measurements are taken; it looks for a solution that fits the data in a certain form. Shape-based schemes are distinctly different however, as they alter the problem from being one of recovering pixel value to assuming the pixel value but not how it is distributed in the imaging domain. As anticipated, the additional information significantly improves the reconstruction of topology and shape of unknown inclusions in this low frequency regime. This is even the case when *a priori* information on the conductivity is over or under estimated [58] (also shown in chapter 4). Therefore, in many practical applications, shape-based methods are a useful approach when interested in detecting and estimating general shapes of unknown inclusions from few data. Moreover, if required, estimation of internal parameter values can be made part of the optimization problem such that incorrect *a priori* information can be corrected during the optimization procedure. We do not include such an additional optimization loop in the algorithm proposed here, but instead will test the shape-based algorithms against incorrectly estimated internal conductivity values. This will be discussed in greater detail in chapter 4.

The set of problems in (2.39) do not adequately describe the shape-based inversion under discussion here, as these shape-based problems have *a priori* assumptions on b from the outset, rather than incorporating information into explicit regularization terms, which through certain optimization techniques recover specific conductivity profiles. Instead, we introduce a new set of problems which we wish to solve. These are:

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \left\{ \mathcal{J}[b(\Phi)] + \frac{1}{2} \sum_{j=1}^{n} \alpha_j M_j(\Phi) \right\}, \tag{2.54}$$

where Φ can be a vector of level set functions, depending on the assumptions made on b. The shape-based approach will become clear as we discuss the algorithms used to solve specific forms of (2.54).

2.1.3.1

Single level set regularization

In this section, we introduce a level set representation for a shape-based inversion approach where the conductivity profile is assumed to be composed of two known values. Throughout this section and subsequent sections of the shape-based approaches, we consider no explicit regularization terms. We focus on implicit regularization terms which result from *a priori* information on the conductivity profile. Therefore, we are interested in solving problems of the form:

$$\phi^* = \underset{\phi}{\operatorname{argmin}} \, \mathcal{J}[b(\phi)].$$

To discuss the technical implications of such a level set based shape evolution approach, let us begin by introducing a sufficiently smooth level set function $\phi\colon \Omega \to \mathbb{R}$, which defines a shape S by

$$b(\phi)(\mathbf{x}) = \begin{cases} b^i(\mathbf{x}) & \text{in } S \text{ where } \phi(\mathbf{x}) \le 0 \\ b^e(\mathbf{x}) & \text{in } \Omega \backslash S \text{ where } \phi(\mathbf{x}) > 0, \end{cases} \tag{2.55}$$

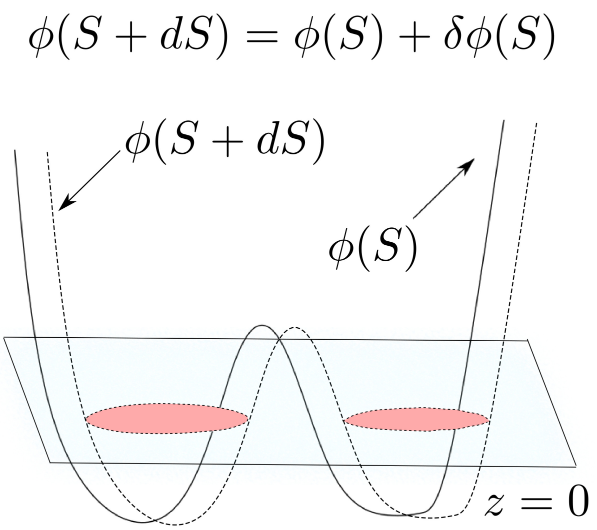


Figure 2.2: Visualization of the single level set representation defined in (2.55).

where b^i and b^e are the interior and exterior conductivities respectively, and are either constant or smoothly varying. We note again that the imaginary component of b is constant throughout the region, hence its omission. The boundary ∂S is given by the zero level set $\partial S = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}) = 0\}$. The representation in (2.55) can be written in the convenient form

$$b(\phi) = b^i(1 - H(\phi)) + b^e H(\phi),$$

where H is the Heaviside function. In level set inversion, we are interested in finding a suitable ϕ such that the data misfit is minimized. For mathematical ease, we redefine the cost functional for this shape-based approach. Therefore, we let

$$\mathcal{T}[\phi] = \mathcal{R}[b(\phi)]\ ,\ \mathcal{J}[\phi] = ||\mathcal{T}[\phi]||^2_{\mathcal{L}^2(\Omega)},$$

which yields the formal gradient

$$\nabla \mathcal{J}[\phi] = \sum_{j=1}^{n_s} \nabla \mathcal{J}_j[\phi].$$

From (2.43, 2.44) we obtain the analog

$$\nabla \mathcal{J}_j[\phi](\mathbf{x}) = \mathcal{T}_j'[\phi]^* \mathcal{T}_j[\phi](\mathbf{x}) = \mathrm{Re}\{b'(\phi)\overline{\mathbf{E}_j(\mathbf{x}) \cdot \mathcal{E}_j(\mathbf{x})}\}, \tag{2.56}$$

where

$$b'(\phi) = (b^e - b^i)\delta(\phi)$$

and $\delta(\cdot)$ is the Dirac Delta distribution. In the advent of numerical experiments, such a distribution has to be approximated by a function in a suitable space. Therefore, we choose a narrowband function [97, 107], such that $\delta(\phi) \approx \chi_{B_d(\Gamma)}$, where

$$B_d(\Gamma) = \{\mathbf{x} : ||\mathbf{x} - \Gamma||_2^2 \leq d/2\},$$

$d > 0$ and χ_D is the characteristic function. In general, we assume that boundaries of objects are smooth. Therefore, like the pixel-based algorithms, one can map the gradient to a smoother space using Sobolev projection. In the level set reconstruction scheme, we also employ the LK optimizer for minimization. It follows that

$$\phi_{k+1} = \phi_k - \tau \widehat{\nabla \mathcal{J}_{[k]}[\phi_k]}, \tag{2.57}$$

where τ is a step size to be discussed. The initial level set function ϕ_0 is typically chosen to be a signed distance function, representing some arbitrary initial shape. Typically, a level set representation of an ellipsoid or sphere in the centre of Ω suffices in simple geometries, such as those considered here. Once this initial shape has been assigned, we can proceed with the iterative scheme in (2.57) for $k \geq 1$. Better adapted initial guesses can speed up the reconstruction, though, and for that reason should be used whenever available. It is possible, for instance, that performing a few steps of algorithms 1 or 2 may given some indication on where to place our initial guess.

As in the two LK pixel-based reconstruction schemes, an efficient line search is vital for success in this shape-based scheme. It turns out that for shape-based schemes a simple yet efficient line search can be designed which does not require any additional forward solves. The line search scheme, as presented in section 2.1.3.2, is used for both single and multi-level set inversion, where the latter is introduced in section 2.1.3.3.

2.1.3.2

Level set line search criteria for Kaczmarz iterations

In shape-based inversion, choosing the step size τ is tricky. It is currently not understood how exactly the number of voxels that change per update relates to the cost functional. Nevertheless, we can make assumptions on how the level set should evolve. As in the pixel-based schemes, we must choose the step size sufficiently small such that large updates are not made with respect to individual sources as we don't want to overfit. Whilst ideas of the step size criterion in section 2.1.2.2 could still apply here, we choose the step size by a different metric.

Unlike pixel-based methods, from one iteration to the next, a finite amount of voxels change hands from interior to exterior conductivity values, and vice-versa, in these shape-based regimes. Here we create a criteria which keeps track of the number of voxels which change hands and adjust the step size accordingly, such that a smooth evolution of the level set function is guaranteed. We introduce the line search criteria for *color* level set inversion (involving multiple level set functions) so that it is compatible for both single and multi level set inversion. The scheme is used to choose a step size vector $\underline{\tau}_s = [\tau_s^1, \tau_s^2, \dots, \tau_s^N]$ in the update formula

$$\Phi^s_{j+1} = \Phi^s_j - \underline{\tau}_s \circ \nabla \mathcal{J}_j[\Phi^s_j], \tag{2.58}$$

where s denotes the sweep number. We choose the step size vector in (2.58) to have the form

$$\underline{\tau}_s = \mathbf{f}(\mathbf{w}_s \circ \underline{\tau}_0),$$

where $\mathbf{f}(\cdot)$ is the vector-valued form of the backtracking scheme given in algorithm 3, $\mathbf{w}_s = [w_s^1, w_s^2, \dots, w_s^N]$ is a vector of scalars which are adjusted at the end of each sweep and $\underline{\tau}_0$ is a vector of initialized step sizes for each level set function in Φ. Hence, we arrive at the update formula:

$$\Phi^s_{j+1} = \Phi^s_j - \mathbf{f}(\mathbf{w}_s \circ \underline{\tau}_0) \circ \nabla \mathcal{J}_j[\Phi^s_j]. \tag{2.59}$$

The step size in (2.59) has two components: the first is applied every update ($\mathbf{f}$) and the second is a dynamic scheme which alters the input of $\mathbf{f}$ at the end of each sweep.

Algorithm 3 Backtracking algorithm for one level set function

```
procedure F($\tau$, $\phi$, $g$,$N_{sup}$)
    Initialize $m \in (0,1)$ and $I \in \mathbb{Z}^{>1}$
    for $i = 0 : I - 1$ do
        $\tau = m\tau$
        $\phi_t = \phi - \tau g$
        Count voxel change $V$ in $x = \phi_t - \phi$
        if $V < N_{sup}$ then
            return $\tau$
    return 0
```

To begin, we discuss the first component of the line search scheme. Here, the idea is to choose each entry in $\underline{\tau}_s$ such that N_j^m (the number of voxels that change per jth source for the mth level set function, $j = 0, 1, \dots, n_s - 1$) is contained in an interval deemed suitable for the shape associated with the mth level set function to evolve smoothly. The relationship between N_j^m and the data misfit is of interest, but as of present is unclear. Although the relationship is unknown, intuition tells us that if we only allow a small amount of voxels to change per update, then we ensure a smooth evolution of the shape which hopefully induces a smooth evolution of the cost. We choose $N_j^m \in I_{target}^m$, where $I_{target}^m = [N_{inf}^m, N_{sup}^m]$ is chosen appropriately. Typically, $N_{inf}^m = 0$ and N_{sup}^m is chosen as a small percentage of the total number of voxels in the domain. This quantity is a metric on how much each level set function evolves per sweep. Although the backtracking line search ensures updates are only made if $N_j^m \in I_{target}^m$, it does not ensure an unequal weighting to each sensitivity pattern generated by the jth source. Ideally, we would like to give greater weighting to those sources which capture objects present in their sensitivity patterns. For example, if we initialize the pth entry in $\underline{\tau}_s$ such that an update to the pth level set function is large, we will consistently accept a voxel change just below N_{sup}^p for all sources. This introduces two problems; the first is that more iterations of the backtracking scheme will have to be computed in order to reduce τ_p^0 such that $N_{sup}^p \in I_{target}^p$ and the second is that we will give equal weighting to 'bad updates' in the Kaczmarz scheme. To alleviate this problem, we design criteria involving the quantity N_j^m. Once a sweep is complete (i.e. we make an Kaczmarz update for each source), we compute the average number of voxels that have changed for that sweep. Mathematically speaking

$$\overline{N}^m = \frac{1}{n_s} \sum_{j=1}^{n_s} N_j^m. \tag{2.60}$$

This brings us to the second component of the line search criteria. We adjust the input space of $\mathbf{f}$ by using a weighting vector $\mathbf{w}_s$, which is chosen by a criterion involving (2.60). Let us now introduce the subinterval $I_{target}^{\prime m} = \left[N_{low}^m, N_{high}^m\right]$, whose bounds are defined as:

$$\begin{aligned} N_{high}^m &= (1 - d_m) N_{sup}^m; \qquad (2.61) \\ N_{low}^m &= d_m N_{sup}, \qquad (2.62) \end{aligned}$$

where $d_m \in \left(0, \frac{1}{2}\right)$. These will be included in the criteria for updating entries in $\mathbf{w}_s$. From this, we can deduce that

$$N_{inf}^m < N_{low}^m < N_{high}^m < N_{sup}^m.$$

We adjust each entry in $\mathbf{w}_s$ according to a criterion involving (2.60, 2.61, 2.62). For brevity, we show how the pth component of $\mathbf{w}_s$ is updated per sweep:

$$w_s^p = \begin{cases} k_s^{p,1} w_{s-1}^p & \text{if } \overline{N}_{s-1}^p > N_{high}^p \text{ where } k_s^{p,1} \in [0.5, 1) \\ k_s^{p,2} w_{s-1}^p & \text{if } \overline{N}_{s-1}^p < N_{low}^p \text{ where } k_s^{p,2} \in (1, 2] \\ w_{s-1}^p & \text{if } N_{low}^p \leq \overline{N}_{s-1}^p \leq N_{high}^p, \end{cases} \tag{2.63}$$

where s denotes the quantity for sweep number s, $s \geq 1$. In other words, if the step size associated with the pth level set function is too large with respect to the average voxel count, then the input to f for this particular level set function will be reduced by a factor $k_s^{p,1}$ in the next sweep. Likewise, if the step size is too small, the input to f will be increased by a factor $k_s^{p,2}$. If the updates are neither too small or too large, we retain the same input to f in the next sweep. Of course, if entries are extremely large or small, then a scalar multiplier in the intervals given may be insufficient. Nonetheless, the scheme performs reasonably well in the application considered here.

Note that up until now, both intervals I^m_{target} and I'^m_{target} have constant lower and upper bounds. These can be generalized to include bounds which are decreasing linear lines dependent on the sweep number. This is not arbitrary, as we expect diminishing returns of gradient descent methods as the sweep number increases. Insisting on the same amount of voxels to be altered later in the algorithm is likely to lead to artefacts in the reconstruction, as the data misfit reduces slower in later iterations. The proportion between N_j^m and data misfit should remain the same, which is what these dynamic intervals hope to capture. Therefore, we introduce a dynamic interval $\hat{I}^m_{target} = \hat{I}^m_{target}(s, \eta_m)$, which induces $\hat{I}'^m_{target} = \hat{I}'^m_{target}(s, \eta_m)$. We choose the bounds of $\hat{I}'^m_{target}$ to behave linearly as a function of the sweep number s:

Algorithm 4 Line search criteria for one sweep in shape-based LK-Level sets

```
procedure INITIAL CONDITIONS
    Choose $N^m_{inf}, N^m_{sup}, d_m, \eta_m, \underline{\tau}_0, \mathbf{w}_0, S$
    Determine $N^m_{low}$ and $N^m_{high}$ using (2.61) and (2.62)
    Initialize $\mathbf{w}_0 \circ \underline{\tau}_0$ so that $\overline{N}^m \in I^m_{target}$ for first sweep
procedure LINE SEARCH CRITERIA FOR A SWEEP
    Compute $\hat{N}^m_{low}(s, \eta_m)$ and $\hat{N}^m_{high}(s, \eta_m)$ using (2.64, 2.65)
    Use interval $\hat{I}'^m_{target}(s, \eta_m) = [\hat{N}^m_{low}, \hat{N}_{high}]$ to determine $\mathbf{w}_s$ from criterion in (2.63)
    Use interval $\hat{I}^m_{target}(s, \eta_m) = [\hat{N}^m_{inf}, \hat{N}^m_{sup}]$ for backtracking scheme in algorithm 3
```

$$\hat{N}^m_{low}(s, \eta_m) = -\frac{1}{S} N^m_{low} \eta_m s + N^m_{low}; \tag{2.64}$$

$$\hat{N}^m_{high}(s, \eta_m) = \frac{1}{S}(N^m_{low} - N^m_{high}) \eta_m s + N^m_{high}, \tag{2.65}$$

where $\eta_m \in [0, 1]$. A recipe defining the line search criteria for one sweep in level set inversion is shown in algorithm 4. The complete LK-Single level set reconstruction scheme with the novel line search criteria and Sobolev smoothing is described in algorithm 5. We will perform numerical experiments using this algorithm in chapter 4.

Algorithm 5 Shape-based LK-Single level set reconstruction with dynamic backtracking line search and Sobolev smoothing

```
procedure Initialization
    Choose $n_s, n_r, f, S, b_0$ and $I_j$ for $j = 1, 2, \dots, n_s$.
    Initialize level set function as a signed distance function $\phi^0$
    Apply initial conditions in algorithm 4 for line search criteria
procedure LK-Single level set Reconstruction
    for $s = 0 : S - 1$ do (loop over sweeps)
        Follow steps in algorithm 4 when applicable
        for $j = 0 : n_s - 1$ do (loop over sources)
            Calculate $\nabla \mathcal{J}_j[\phi_j^s]$ using (2.56)
            Compute smoothed gradient $\widehat{\nabla \mathcal{J}_j[\phi_j^s]}$ by solving (2.38)
            Compute step size $\tau_s = f(w_s \tau_0)$   [$f(\cdot)$ is the backtracking function]
            Update level set function: $\phi_{j+1}^s = \phi_j^s - \tau_s \widehat{\nabla \mathcal{J}_j^s[\phi_j^s]}$
            Rescale $\phi_{j+1}^s \mapsto \xi \phi_{j+1}^s$ with scaling parameter $\xi \in \mathbb{R}^+$.
        Set $\phi_0^{s+1} = \phi_{n_s}^s$
```

2.1.3.3

Color level set regularization

It is natural to extend the single level set approach in the previous section to include more than two conductivity domains. Since we cannot uniquely define these domains with one level set function, we must introduce more level set functions. We require N level set functions to characterize 2^N conductivity domains uniquely.

In this section, we will consider two level set formulations; the first considers three conductivity domains and the second considers four conductivity domains. We will provide the general algorithm at the end of the section for both of these formulations, as one is a small extension of the other.

As discussed, level set methods provide additional regularization to the inverse problem as they assume *a priori* knowledge on the conductivity value. Single level set methods, which assume the domain is composed of two parameter values, have been shown to perform well in electromagnetic inverse problems [39, 41, 58]. However, in more realistic scenarios, such as multiple conductivity values present in the domain, single level set methods suffer. This is especially if the parameter values differ in several orders of magnitude (high contrast), as will be observed in chapter 4. In such scenarios, color level set methods can be applied in an attempt to deal with these new challenging problems. Color level set inversion has received limited attention when compared with single level set inversion. But for recent applications, including breast cancer screening, crack detection and multi-phase permittivity imaging, see [3, 39, 41, 65] for more details.

We begin by introducing two sufficiently smooth level set functions $\phi_{1,2} : \mathbb{R}^3 \to \mathbb{R}$, such that

$$b(\Phi)(\mathbf{x}) = \begin{cases} b^1(\mathbf{x}) & \text{in } S_1 \text{ where } \phi_1(\mathbf{x}) \leq 0 \\ b^2(\mathbf{x}) & \text{in } S_2 \text{ where } \phi_1(\mathbf{x}) > 0 \text{ and } \phi_2(\mathbf{x}) \leq 0 \\ b^3(\mathbf{x}) & \text{in } S_3 = \Omega \backslash (S_1 \cup S_2) \text{ where } \phi_1(\mathbf{x}) > 0 \text{ and } \phi_2(\mathbf{x}) > 0, \end{cases} \tag{2.66}$$

where $\Phi = (\phi_1, \phi_2)$ and we label (b^1, b^2) and b^3 as the interior and exterior conductivities respectively. As in single level set inversion, b^1, b^2 and b^3 can be smoothly varying or constant.

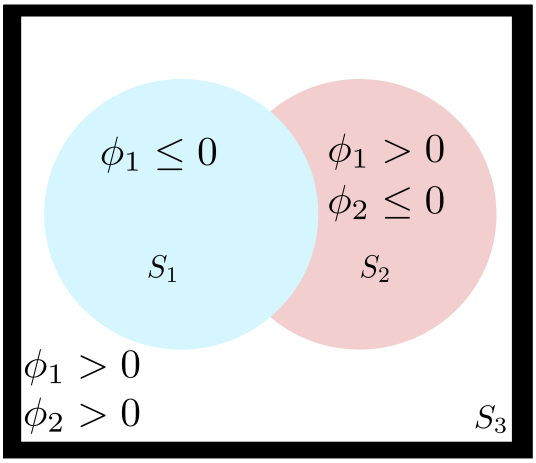


Figure 2.3: 2D Visualization of color level set representation defined in (2.66).

An alternative equality of the conductivity, which will help us when deriving the gradient scheme, is

$$b(\Phi) = b^1(1 - H(\phi_1)) + b^2 H(\phi_1)[1 - H(\phi_2)] + b^3 H(\phi_1) H(\phi_2),$$

where $H(\cdot)$ is the Heaviside function. With the color level set representation of the conductivity profile formally defined, we now introduce the relevant data misfit and cost functional:

$$\mathcal{T}[\Phi] = \mathcal{R}[b(\Phi)], \ \mathcal{J}[\Phi] = ||\mathcal{T}[\Phi]||^2_{\mathcal{L}^2(\Omega)}. \tag{2.67}$$

We wish to minimize the cost $\mathcal{J}$ in (2.67) such that a suitable $\mathbf{\Phi}$ is found. Using (2.43) and (2.44) we obtain the formal gradient

$$\nabla \mathcal{J}[\mathbf{\Phi}] = \sum_{j=1}^{n_s} \nabla \mathcal{J}_j[\mathbf{\Phi}] = \sum_{j=1}^{n_s} \mathcal{T}_j'[\mathbf{\Phi}]^* \mathcal{T}_j[\mathbf{\Phi}], \tag{2.68}$$

where

$$\mathcal{T}_j'[\mathbf{\Phi}]^* \mathcal{T}_j[\mathbf{\Phi}] = \mathrm{Re} \begin{bmatrix} b_{\phi_1}(\mathbf{\Phi}) \mathcal{R}_j'[b]^* \mathcal{R}_j[b] \\ b_{\phi_2}(\mathbf{\Phi}) \mathcal{R}_j'[b]^* \mathcal{R}_j[b] \end{bmatrix}, \tag{2.69}$$

and

$$\begin{aligned} b_{\phi_1}(\mathbf{\Phi}) &= [(b^2 - b^1) + (b^3 - b^2) H(\phi_2)] \delta(\phi_1); \\ b_{\phi_2}(\mathbf{\Phi}) &= [(b^3 - b^2) H(\phi_1)] \delta(\phi_2), \end{aligned}$$

where the subscript in b refers to the partial derivative with respect to that quantity and $\delta(\cdot)$ is the Dirac Delta distribution. Analogous to the single level set case, $\delta(\phi_i)$ has to be approximated by a function in a suitable space in the advent of numerical experiments, since it is undefined at the zeros of ϕ_i. Therefore, we choose to use a narrowband function as an approximation [97, 107], such that $\delta(\phi_i) \approx \chi_{B_d(\Gamma_i)}$, where

$$B_{d_i}(\Gamma_i) = \{\mathbf{x} : ||\mathbf{x} - \Gamma_i||_2^2 \leq d_i/2\},$$

$d_i > 0$ and χ_D is the characteristic function. In the general case, we assume that boundaries of objects are smooth. Therefore, each component in the gradient from (2.69) is projected to a smoother space using the equality in (2.38). As in the single level set scheme, we use the LK optimizer for updating the level set vector:

$$\mathbf{\Phi}_{j+1}^s = \mathbf{\Phi}_j^s - \underline{\tau}_s \circ \nabla \widehat{\mathcal{J}_{[s]}[\mathbf{\Phi}}_j^s],$$

where $\underline{\tau}_s = [\tau_s^1, \tau_s^2]^T$ is chosen by the line search criteria developed in section 2.1.3.2 and $\circ$ is the Hadamard product. Before introducing the color level set reconstruction scheme in full, we will briefly mention another formulation which assumes a slightly more complicated structure of the conductivity profile. Let us consider three sufficiently smooth level set functions $\phi_{1,2,3} : \mathbb{R}^3 \mapsto \mathbb{R}$ such that

$$b(\Phi)(\mathbf{x}) = \begin{cases} b^1(\mathbf{x}) & \text{in } S_1 \text{ where } \phi_1(\mathbf{x}) \leq 0 \\ b^2(\mathbf{x}) & \text{in } S_2 \text{ where } \phi_1(\mathbf{x}) > 0 \text{ and } \phi_2(\mathbf{x}) \leq 0 \\ b^3(\mathbf{x}) & \text{in } S_3 \text{ where } \phi_1(\mathbf{x}) > 0, \phi_2(\mathbf{x}) > 0 \text{ and } \phi_3(\mathbf{x}) \leq 0 \\ b^4(\mathbf{x}) & \text{in } S_4 = \Omega \backslash (S_1 \cup S_2 \cup S_3) \text{ where } \phi_1(\mathbf{x}) > 0, \phi_2(\mathbf{x}) > 0 \text{ and } \phi_3(\mathbf{x}) > 0, \end{cases} \tag{2.70}$$

where $b^{1,2,3,4}$ are constant or smoothly varying in $S_{1,2,3,4}$. Though unless otherwise stated, we assume they are constant. The representation in (2.70) can be conveniently written as

$$b(\Phi) = b^1(1 - H(\phi_1)) + H(\phi_1)\left[b^2(1 - H(\phi_2)) + H(\phi_2)(b^3(1 - H(\phi_3)) + b^4 H(\phi_3))\right],$$

where $H(\cdot)$ is the Heaviside function. In this case, the gradient vector is:

$$\mathcal{T}_j'[\Phi]^* \mathcal{T}_j[\Phi] = \operatorname{Re} \begin{bmatrix} b_{\phi_1}(\Phi) \mathcal{R}_j'[b]^* \mathcal{R}_j[b] \\ b_{\phi_2}(\Phi) \mathcal{R}_j'[b]^* \mathcal{R}_j[b] \\ b_{\phi_3}(\Phi) \mathcal{R}_j'[b]^* \mathcal{R}_j[b] \end{bmatrix},$$

where

$$\begin{aligned} b_{\phi_1}(\Phi) &= \left[(b^2 - b^1) + (b^3 - b^2) H(\phi_2) + (b^4 - b^3) H(\phi_2) H(\phi_3)\right] \delta(\phi_1); \\ b_{\phi_2}(\Phi) &= \left[(b^3 - b^2) H(\phi_1) + (b^4 - b^3) H(\phi_1) H(\phi_3)\right] \delta(\phi_2); \\ b_{\phi_3}(\Phi) &= \left[(b^4 - b^3) H(\phi_1) H(\phi_2)\right] \delta(\phi_3). \end{aligned}$$

The two level set formulations described in (2.66) and (2.70) adopt a hierarchical structure for defining the conductivities. Notice, for example, that b^1 only depends one sign of one level set function. We consider the hierarchical structure throughout this thesis, though other level set formulations are possible and could be explored.

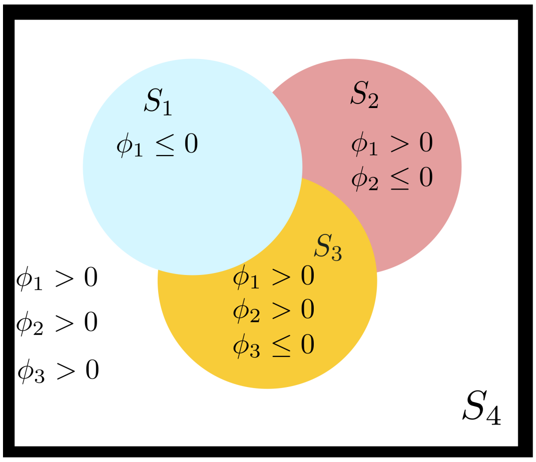


Figure 2.4: 2D Visualization of color level set representation defined in (2.66).

Figure 2.4 shows a visualization of the conductivity formulation described in (2.70). Whilst increasing complexity of the conductivity profile is appealing, since it reflect more complicated scenarios, color level set inversion encounters new problems. Sometimes, in single level set inversion, the algorithm cannot recover the main characteristics of the true content inside boxes or containers. For example, only one of two objects which are present in the content may be recovered. Furthermore, if *a priori* information is incorrect, either by underestimating or overestimating the value of conductivity, the corresponding reconstruction either grows and shrinks the domain with respect to what it sees in the data. Whilst incorrect *a priori* information seems to show stability of the single level set regime, it may lead to new problems in the color level set regime.

Flexibility of the reconstruction with respect to the *a priori* information filters into the color level set regime. In particular, this behaviour can lead to unwanted ambiguous reconstructions. For example, depending on where the initial guess is placed, the reconstruction may lead to correct conductivity domains from the *a priori* information, but incorrect locations. We can summarize two main problems from this discussion: the first is not identifying all objects present in the box or container and the second is incorrect location of reconstructed objects. This leads us to introduce a method which in some situations alleviates these problems.

2.1.3.4
A stochastic seeding process

Unlike single level set methods, when the color level set scheme converges to local minima, it may find the correct location of an object but it may also incorrectly classify the wrong conductivity at that location. We can verify this behaviour when we have access to the true content of the box or container, however in practice this is not the case. Though improving this unwanted behaviour whilst knowing the true content can only improve confidence in the reconstruction, however. To address this issue, we introduce a technique to avoid these unwanted local minima.

Topological perturbations have been used when recovering lithofacies in reservoir imaging [44, 127]. Typically, artificial objects are placed inside the imaging domain, resulting in an alteration to the level set function. Then, the reconstruction algorithm proceeds and either grows or shrinks these *seed objects* depending on what it sees in the data. This method is particularly useful in low sensitivity regions, as shown in [127], where they successfully recover other lithofacie of the same value in regions where no objects were reconstructed with the standard algorithm. With its omission, artificial objects have been observed to aid single level set inversion in finding other objects present in the imaging domain which were otherwise not found by the standard single level set inversion in this application. The intention here is to use this method for a slightly different purpose. Rather, we would like to resolve incorrect classification of objects. We intend to alter both level set functions such that *seed objects* are created inside the imaging domain, where the term *seed objects* refers to the objects being artificially placed. It is intended that the artificial objects grow or shrink in the optimization process, depending on what it sees in the data.

Let us assume that we want to place an object centred in $\Omega_2 \subset \Omega$. Without loss of generality, we consider this for one level set function. Placing an object in this region amounts to lowering the level set function until it is negative in Ω_2. Since numerical experiments will have coarse grids for computational reasons, we only consider two regions: Ω_1 and Ω_2, such that $\Omega = \Omega_1 \cup \Omega_2$ where Ω_2 is the domain containing a *seed object* and Ω_1 is that without. Whilst a interim region of smoothing the level set function between Ω_1 and Ω_2 would be useful, in this application we will not consider this smoothing. A smoothing region will have marginal effect on how discontinuous the level set function is, as we only have a small amount of grid cells. Using a fine grid in this challenging 3D imaging modality is computationally demanding and would be currently infeasible. To create *seed objects* let us introduce a cost function:

$$\mathcal{K}(\Psi) = \frac{a}{2}||\chi_1(\Psi - \phi)||^2 + \frac{b}{2}||\chi_2(\Psi - \mu)||^2, \tag{2.71}$$

where $a, b > 0$ are weighted parameters, $\mu < 0$ and

$$\chi_k(\mathbf{x}) = \begin{cases} 1 & \text{in } \Omega_k; \\ 0 & \text{in } \Omega \backslash \Omega_k, \end{cases}$$

which is the characteristic function. The first term in (2.71) is a penalty on the distance between new and old level set functions in the domain outside the seed region (Ω_1) and the second penalizes distance between the desired minimizer Ψ and some negative μ inside Ω_2. The new level set function ϕ_{new}, which is created upon minimization of (2.71), replaces the old level set function ϕ_{old}. Mathematically speaking

$$\phi_{new} = \underset{\Psi}{\operatorname{argmin}}\, \mathcal{K}(\Psi) = \frac{\alpha_1}{2}||(\Psi - \phi_{old})\chi_1||^2 + \frac{\alpha_2}{2}||(\Psi - \mu)\chi_2||^2. \tag{2.72}$$

In the numerical experiments considered here, the minimization of (2.71) takes place during a period of iterations in the level set inversion routine, which we define as the *seeding phase*. This process is disjoint to the level set inversion routine and can be seen as a pre or post processing step to the updates of Φ. We prefer a pre processing step as it creates more opportunity for the algorithm to populate the imaging domain and decide what to grow and shrink. In the context of the level set functions themselves, all we want to do is make them have the value μ inside Ω_2.

To solve the minimization problem in (2.72), we employ a gradient descent method. The iteration formula for finding a minimizer of (2.71) can be found by perturbing the input space of the cost to find an expression for the gradient. Following that, we can use standard gradient descent methods to find a minimizer. Perturbing the cost function results in

$$\begin{aligned} \mathcal{K}(\Psi + h) &= \mathcal{K}(\Psi) + \alpha_1 \langle \chi_1(\Psi - \phi), h \rangle + \alpha_2 \langle \chi_2(\Psi - \mu), h \rangle + O(||h||^2); \\ &= \mathcal{K}(\Psi) + \langle \mathcal{K}'(\Psi), h \rangle + O(||h||^2), \end{aligned}$$

where

$$\mathcal{K}'(\Psi) = \alpha_1\chi_1(\Psi - \phi) + \alpha_2\chi_2(\Psi - \mu).$$

In its discrete form, this yields the iteration formula:

$$\begin{aligned} \Psi^{(0)} &= \phi \\ \Psi^{(n+1)} &= \Psi^{(n)} - \kappa\left[\alpha_1\chi_1(\Psi - \phi) + \alpha_2\chi_2(\Psi - \mu)\right] \end{aligned} \tag{2.73}$$

where $n = 1, 2, \ldots, M$.

Algorithm 6 Shape-based LK-Color Level set reconstruction with dynamic backtracking line search, Sobolev smoothing and optional seeding phase

```
procedure INITIALIZATION
    Choose $n_s, n_r, f, \gamma, M, S$ and $I_j$ for $j = 0, 1, \ldots, n_s - 1$
    Initialize level set vector $\Phi^0 = (\phi_1^0, \phi_2^0)$, where each entry in $\Phi^0$
    are signed distance functions representing ellipsoids
    Apply initial conditions shown in algorithm 4
    If seeding choose $\gamma \in \mathbb{N}$ else choose $\gamma < 0$
procedure LK-COLOR LEVEL SET RECONSTRUCTION
    for $s = 0 : S - 1$ do (loop over sweeps)
        Follow steps in algorithm 4 when applicable
        if $0 \leq s < \gamma$ then
            procedure SEEDING PHASE
                for each entry in $\Phi^s$ do
                    Choose $\Omega_2$ by random selection (without replacement)
                    Perform $M$ iterations of (2.73)
        for $j = 0 : n_s - 1$ do (loop over sources)
            Calculate $\nabla \mathcal{J}_j[\Phi_j^s]$ using (2.68)
            Compute $\widehat{\nabla \mathcal{J}_j[\Phi_j^s]}$ using (2.38)
            Compute step size $\underline{\tau}_s = \mathbf{f}(\mathbf{w}_s \circ \underline{\tau}_0)$  [$\mathbf{f}(\cdot)$ is the vector backtracking function]
            Update level set vector: $\Phi_{j+1}^s = \Phi_j^s - \underline{\tau}_s \circ \widehat{\nabla \mathcal{J}_j[\Phi_j^s]}$
            Rescale $\Phi_{j+1}^s \mapsto \xi \Phi_{j+1}^s$ with scaling parameter $\xi \in \mathbb{R}^+$.
        Set $\Phi_0^{s+1} = \Phi_{n_s}^s$
```

The *seeding phase* is flexible and can be administered at any stage of the inversion routine. In [127], they decide to seed in various cycles. We decide to have an initial 'burn-in' phase, whereby we seed Φ at the beginning of each sweep s for $0 \leq s < \gamma < S$. This amounts to solving γ minimization problems of (2.71) in the burn-in phase for each level set function [1]. This decision arises from the observation that once level set functions in the color level set regime have converged, it tends to be difficult for the shapes to be flipped in value by seeding. Therefore, the *seeding phase* can be viewed as a dynamic initial guess scheme to

---

[1] Note that the seeding can be carried out in both single and color level set regimes. The vector quantity Φ can have multiple entries which are altered via the cost function in (2.71)

encourage the minimization process to leave the vicinity of a local minimizer resembling incorrect classification, or at least a method to collect distinct local minima. The LK-Color level set algorithm with an optional seeding phase, novel line search criteria and Sobolev smoothing is shown in algorithm 6. Note that an equivalent algorithm with inclusion of the seeding phase can be made for the single level set case shown in algorithm 5.

Notice we have used notation b to define the conductivity in parts of this chapter. To be clear, we refer to conductivity as simply 'b' in these cases as the imaginary component in this thesis does not change. Though it must be said that the gradient calculations in the pixel-based regimes (also projected into shape-based) are with respect to the complex admittivity, even if the imaginary component is eventually made redundant in the update.

CHAPTER 3

Forward Problem Review

The inverse problem associated with the near-field electromagnetic imaging problem, described in section 1.2, requires solutions to a mathematical model of electromagnetism in the form of Maxwell's equations. This chapter provides a review of existing methods which solve Maxwell's equations, as described in section 1.1, both analytically and numerically.

All regularization schemes in chapter 2 use the adjoint field method, whether that be used directly in the pixel-based regime or projected, as in the shape-based regime. Both types require repeated calculation of electromagnetic field solutions to Maxwell's equations for newly updated conductivity profiles resulting from the nonlinear LK optimizer. This allows us to calculate new electromagnetic data measured at the receiver locations, which can then be used to calculate the next gradient for the next iteration in the optimization process. In fact, any gradient-based optimizer for this inverse problem requires repeated calculation of electromagnetic field solutions using Maxwell's equations. In one step of a LK iteration we require access to a solution from both forward and adjoint problems, which plainly speaking involves two solutions to Maxwell's equations. Therefore, in order to solve the inverse problem practically, we must have a method of solving Maxwell's equations that is computationally inexpensive and reliable when the conductivity profile is highly discontinuous (as would be anticipated when scanning boxes or containers). For example, a typical box may have a background conductivity of air against some embedded highly conductive material. In a numerical setting, this would mean that the conductivity varies over several orders of magnitude from one cell to the next. We would ideally have access to a numerical scheme which lends itself well to these dramatic changes.

3.1

Literature survey

This section is devoted to reviewing literature surrounding numerical and analytical solutions to Maxwell's equations. The two main choices for computational electromagnetic (CEM) modelling are those based on differential equations (DE) and integral equations (IE). For integral equation methods we approximate integrals as finite sums and for differential equation methods we approximate derivatives as finite differences [32]; both leading to a linear system of equations. In general, DE methods generate sparse matrices and IE methods generate full matrices, relating to their different connectivity structures (local vs global).

Due to its sparse structure, numerical techniques can be employed to make DE methods more computationally efficient. Furthermore, the complexity of DE methods is irrespective to the type of material being modelled, whereas the complexity of IE methods vary dependent upon the type of material (e.g. isotropic, anisotropic, inhomogeneous, homogeneous, amongst others). However, IE solutions are generally more accurate and efficient. For example, spurious solutions exist in DE methods which do not in IE methods.

Since both methods have their benefits we discuss both, including: boundary element method (BEM), finite difference frequency/time domain methods (FDFD/FDTD), finite integration technique (FIT), finite volume method (FVM) and the finite element method (FEM).

The FDFD method is a differential equation solver used to find numerical solutions to Maxwell's equations in the frequency domain, which are also sometimes called the time-harmonic Maxwell equations. It is closely related to the FDTD method, which is an analogous finite difference scheme of the time domain system described in (1.3). The FDTD acronym was first coined by Taflove in his 1980 paper describing solutions to sinuisodial steady state electromagnetic penetration problems [121]. Nearly ten years later the FDFD acronym appeared in Ling's paper, which was on electromagnetic scattering problems [77]. The FDTD method discretizes Maxwell's equations in the time domain, creating a time stepping algorithm in the process. The similarities between FDFD and FDTD arise because they follow similar techniques for discretization. Both methods discretize their respective systems using the Yee grid, which is a orthogonal staggered Cartesian grid of the electric and magnetic fields. The grid was devised by Kane Yee in his seminal paper [135], and is still widely used due to its simplicity and computational inexpense [53]. The Yee grid was originally developed for the FDTD method, but has since been adopted for the FDFD method. Like any discretization, the Yee grid has numerical artefacts. The discretization comes with a staircase approximation at the boundaries, which yields inaccuracies in the resolution. This is mainly due to the Yee grid being rectangular, therefore complex geometries are not handled well. Despite this, there have been techniques developed to improve the accuracy of this staircase approximation. For example: local grid refinements [85], locally conforming FDTD methods [61][46] and time-domain FVMs on fully unstructured grids [84, 136] have shown to demonstrate higher accuracy, albeit more expensive [33]. Though in cases of simple geometries, such as those considered in this thesis, these inaccuracies are negligible.

The uses of both FDFD and FDTD are entirely dependent on the problem at hand; for example, if we are interested in the scattering pattern that a broadband pulse of energy makes from a particular scatterer or investigating transient states and dynamics, then we could consider a FDTD numerical solution [112]. Similarly, if we are interested in an steady state solution then we may choose to use the FDFD method because it treats steady states accurately [64]. It is important to not overstate this though, as we would anticipate that we would get similar results when using either the FDFD or FDTD method.

We are interested in solving Maxwell's equations in the frequency domain. Our interest lies here as we want to calculate low frequency contributions to the time signal. This means much of the time signal in the FDTD solution would be discarded since electromagnetic shielding stops high frequency electromagnetic waves from penetrating. However, we must not dismiss research using the FDTD method, as many of the methods and analysis surrounding it can be directly applied to the FDFD method (for reasons already discussed). The FDFD method has a couple of notable advantages:

1. Allows simulation of the electric and magnetic field of nearly arbitrarily shaped structures;

2. Dispersive materials are trivial to implement.

However, some problems can occur:

1. Stair casing error due to grid discretization, meaning it doesn't fare well with complex geometries;

2. FDFD method requires solving large sparse linear system, even for the simplest of problems. Therefore, it is computationally expensive.

The FDFD method has proven highly useful within the geophysical community, which typically handle low frequency problems. A 3D FDFD method using a finite-difference discretization scheme was developed in the paper [21] for an application in geophysics, and was later used for an inversion routine which involved recovery of injected plumes of conducting steam [36, 37].

The FIT is another differential equation solver, and was first introduced in Weiland's paper in 1977 [132]. The FIT method is similar to the FDTD method in its discretization for homogeneous media. The difference is that the FIT method transforms the Maxwell system from integral form into a linear system of equations. The FIT has a greater flexibility in modeling complex shapes, which is one of its strengths [102]. Like the FDFD and FDTD methods, the FIT uses a Yee grid for its discretization. Analogous to FDTD research, refinements of the Yee discretization have been studied. For example, adaptive meshes, subgridding [98], conformal FIT (CFIT) [26] and nonorthogonal grids have all been introduced to try and overcome the Yee grid's drawbacks. Improvements to the FIT follow suit with the refinements that have been made to the FDTD method; the computational cost increases. However, Z.Rahimi's thesis addresses these problems by extending the FIT, where he strikes a balance between speed and accuracy of the FIT [102]. This differential equation solver can be formulated in both time and frequency domains, which is convenient. The MAFIA (solution of MAxwell's Equations by the Finite Integration Algorithm) code, developed by T.Weiland, is a software package that solves Maxwell's equations on structured grids. It is an internationally recognized package, with various solvers for different problems. For example, MAFIA contains a frequency domain solver in Cartesian and cylindrical coordinates and a 3D time domain solver in Cartesian coordinates, which may be useful for our problem.

The FEM is a computational method which subdivides the domain of interest into small finite-sized elements of geometrically simple shapes, such as triangles. The union of this yields the finite-element mesh. In general, a PDE is then applied to and formulated for each element. The fields of interest are approximated in each element as a simple function, such as a quadratic polynomial. This typically reduces to a linear system of equations, which is a shared property with FDFD/FDTD/FIT. Unlike the finite difference schemes, the FEM can deal with complex geometries well. Typically, this is done by using unstructured grids/meshes. For example, unstructured meshes with tetrahedra allow for a good approximation of curved objects [10]. Moreover, unstructured meshes allow for local refinement where the solution may have inaccuracies. Furthermore, the FEM produces a well defined solution function everywhere in the domain, making it possible to apply stability and convergence analysis with relative ease [10].

For electromagnetic problems, the FEM has (mainly) two avenues [88]: nodal elements and edge elements. If discontinuities in the electric and/or magnetic field arise, edge elements can resolve this problem, whereas nodal elements cannot, since they are automatically continuous functions of the spatial variables. However, the computational cost of edge elements are over twice as much as nodal elements. The first reason is due to orthogonality of the Cartesian directions used in nodal elements, rendering many matrix entries to zero and thus allowing for code optimization. The second being that the continuity conditions at the interfaces between edge elements increase connectivity of the matrix [88]. Although edge elements may be more computationally expensive, the ability of being able to model discontinuous media with accuracy allows us to model both homogeneous and inhomogeneous media whereas nodal elements are largely restricted to the former [88]. Due to this restriction, more consideration should be given to edge elements because we are largely interested in inhomogeneous media.

The BEM is a different technique to the others because it is derived from the integral formulation of the quasi-static Maxwell system. One example of a BEM is the work done by Misaki and Tsuboi [83]. The method is based on Green's theorem. In their method, conductor regions and boundaries of two materials are split into tetrahedra and triangular elements respectively. Furthermore, they assume unknown electric field vectors on the boundaries of the two materials and the conductor regions. In addition, they assume unknown magnetic flux density vectors on the boundary of the two materials [83]. The problem is formulated in terms of scalar and vector potentials with corresponding boundary conditions, since the vector potential across the boundaries of the two materials must be consistent. The BEM has many different formulations for three dimensional eddy current problems. For example, Mayergoyz [81] presents a boundary integral equation (BIE) formulation in terms of the equivalent surface current and a magnetic charge. As discussed previously, Misaki and Tsuboi [83] formulate the quasi-static Maxwell system in terms of the electric field vector and the magnetic flux density. Another formulation is the $A-\Omega$ approach, where A is the magnetic vector potential in the conductor and the magnetic scalar potential in non-conductor media, has been applied to low frequency eddy current problems [92, 109]. Furthermore, a $A,V-A$ formulation has been presented, where the magnetic vector potential is used in both conducting and non-conducting regions as well as the electric scalar potential V being used in the eddy current region for a 3D eddy current problem [103]. The BEM formulations discussed are all discretized and reduce to a linear system of equations, which is a shared property with the other forward solvers discussed. The BEM formulations discussed here are brief, however the purpose is to demonstrate that there exists multiple BEM which are tailored for specific eddy current problems. Thus, for our problem, the $A - \Omega$ approach may be worth considering since it has been applied to low frequency problems.

The FVM formulation devised in [50, 51, 52] is based on a staggered grid using a potential-generalized current formulation $(\mathbf{A}, \phi, \hat{\mathbf{J}})$ coupled with the Coulomb gauge condition. The discretization of the Maxwell system for this formulation is similar to the Yee grid, where $(\mathbf{A}, \hat{\mathbf{J}})$ is defined on the centre of each face on a cube volume. Haber and Ascher solve the full Maxwell system including the displacement current term. To deal with highly discontinuous conductive media, a harmonic averaging scheme of the conductivity is applied. This lends itself well to the imaging problem considered here as we anticipate media to be highly discontinuous conductivity-wise. Typically, Krylov space methods are employed to find solutions for the resultant large linear sparse system. Once these solutions are found, the electric and magnetic fields are computed through post processing, since they are related to the generalized current and vector potential.

The forward problem, analytically, has been extensively researched for various eddy current problems [34, 72] which involve solutions to Maxwell's equations for specific setups. In the Dodd and Deeds paper (see [34]), a multi layered structure of material is considered where a coil, centred on the z axis, is excited. They reduce the quasi-static Maxwell system to a scalar differential equation for the vector potential by assuming axial symmetry of the coil. An analytical vector potential solution in each region is found in terms of Green's functions. Due to the many relationships between electromagnetic quantities found either empirically or intrinsic to nature, we can also derive forward electromagnetic field solutions from vector potential, coil impedance and induced voltage analytical solutions. From the quasi-static Maxwell system, we can find a linear relationship between the vector potential and the electric field in the frequency domain. Therefore, using Dodd and Deeds' analytical vector potential solutions, we can arrive at both forward and adjoint electric field solutions, which can then be used to calculate gradients in the solution to our underlying inverse problem. Despite these analytical forms, we must reduce the analytical expressions to semi-analytical, since the integrand is difficult to integrate and the integral is improper. Therefore, we truncate the integral limits, whilst trying to maintain accuracy, and then employ numerical integration accordingly.

The Dodd and Deeds analytical solutions assume the conductivity profile has a layered form (not robust in our scenario). In the inverse problem introduced in chapter 2, when updating the conductivity profile, the iteration scheme does not care about the form of the forward problem. In fact, it only cares about what it sees in the data. Therefore, we must make sure that the forward solver is not violated with the newly updated conductivity profile. Given that we have access to limited data, it is unlikely that the updated conductivity profile would exactly or approximately resemble a layered structure after even just one step of the iteration process. Therefore, the Dodd and Deeds analytical solutions become unreliable after just one iteration of updating the conductivity profile. Despite all of this, the Dodd and Deeds formulation, and several other analytical forms, could be used in solving the linearized inverse problem introduced in chapter 2. The linearized approach, for example, could produce similar if not better results than difference imaging of the data.

The analytical solutions derived in the Dodd and Deeds paper has also been extended to include a more general setting involving k conductor planes [72]. Using this method for the forward problem, work has been done on a parameter inversion scheme assuming a 3 layered structure; in particular, reconstruction of material depth and conductivity [14]. In addition, reconstructing a unknown layer conductivity distribution and the shape of a crack have also been treated, through use of coil impedance measurements [95]. However, an inverse crime is committed due to the fact that Bowler and Norton use the same solver to generate the data [14]. If one was to not use the same model for the data, then after one reconstruction (or one iteration of the inversion algorithm) the analytical solutions would no longer be valid since it is not guaranteed that a layered structure would still exist. This is where the method suffers, because it is severely restricted to its geometry.

In summary, we have discussed both analytical and numerical methods to solve Maxwell's equations, and subsets of those, in both time and frequency domains. The numerical methods discussed here are either derived from integral or differential forms of Maxwell's equations. Our application suggests that a numerical solver is certainly required, since it is not guaranteed that the contents of a box or cargo container is layered or even approximately layered. The forward model must be robust and flexible, meaning models which are not too restricted to geometry or composition of material are preferred. The FVM for the vector potential formulation of Maxwell's equations, as developed in [50, 51, 52], provides some stability if conductivities in neighbouring cells are highly discontinuous. Therefore, when demonstrating the novel schemes introduced in chapter 2, we use this methodology for as the Maxwell solver.

This survey is by no means a complete picture of available analytical or numerical methods of solution to Maxwell's equations, but hopefully gives a flavour of what is available for various applications.

3.2

Vector potential formulation

This section introduces a vector potential formulation of Maxwell's equations for isotropic inhomogeneous media. The work here closely follows that in [6, 50, 51, 52].

Let us firstly reformulate (1.6a-1.6b) such that it is in second-order form. This is obtained by eliminating one of the fields. Hence,

$$\begin{aligned} \nabla \times (\mu^{-1}\nabla \times \mathbf{E}) - i\omega b\mathbf{E} &= i\omega\mathbf{J}_e + \nabla \times (\mu^{-1}\mathbf{J}_m); && (3.1a) \\ \nabla \times (b^{-1}\nabla \times \mathbf{H}) - a\mathbf{H} &= \mathbf{J}_m + \nabla \times b^{-1}\mathbf{J}_e, && (3.1b) \end{aligned}$$

where we relabel the magnetic and electric sources as $\mathbf{J}$ with subscripts m and e respectively. Choosing which equation to solve in (3.1) depends on the application. For the application considered here, we solve (3.1a). This is due to the conductivity having a greater spatial inhomogeneity when compared with the permeability. Moreover, conductivities in security screening applications are likely to be highly discontinuous. Furthermore, the second order term in equation (3.1b) is dominant and involves the spatially inhomogeneous complex admittivity term $b = b(\sigma, .)$. Hence, we introduce greater error into the model when discretizing in space. We assume that the electromagnetic fields are sufficiently smooth so that we can apply the operators in (3.1). Let us introduce the Helmholtz Decomposition:

$$\mathbf{E} = \mathbf{A} + \nabla\phi, \qquad (3.2)$$

where $\mathbf{A}$ and ϕ are vector and scalar fields respectively. The vector field $\mathbf{A}$ lies in the active space of the curl operator, whereas $\nabla\phi$ lies in the null space of the curl operator. By substitution of (3.2) into (3.1a) we obtain

$$(i\omega)^{-1}\nabla \times \mu^{-1}\nabla \times \mathbf{A} - b(\mathbf{A} + \nabla\phi) = \mathbf{J}_e + (i\omega)^{-1}\nabla \times (\mu^{-1}\mathbf{J}_m).$$

And so, the unknown quantity $\mathbf{E}$ is mapped to $\mathbf{A}$ and ϕ. Thus, we have replaced three scalar equations with four. Therefore, we introduce a gauge condition

$$\nabla \cdot \mathbf{A} = 0,$$

known as the *Coulomb gauge condition*. This condition yields a new PDE system in terms of the unknowns **A** and ϕ. Namely,

$$\begin{aligned} (i\omega)^{-1}\nabla\times\mu^{-1}\nabla\times\mathbf{A}-b(\mathbf{A}+\nabla\phi) &= \mathbf{J}_e+(i\omega)^{-1}\nabla\times(\mu^{-1}\mathbf{J}_m); && (3.3a) \\ \nabla\cdot\mathbf{A} &= 0. && (3.3b) \end{aligned}$$

Let us now note that

$$\nabla\times\mu^{-1}\nabla\times\mathbf{A}=\mu^{-1}(\nabla(\nabla\cdot\mathbf{A})-\nabla^2\mathbf{A}) \tag{3.4}$$

if and only if μ is constant. In this case, (3.4) reduces to

$$\nabla\times\nabla\times\mathbf{A}=-\nabla^2\mathbf{A}$$

by using the Coulomb gauge condition. However, when μ is inhomogeneous, this identity no longer holds. Therefore, we must treat the null space of $\nabla\times\mathbf{A}$ differently for this case. We begin by introducing the following Sobolev spaces

$$\begin{aligned} \mathbf{W}(\Omega) &= \{\mathbf{v}\in[\mathcal{L}^2(\Omega)]^3;\nabla\times\mathbf{v}\in[\mathcal{L}^2(\Omega)]^3;\nabla\cdot\mathbf{v}\in\mathcal{L}^2(\Omega)\}; \\ \mathbf{W}_0(\Omega) &= \{\mathbf{v}\in\mathbf{W}(\Omega);\nabla\cdot\mathbf{v}|_{\partial\Omega}=0,\nabla\times\mathbf{v}\times\mathbf{n}|_{\partial\Omega}=\mathbf{0}\}, \end{aligned}$$

which are useful in Vector Identity 1.

Vector Identity 1

For any $\mathbf{u} \in \mathbf{W}(\Omega)$, $\mathbf{v} \in \mathbf{W}_0(\Omega)$, we have that

$$\begin{aligned} &\left\langle \nabla \times (\mu^{-1}\nabla \times \mathbf{u}), \mathbf{v} \right\rangle_{[\mathcal{L}^2(\Omega)]^3} - \left\langle \nabla(\mu^{-1}\nabla \cdot \mathbf{u}), \mathbf{v} \right\rangle_{[\mathcal{L}^2(\Omega)]^3} \\ &= \mu^{-1}\left[\langle \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle_{[\mathcal{L}^2(\Omega)]^3} + \langle \nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v} \rangle_{\mathcal{L}^2(\Omega)} \right]. \end{aligned}$$

Proof.
Using integration by parts:

$$\left\langle \nabla(\mu^{-1}\nabla \cdot \mathbf{u}), \mathbf{v}) \right\rangle_{[\mathcal{L}^2(\Omega)]^3} = -\mu^{-1} \langle \nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v} \rangle_{\mathcal{L}^2(\Omega)}.$$

By Green's formula for the curl operator:

$$\left\langle \nabla \times (\mu^{-1}\nabla \times \mathbf{u}), \mathbf{v} \right\rangle_{[\mathcal{L}^2(\Omega)]^3} = \mu^{-1} \langle \nabla \times \mathbf{u}, \nabla \times \mathbf{v} \rangle_{[\mathcal{L}^2(\Omega)]^3}.$$

And so, the vector identity above holds ■.

With Vector Identity 1, we can stabilize the PDE system when μ is inhomogeneous. We map the operator on the left hand side of (3.4) such that:

$$\nabla \times (\mu^{-1}(\nabla \times (\cdot))) \mapsto \nabla \times (\mu^{-1}(\nabla \times (\cdot))) - \alpha\nabla(\beta\nabla \cdot (\cdot)), \tag{3.6}$$

where setting $\alpha = 1$ and $\beta = \mu^{-1}$ reproduces (3.4) for constant μ. The new operator mapping in (3.6) doesn't change the PDE system in (3.3). The new operator indirectly tries to satisfy the identity in (3.4) by stabilizing the solution such that if the Coulomb gauge condition doesn't hold, we subtract the contribution it has in the PDE. We note that if μ is inhomogeneous and the Coulomb gauge does not satisfy exactly, we get penalized twice. For notational purposes, let us define the new operator in (3.6) as:

$$\Delta_\mu := \nabla \times (\mu^{-1}(\nabla \times (\cdot))) - \nabla(\mu^{-1}\nabla \cdot (\cdot)).$$

Then, we obtain the following system:

$$\begin{aligned} (i\omega)^{-1}\Delta_\mu \mathbf{A} - b(\mathbf{A} + \nabla\phi) &= \mathbf{J}_e + (i\omega)^{-1}\nabla \times (\mu^{-1}\mathbf{J}_m); &\text{(3.7a)} \\ \nabla \cdot \mathbf{A} &= 0. &\text{(3.7b)} \end{aligned}$$

By taking the divergence of (3.7a), we obtain a new system:

$$\begin{aligned}(i\omega)^{-1}\Delta_{\mu}\mathbf{A} - b(\mathbf{A}+\nabla\phi) &= \mathbf{J}_e + (i\omega)^{-1}\nabla\times(\mu^{-1}\mathbf{J}_m);\\ \nabla\cdot(b(\mathbf{A}+\nabla\phi)) &= -\nabla\cdot\mathbf{J}_e.\end{aligned}$$

The system shown above can be reduced to the linear operator equation

$$\mathcal{A}\mathbf{x} = \mathbf{b}, \tag{3.9}$$

where

$$\mathcal{A} = \begin{pmatrix} (i\omega)^{-1}\Delta_{\mu} - bI & -b\nabla(\cdot)\\ \nabla\cdot(bI) & \nabla\cdot(b\nabla(\cdot))\end{pmatrix},$$

$$\mathbf{x} = \begin{pmatrix}\mathbf{A}\\ \phi\end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix}\mathbf{J}_e + (i\omega)^{-1}\nabla\times(\mu^{-1}\mathbf{J}_m)\\ -\nabla\cdot\mathbf{J}_e\end{pmatrix}.$$

This is the PDE system which we want to solve numerically. Once $\mathbf{x}$ is obtained, the components of this solution vector can be post processed to find electromagnetic quantities which can then be used for solving the inverse problem defined in chapter 2. To find numerical solutions, one must prescribe boundary conditions to the system so that the eventual discretization on a bounded domain mimics electromagnetic wave propagation in an unbounded domain. Let us begin by introducing an auxiliary variable

$$\hat{\mathbf{J}} := -b\mathbf{E}.$$

This yields the following boundary value problem:

$$\begin{aligned}(i\omega)^{-1}\nabla\times(\mu^{-1}\nabla\times\mathbf{E}) + \hat{\mathbf{J}} &= \mathbf{Y} \quad (\mathbf{x}\in\Omega); && (3.10a)\\ \hat{\mathbf{J}} + b\mathbf{E} &= \mathbf{0} \quad (\mathbf{x}\in\Omega); && (3.10b)\\ \mathbf{n}\times\mu^{-1}\nabla\times\mathbf{E} &= \mathbf{n}\times\mathbf{h} \quad (\mathbf{x}\in\partial\Omega), && (3.10c)\end{aligned}$$

where the source and boundary terms are

$$\begin{aligned} \mathbf{h} &:= \mu^{-1}\mathbf{J}_m; \\ \mathbf{Y} &:= \mathbf{J}_e + (i\omega)^{-1}\nabla\times\mathbf{h}. \end{aligned}$$

The boundary condition is obtained by assuming that the boundary is a perfect magnetic conductor (PMC). This yields the condition

$$\mathbf{n}\times\mathbf{H} = \mathbf{0} \quad (\mathbf{x}\in\partial\Omega). \tag{3.11}$$

To obtain that in (3.10c), we make $\mathbf{H}$ the subject in (1.6a) and substitute this quantity into (3.11). Let us now recall the Helmholtz Decomposition defined in (3.2). For the components that describe $\mathbf{E}$ in the Helmholtz Decomposition, we choose the following conditions:

$$\mathbf{n}\cdot\mathbf{A} = 0, \quad (\mathbf{x}\in\partial\Omega), \tag{3.12a}$$

$$\int_\Omega \phi\, dV = 0. \tag{3.12b}$$

These conditions suffice for the uniqueness of (3.2) [51]. Hence, the new boundary value problem is:

$$\begin{aligned} (i\omega)^{-1}\nabla\times(\mu^{-1}\nabla\times\mathbf{A}) + \hat{\mathbf{J}} &= \mathbf{Y}; \quad (\mathbf{x}\in\Omega) && (3.13a)\\ \nabla\cdot\mathbf{A} &= 0; \quad (\mathbf{x}\in\Omega) && (3.13b)\\ \hat{\mathbf{J}} + b(\mathbf{A}+\nabla\phi) &= \mathbf{0}; \quad (\mathbf{x}\in\Omega) && (3.13c)\\ \mathbf{n}\times(\mu^{-1}\nabla\times\mathbf{A}) &= \mathbf{n}\times\mathbf{h} \quad (\mathbf{x}\in\partial\Omega), && (3.13d)\\ \mathbf{n}\cdot\mathbf{A} &= 0; \quad (\mathbf{x}\in\partial\Omega) && (3.13e)\\ \int_\Omega \phi\, dV &= 0. && (3.13f) \end{aligned}$$

The boundary value problem shown in (3.10) is known to be well posed [12] and the Helmholtz Decomposition is unique given the conditions in (3.12a) and (3.12b). Thus, our boundary value problem in (3.13) also has a unique solution [6]. From (3.6), we can express (3.13a) as a stabilized PDE. Namely,

$$(i\omega)^{-1}\nabla\times(\mu^{-1}\nabla\times\mathbf{A} - \nabla\psi) + \hat{\mathbf{J}} = \mathbf{Y} \quad (\mathbf{x}\in\Omega), \tag{3.14}$$

where we define the auxiliary variable

$$\psi := \mu^{-1}\nabla\cdot\mathbf{A}. \tag{3.15}$$

Note that applying the Coulomb gauge condition automatically results in $\psi \equiv 0$. Taking the divergence of (3.14) results in:

$$\nabla\cdot\hat{\mathbf{J}} = \nabla\cdot\mathbf{J}_e.$$

This equality replaces the Coulomb gauge condition defined in (3.13b). To close the PDE problem, we must derive an additional boundary condition. Assuming that the law in (1.6b) holds on a surface $S \subset \partial\Omega$, then, integrating (1.6b) on S yields

$$\int_{\partial S}\mathbf{H}\cdot\mathbf{n}\,dl + \int_S \mathbf{n}\cdot\hat{\mathbf{J}}\,dS = \int_S \mathbf{n}\cdot\mathbf{J}_e\,dS.$$

The boundary condition in (3.10) shows that $\partial\Omega$ is a PMC. Therefore, the first term above vanishes and we can equate integrands to obtain

$$\mathbf{n}\cdot\hat{\mathbf{J}} = \mathbf{n}\cdot\mathbf{J}_e \quad (\mathbf{x}\in\partial\Omega).$$

This is an analog of a derivation for the pressure boundary condition for the pressure-Poisson equation in [47]. We also integrate the stabilized PDE over the same surface S to obtain

$$\int_{\partial S}\mathbf{H}\cdot\underline{\mathbf{n}}\,dl - \int_S \mathbf{n}\cdot\nabla\psi\,dS + \int_S \mathbf{n}\cdot\hat{\mathbf{J}}\,dS = \int_S \mathbf{n}\cdot\mathbf{J}_e\,dS,$$

therefore

$$\int_S \mathbf{n}\cdot\nabla\psi\,dS = 0.$$

In other words, we satisfy the homogeneous Neumann boundary condition $\partial\psi/\partial n = 0$ on $\partial\Omega$. Integrating up yields ψ as constant across the domain Ω. However, to satisfy the Coulomb gauge condition, we select $\psi = 0$ such that ψ is uniquely determined. And so,

$$\int_\Omega \psi\,dV = 0.$$

This closes the boundary value problem. By making $\mathbf{A}$ and ϕ the subject of our boundary value problem, we obtain the following:

$$
\begin{aligned}
(i\omega)^{-1}(\nabla\times(\mu^{-1}\nabla\times\mathbf{A})-\nabla(\mu^{-1}\nabla\cdot\mathbf{A}))-b(\mathbf{A}+\nabla\phi) &= \mathbf{J}_{e,m}; && (3.16a)\\
\nabla\cdot(b(\mathbf{A}+\nabla\phi)) = -\nabla\cdot\mathbf{J}_e, & && (3.16b)
\end{aligned}
$$

with boundary conditions

$$
\begin{aligned}
\mathbf{n}\times(\mu^{-1}\nabla\times\mathbf{A}) &= \mathbf{n}\times\mathbf{h};\\
\mathbf{n}\cdot\mathbf{A} &= 0;\\
\mathbf{n}\cdot[b(\mathbf{A}+\nabla\phi)] &= \mathbf{n}\cdot\mathbf{J}_e,
\end{aligned}
$$

and constraints

$$
\begin{aligned}
\int_\Omega \mu^{-1}\nabla\cdot\mathbf{A}\,dV &= 0,\\
\int_\Omega \phi\,dV &= 0.
\end{aligned}
$$

This is the boundary value problem we wish to solve.

3.3

Finite volume discretization

This section derives a finite volume discretization of the vector potential formulation introduced in section 3.2. We want to obtain discrete forms of this formulation so that we can compute numerical solutions to the inverse problem defined in chapter 2.

3.3.1

Domain discretization

Let us assume that the domain $\Omega = (0,H_x)\times(0,H_y)\times(0,H_z)\subset\mathbb{R}^3$. We want to discretize Ω by using a tensor product of Cartesian coordinates, which map to locations in space at which discrete grid functions approximate the continuous functions in (3.16). Without loss of generality, let us consider the x direction when deriving our rectangular grid. To discretize, we partition $(0,H_x)$ into N_x pieces and N_{x+1} vertices. Thus, we define the following domains:

$$\begin{aligned}
\overline{\Omega}^{x} &:= \left\{x_{i+\frac{1}{2}} : x_{1/2} < x_{3/2} < \ldots < x_{N_{x+1/2}}, i = 0, 1, \ldots, N_x\right\}; \qquad (3.17a)\\
\overline{\Omega}^{x'} &:= \{x_i : i = 0, 1, \ldots, N_{x+1}\} \quad \text{where}\\
x_i &:= \begin{cases} x_{1/2} & i = 0. \\ \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}) & i = 1, 2, \ldots, N_x. \\ x_{N_x+1/2} & i = N_{x+1}. \end{cases} \qquad (3.17b)
\end{aligned}$$

The domains $\overline{\Omega}^{x}$ and $\overline{\Omega}^{x'}$ are the primary and dual grids respectively. The former is a set of all locations in x at the end points of each subinterval, whereas the latter is a set of all locations in x at the centre of each subinterval. We can derive grids for the y and z directions but these are omitted as they are identical. We denote $V_{i,j,k}$ as the (i, j, k)th finite volume whose length scales

$$h_i^x := x_{i+1/2} - x_{i-1/2} \quad i = 1, 2, \ldots, N_x; \qquad (3.18a)$$

$$h_j^y := y_{j+1/2} - y_{j-1/2} \quad j = 1, 2, \ldots, N_y; \qquad (3.18b)$$

$$h_k^z := z_{k+1/2} - z_{k-1/2} \quad k = 1, 2, \ldots, N_z. \qquad (3.18c)$$

The length scales for the dual grid follow from (3.18), together with the coordinate definitions in (3.17). A qualitative description of cells $V_{i,j,k}$ and $V_{i+1/2,j+1/2/k+1/2}$ is shown in figure 3.1. From (3.18), we then have

$$\begin{aligned}
V_{i,j,k} &:= [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}] \times [z_{k-1/2}, z_{k+1/2}];\\
|V_{i,j,k}| &:= h_i^x h_j^y h_k^z \quad i = 1, 2, \ldots, N_{x+1/2};\\
&\qquad\qquad j = 1, 2, \ldots, N_{y+1/2};\\
&\qquad\qquad k = 1, 2, \ldots, N_{z+1/2}.
\end{aligned}$$

The spatial domain of discretization is then

$$\Omega^h := \overline{\Omega}^{x} \times \overline{\Omega}^{y} \times \overline{\Omega}^{z}.$$

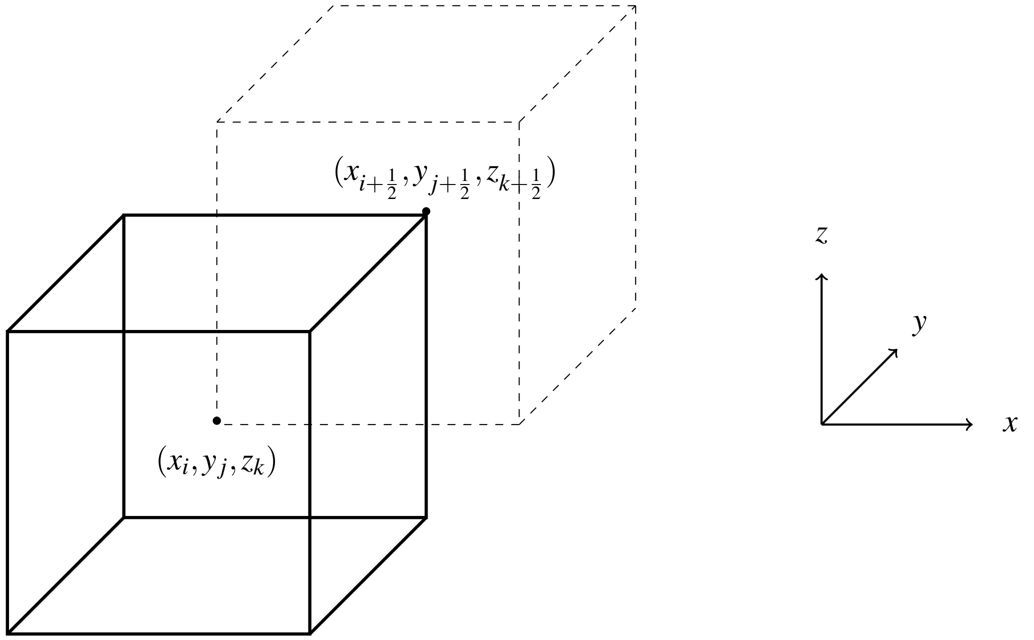


Figure 3.1: A qualitative depiction of finite volumes $V_{i,j,k}$, $V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ centred at (x_i, y_j, z_k) and $\left(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, z_{k+\frac{1}{2}}\right)$ respectively. The primary grid $\overline{\Omega}^x$ is used on $V_{i,j,k}$ and its dual grid $\overline{\Omega}^{x'}$ is used on $V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$.

The finite volume $V_{i,j,k}$ has corners $(x_{i\pm 1/2}, y_{j\pm 1/2}, z_{k\pm 1/2})$ and center (x_i, y_j, z_k), where $i = 1, 2, \ldots, N_x$, $j = 1, 2, \ldots, N_y$ and $k = 1, 2, \ldots, N_z$ [1]. Within each $V_{i,j,k}$, we assume material parameters to be constant. For example, $\sigma = \sigma_{i,j,k}$ throughout $V_{i,j,k}$. Note that this allows material parameters to be discontinuous across finite volume interfaces. With respect to the PDE system defined in (3.16), the finite volumes are what the system is integrated over.

3.3.2

Yee discretization

In the previous section, we introduced a discretization of the spatial domain. Here we explore the placement of electromagnetic quantities on the finite volumes $V_{i,j,k}$. The basic idea of the Yee Discretization is to centre the components of $\mathbf{E}$ and $\mathbf{H}$ so that $\mathbf{E}$ is surrounded by four circulating components of $\mathbf{H}$ and vice-versa. We place components of the discrete magnetic field $\mathbf{H}^h$ on the edges of $V_{i,j,k}$, whereas components of the discrete electric field $\mathbf{E}^h$ are

---

[1] The primary and dual grids are rectangular grids that are centred about (x_i, y_j, z_k) and $(x_{i+1/2}, y_{j+1/2}, z_{k+1/2})$ respectively, where $i = 0, 1, \ldots, N_x$, $j = 0, 1, \ldots, N_y$, $k = 0, 1, \ldots, N_z$. These grids are useful because of the interlaced behaviour between the electromagnetic fields $\mathbf{E}$ and $\mathbf{H}$; as will become clear when we introduce the Yee Discretization.

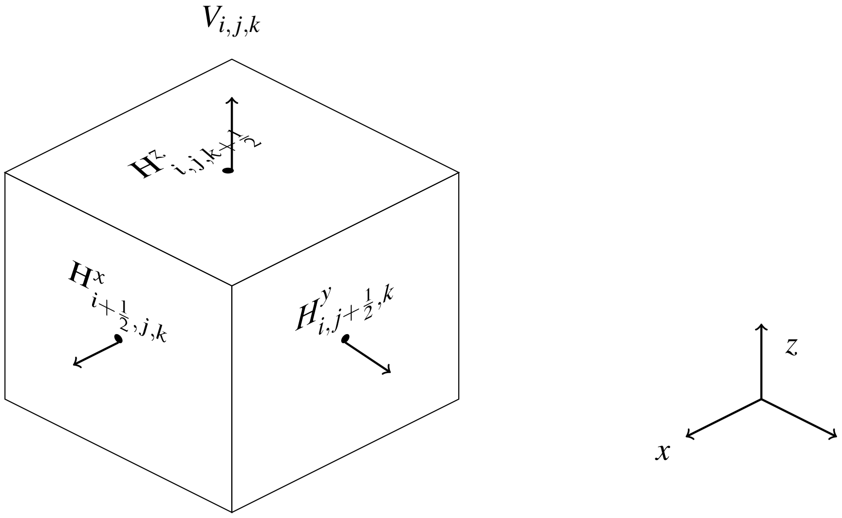


Figure 3.2: A qualitative depiction of the discrete electric field $\mathbf{H}^h$ on $V_{i,j,k}$ in the Yee scheme.

prescribed to the faces of $V_{i,j,k}$. This is known as a staggered grid, and ensures that the conservation laws of Maxwell's equations hold in a discrete setting. See [135] for more details. Mathematically, the electromagnetic fields in the finite volumes $V_{i,j,k}$ are defined as:

$$\begin{aligned}
E^x_{i,j\pm1/2,k\pm1/2} &\approx E^x(x_i, y_{j\pm1/2}, z_{k\pm1/2}); \\
E^y_{i\pm1/2,j,k\pm1/2} &\approx E^y(x_{i\pm1/2}, y_j, z_{k\pm1/2}); \\
E^z_{i\pm1/2,j\pm1/2,k} &\approx E^z(x_{i\pm1/2}, y_{j\pm1/2}, z_k); \\
H^x_{i\pm1/2,j,k} &\approx H^x(x_{i\pm1/2}, y_j, z_k); \\
H^y_{i,j\pm1/2,k} &\approx H^y(x_i, y_{j\pm1/2}, z_k); \\
H^z_{i,j,k\pm1/2} &\approx H^z(x_i, y_j, z_{k\pm1/2}),
\end{aligned}$$

where $i = 1, 2, \ldots, N_x$, $j = 1, 2, \ldots, N_y$ and $k = 1, 2, \ldots, N_z$. A qualitative representation of these discrete fields can be seen in figures 3.2 and 3.3.

The placement of electromagnetic fields in our discrete domain Ω^h is important as it mimics the rotational behaviour between $\mathbf{E}$ and $\mathbf{H}$ and preserves vector identities in a discrete setting [52]. In the discretization considered here, we exchange the placements of $\mathbf{E}$ and $\mathbf{H}$. This exchange means that the electric field $\mathbf{E}$ is now multi-valued at the same spatial location, since neighbouring cells share the same interface. Though this problem is solved by the generalized current density.

The next step is to integrate the PDE system over these finite volumes to obtain a discrete set of equations. For reasons that will become clear, we decide to use the boundary value

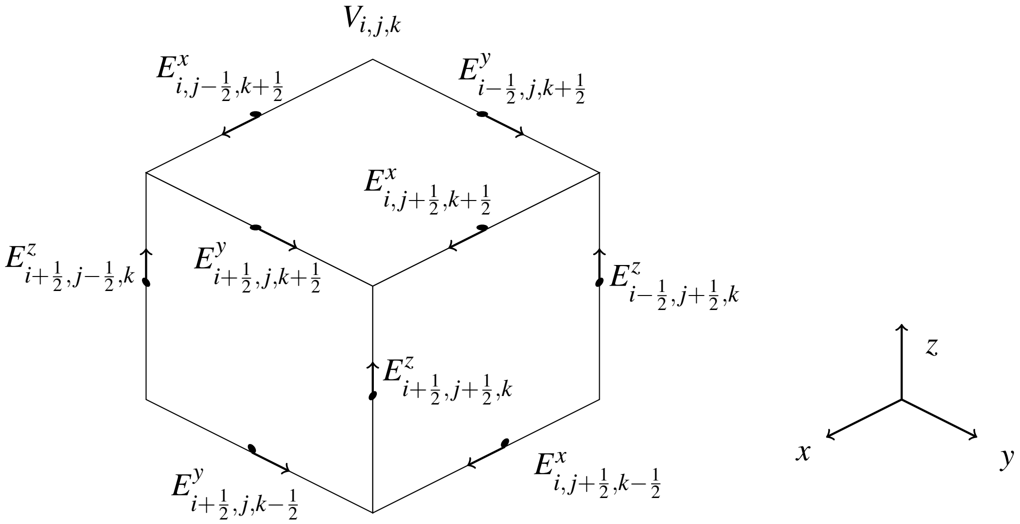


Figure 3.3: A qualitative depiction of the discrete electric field $\mathbf{E}^h$ on $V_{i,j,k}$ in the Yee scheme.

problem in terms of the auxiliary variables ψ and $\hat{\mathbf{J}}$ to integrate over. Therefore, we must first decide the placement of $(\mathbf{A}^h, \hat{\mathbf{J}}^h, \mathbf{H}^h, \phi^h, \psi^h)$ in Ω^h. We wish to derive a discrete system for linear isotropic inhomogeneous media. We determine where to place the quantities by using interface conditions between different media, and the PDE system discussed in the previous section. Interface conditions for the electromagnetic quantities are:

$$\begin{array}{rcl} \mathbf{n}^1 \cdot [\mathbf{A}^1 - \mathbf{A}^2] & = & 0; \\ \mathbf{n}^1 \times [\mathbf{A}^1 - \mathbf{A}^2] & = & \mathbf{0}; \\ \mathbf{n}^1 \cdot [\hat{\mathbf{J}}^1 - \hat{\mathbf{J}}^2] & = & 0; \\ \mathbf{n}^1 \times [\mathbf{H}^1 - \mathbf{H}^2] & = & \mathbf{0}; \\ \mathbf{n}^1 \cdot [\nabla\phi^1 - \nabla\phi^2] & = & \rho_e/\varepsilon_0, \end{array}$$

where $\mathbf{n}^1$ is the outward normal vector from medium 1 [6]. The interface conditions imply where to place the electromagnetic quantities in $V_{i,j,k}$, as they inform us which are discontinuous across adjacent finite volumes. Namely, $\mathbf{A}^h$ and $\hat{\mathbf{J}}^h$ are prescribed to finite volume faces on the primary grid. This automatically places $\nabla \cdot \mathbf{A}^h$ at the centre in $V_{i,j,k}$. Given (3.15), ψ^h is placed the same location. Equivalently, ϕ^h is defined at the centre in $V_{i,j,k}$. Therefore,

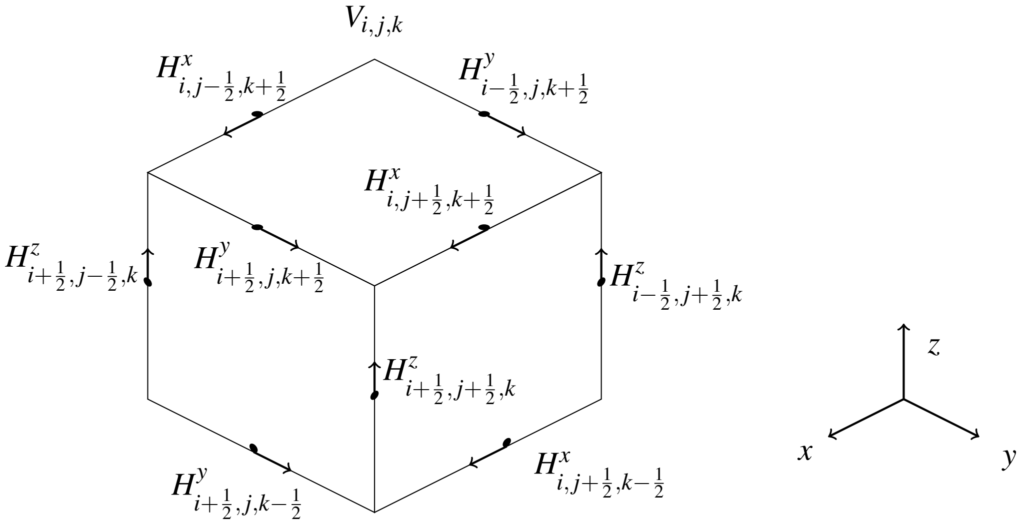


Figure 3.4: A qualitative depiction of the discrete magnetic field $\mathbf{H}^h$ on $V_{i,j,k}$.

we have the following discrete grid functions:

$$
\begin{aligned}
H^x_{i,j\pm1/2,k\pm1/2} &\approx H^x(x_i, y_{j\pm1/2}, z_{k\pm1/2}); \\
H^y_{i\pm1/2,j,k\pm1/2} &\approx H^y(x_{i\pm1/2}, y_j, z_{k\pm1/2}); \\
H^z_{i\pm1/2,j\pm1/2,k} &\approx H^z(x_{i\pm1/2}, y_{j\pm1/2}, z_k); \\
A^x_{i\pm1/2,j,k} &\approx A^x(x_{i\pm1/2}, y_j, z_k); \\
A^y_{i,j\pm1/2,k} &\approx A^y(x_i, y_{j\pm1/2}, z_k); \\
A^z_{i,j,k\pm1/2} &\approx A^z(x_i, y_j, z_{k\pm1/2}); \\
\hat{J}^x_{i\pm1/2,j,k} &\approx \hat{J}^x(x_{i\pm1/2}, y_j, z_k); \\
\hat{J}^y_{i,j\pm1/2,k} &\approx \hat{J}^y(x_i, y_{j\pm1/2}, z_k); \\
\hat{J}^z_{i,j,k\pm1/2} &\approx \hat{J}^z(x_i, y_j, z_{k\pm1/2}), \\
\phi_{i,j,k} &\approx \phi(x_i, y_j, z_k), \\
\psi_{i,j,k} &\approx \psi(x_i, y_j, z_k),
\end{aligned}
$$

where $i = 1, 2, \ldots, N_x$, $j = 1, 2, \ldots, N_y$ and $k = 1, 2, \ldots, N_z$. These discrete grid functions automatically place the source fields $\mathbf{J}_e$ and $\mathbf{J}_m$ normal to $V_{i,j,k}$'s faces and edges respectively.

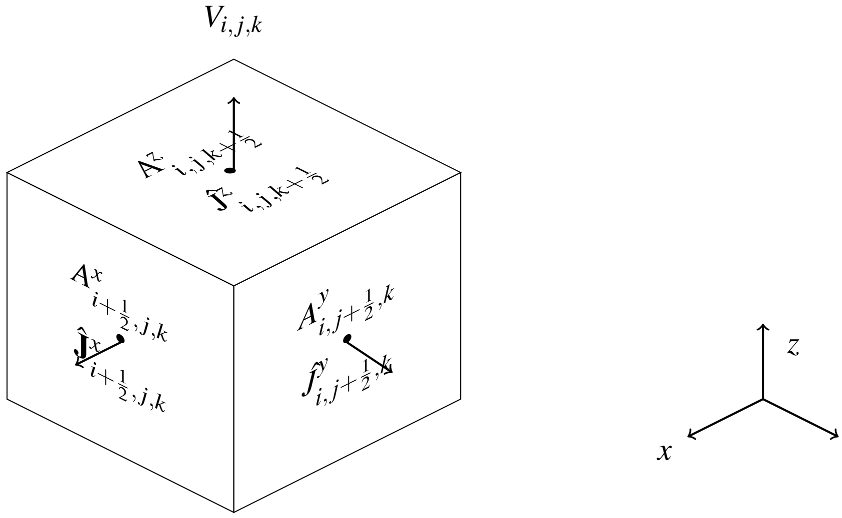


Figure 3.5: A qualitative depiction of the discrete vector potential $\mathbf{A}^h$ and generalized current density $\hat{\mathbf{J}}$ on $V_{i,j,k}$.

Let us now recall the system we wish to integrate:

$$\nabla \times \mathbf{H} - (i\omega)^{-1}\nabla\psi + \hat{\mathbf{J}} = \mathbf{J}_e; \tag{3.21a}$$

$$\nabla \cdot \hat{\mathbf{J}} = \nabla \cdot \mathbf{J}_e; \tag{3.21b}$$

$$\mu\psi - \nabla \cdot \mathbf{A} = 0; \tag{3.21c}$$

$$\hat{\mathbf{J}} + b(\mathbf{A} + \nabla\phi) = \mathbf{0}; \tag{3.21d}$$

$$\nabla \times \mathbf{A} - a\mathbf{H} = \mathbf{J}_m. \tag{3.21e}$$

We will integrate this system then find the discrete form of (3.9). Let us begin by integrating (3.21a) over faces of the primary grid. Hence

$$\int_S (\nabla \times \mathbf{H}) \cdot \mathbf{n}\, dS - (i\omega)^{-1} \int_S \nabla\psi \cdot \mathbf{n}\, dS + \int_S \hat{\mathbf{J}} \cdot \mathbf{n}\, dS = \int_S \mathbf{J}_e \cdot \mathbf{n}\, dS, \tag{3.22}$$

where S is one of the surfaces $S^x_{i-1/2,j,k}$, $S^y_{i,j-1/2,k}$ and $S^z_{i,j,k-1/2}$. These are:

$$\begin{aligned}
S^x_{i-1/2,j,k} &= \left\{x_{i-1/2}\right\} \times (y_{j-1/2}, y_{j+1/2}) \times (z_{k-1/2}, z_{k+1/2}); \\
S^y_{i,j-1/2,k} &= (x_{i-1/2}, x_{i+1/2}) \times \left\{y_{j-1/2}\right\} \times (z_{k-1/2}, z_{k+1/2}); \\
S^z_{i,j,k-1/2} &= (x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2}) \times \{z_{k-1/2}\}.
\end{aligned}$$

The integral operator in (3.22) is computed for each of the surfaces defined above, yielding a discrete equation for each surface in the domain. We will consider one surface and state the resultant discrete equations from the remaining surfaces. To be consistent, this will be done for the remaining integration on the equations in (3.21).

Hence, let us choose $S = S^x_{i-1/2,j,k}$. Then, by Stokes' Theorem, (3.22) becomes

$$\oint_{\partial S} \mathbf{H}\cdot\mathbf{n}\,dl - (i\omega)^{-1}\int_S \nabla\psi\cdot\mathbf{n}\,dS + \int_S \hat{\mathbf{J}}\cdot\mathbf{n}\,dS = \int_S \mathbf{J}_e\cdot\mathbf{n}\,dS. \tag{3.23}$$

By observation of the surface S in figure 3.6, we have that

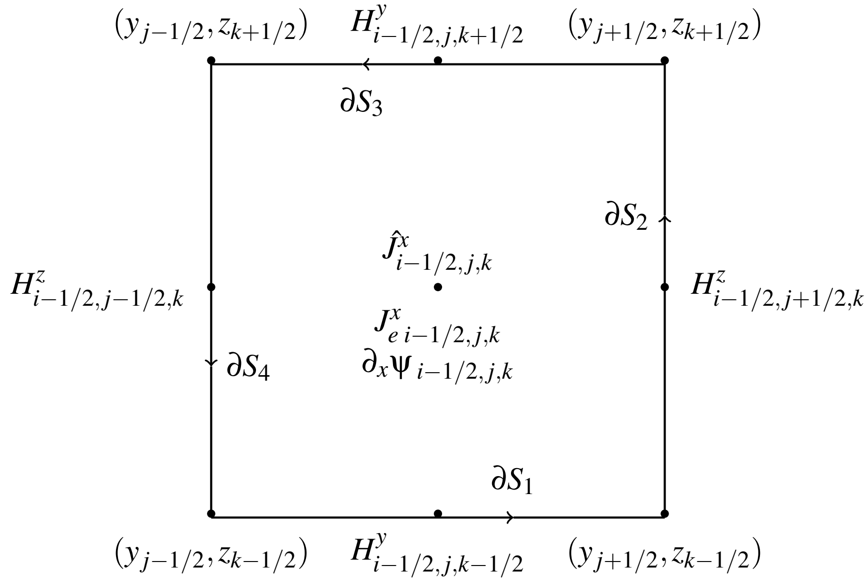


Figure 3.6: The surface $S = S^x_{i-1/2,j,k}$.

$$\begin{aligned}\oint_{\partial S} \mathbf{H}\cdot\mathbf{n}\,dl &= \oint_{\partial S_1\cup\partial S_2\cup\partial S_3\cup\partial S_4} \mathbf{H}\cdot\mathbf{n}\,dl;\\ &= \sum_{k=1}^{4}\oint_{\partial S_k} \mathbf{H}\cdot\mathbf{n}\,dl.\end{aligned}$$

The unit vector **n** traverses anti-clockwise on ∂S and picks out discrete grid functions on the boundary (which approximates the integral involving **H** in (3.23)). Therefore,

$$\text{On } \partial S_1,\ \mathbf{n}(t) = (0,t,0),\ \mathbf{n}'(t) = (0,1,0),\ dl = dt; \tag{3.24a}$$

$$\text{On } \partial S_2,\ \mathbf{n}(t) = (0,0,t),\ \mathbf{n}'(t) = (0,0,1),\ dl = dt; \tag{3.24b}$$

$$\text{On } \partial S_3,\ \mathbf{n}(t) = (0,-t,0),\ \mathbf{n}'(t) = (0,-1,0),\ dl = dt; \tag{3.24c}$$

$$\text{On } \partial S_4,\ \mathbf{n}(t) = (0,0,-t),\ \mathbf{n}'(t) = (0,0,-1),\ dl = dt. \tag{3.24d}$$

For example,

$$\begin{aligned}
\oint_{\partial S_1} \mathbf{H}\cdot\mathbf{n}\,dl &= \int_{y_{j-1/2}}^{y_{j+1/2}} H^y_{i-1/2,j,k-1/2}\,dt; \\
&= H^y_{i-1/2,j,k-1/2}[t]^{y_{j+1/2}}_{y_{j-1/2}}; \\
&= h^y_j H^y_{i-1/2,j,k-1/2}.
\end{aligned}$$

Therefore,

$$\oint_{\partial S} \mathbf{H}\cdot\mathbf{n}\,dl = h^y_j(H^y_{i-1/2,j,k-1/2} - H^y_{i-1/2,j,k+1/2}) + h^z_k(H^z_{i-1/2,j+1/2,k} - H^z_{i-1/2,j-1/2,k}).$$

Next, we compute the other integral terms on S in (3.23). We approximate the gradient of ψ by a centred finite different method. Hence,

$$\partial_x \psi_{i-1/2,j,k} \approx \frac{\psi_{i-1,j,k} - \psi_{i,j,k}}{h^x_{i-1/2}}.$$

On S, the outward pointing normal vector $\mathbf{n} = (1,0,0)$. Therefore,

$$\begin{aligned}
\int_S \nabla\psi\cdot\mathbf{n}\,dS &\approx \frac{\psi_{i-1,j,k} - \psi_{i,j,k}}{h^x_{i-1/2}} \int_{z_{k-1/2}}^{z_{k+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} dydz; \\
&= \frac{\psi_{i-1,j,k} - \psi_{i,j,k}}{h^x_{i-1/2}} h^y_j h^z_k.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\int_S \hat{\mathbf{J}} \cdot \mathbf{n}\, dS &= \int_{z_{k-1/2}}^{z_{k+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \hat{J}^x_{i-1/2,j,k}\, dydz; \\
&= \hat{J}^x_{i-1/2,j,k} h^y_j h^z_k. \\
\int_S \mathbf{J}_e \cdot \mathbf{n}\, dS &= \int_{z_{k-1/2}}^{z_{k+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} J^x_{e\, i-1/2,j,k}\, dydz; \\
&= J^x_{e\, i-1/2,j,k} h^y_j h^z_k.
\end{aligned}$$

And so, for $S = S^x_{i-1/2,j,k}$, we have that

$$\begin{aligned}
h^y_j \left(H^y_{i-1/2,j,k-1/2} - H^y_{-1/2,j,k+1/2} \right) + h^z_k \left(H^z_{i-1/2,j+1/2,k} - H^z_{i-1/2,j-1/2,k} \right) \\
+ h^y_j h^z_k \left(\frac{\psi_{i-1,j,k} - \psi_{i,j,k}}{i\omega h^x_{i-1/2}} h^y_j h^z_k + \hat{J}^x_{i-1/2,j,k} \right) = h^y_j h^z_k J^x_{e\, i-1/2,j,k}.
\end{aligned}$$

Performing the same computation for $S = S^y_{i,j-1/2,k}$ and $S = S^z_{i,j,k-1/2}$ results in:

$$\begin{aligned}
h^x_i \left(H^x_{i,j-1/2,k+1/2} - H^x_{i,j-1/2,k-1/2} \right) - h^z_k \left(H^z_{i+1/2,j-1/2,k} - H^z_{i-1/2,j-1/2,k} \right) \\
+ h^x_i h^z_k \left(\frac{\psi_{i,j,k-1} - \psi_{i,j,k}}{i\omega h^y_{j-1/2}} + \hat{J}^y_{i,j-1/2,k} \right) = h^x_i h^z_k J^y_{e\, i,j-1/2,k}. \\
h^y_j \left(H^y_{i+1/2,j,k-1/2} - H^y_{i-1/2,j,k-1/2} \right) - h^x_i \left(H^x_{i,j+1/2,k-1/2} - H^x_{i,j-1/2,k-1/2} \right) \\
+ h^x_i h^y_j \left(\frac{\psi_{i,j,k-1} - \psi_{i,j,k}}{i\omega h^z_{k-1/2}} + \hat{J}^z_{i,j,k-1/2} \right) = h^x_i h^y_j J^z_{e\, i,j,k-1/2},
\end{aligned}$$

respectively. The discrete equations in Ω^h for (3.21a) have now been computed. Next, we derive the discrete equations for (3.21b). Thus,

$$\int_V \nabla \cdot \hat{\mathbf{J}}\, dV = \int_V \nabla \cdot \mathbf{J}_e\, dV.$$

By Divergence Theorem, for any volume $V \subset \Omega$, we have that

$$\oint_S \hat{\mathbf{J}} \cdot \mathbf{n}\, dS = \oint_S \mathbf{J}_e \cdot \mathbf{n}\, dS.$$

Choosing $V = V_{i,j,k}$, we have the following surfaces to integrate over:

$$S = S^x_{i\pm1/2,j,k}, S^y_{i,j\pm1/2,k}, S^z_{i,j,k\pm1/2}. \tag{3.25}$$

Then,

$$\oint_{S^x_{i-1/2,j,k}\cup S^x_{i+1/2,j,k}\cup\ldots\cup S^z_{i,j,k+1/2}} \hat{\mathbf{J}}\cdot\mathbf{n}\, dS = \oint_{S^x_{i-1/2,j,k}\cup S^x_{i+1/2,j,k}\cup\ldots\cup S^z_{i,j,k+1/2}} \mathbf{J}_e\cdot\mathbf{n}\, dS.$$

Hence, we have the following:

$$\begin{aligned}
&\text{On } S^x_{i-1/2,j,k},\ \mathbf{n} = (-1,0,0),\ \text{On } S^x_{i+1/2,j,k},\ \mathbf{n} = (1,0,0);\\
&\text{On } S^y_{i,j-1/2,k},\ \mathbf{n} = (0,-1,0),\ \text{On } S^y_{i,j+1/2,k},\ \mathbf{n} = (0,1,0);\\
&\text{On } S^z_{i,j,k-1/2},\ \mathbf{n} = (0,0,-1),\ \text{On } S^z_{i,j,k+1/2},\ \mathbf{n} = (0,0,1).
\end{aligned}$$

For demonstration, let us choose the same surface as in figure 3.6. Hence,

$$-\int_{z_{k-1/2}}^{z_{k+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} J^x_{i-1/2,j,k}\, dydz = -\int_{z_{k-1/2}}^{z_{k+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} J^x_{e\,i-1/2,j,k} dydz.$$

Thus,

$$\begin{aligned}
h^y_j h^z_k\left(\hat{J}^x_{i+1/2,j,k} - \hat{J}^x_{i-1/2,j,k}\right) + h^x_i h^z_k\left(\hat{J}^y_{i,j+1/2,k} - \hat{J}^y_{i,j-1/2,k}\right)\\
+h^x_i h^y_j\left(\hat{J}^z_{i,j,k+1/2} - \hat{J}^z_{i,j,k-1/2}\right) = h^y_j h^z_k\left(J^x_{e\,i+1/2,j,k} - J^x_{e\,i-1/2,j,k}\right)\\
+h^x_i h^z_k\left(J^y_{e\,i,j+1/2,k} - J^y_{e\,i,j-1/2,k}\right) + h^x_i h^y_j\left(J^z_{e\,i,j,k+1/2} - J^z_{e\,i,j,k-1/2}\right).
\end{aligned}$$

Now, let us consider (3.21c). For any volume $V \subset \Omega$, we have that

$$\int_V \mu\psi\, dV = \int_V \nabla\cdot\mathbf{A}\, dV.$$

By Divergence Theorem,

$$\int_V \mu\psi\, dV = \oint_S \mathbf{A}\cdot\mathbf{n}\, dS.$$

As before, choose $V = V_{i,j,k}$ and S as in (3.25) to yield

$$\psi_{i,j,k} = \mu^{-1}_{i,j,k}\Big[h^y_j h^z_k \left(A^x_{i+1/2,j,k} - A^x_{i-1/2,j,k}\right)$$
$$+h^x_i h^z_k \left(A^y_{i,j+1/2,k} - A^y_{i,j-1/2,k}\right) + h^x_i h^y_j \left(A^z_{i,j,k+1/2} - A^z_{i,j,k-1/2}\right)\Big].$$

The field $\nabla\phi$ is potentially not as smooth as $b\nabla\phi$ across an interface between adjacent $V_{i,j,k}$, due to the interface conditions. Therefore, we divide (3.21d) by b. Then, for any path $C \subset \Omega$, we have that

$$\int_C [b^{-1}\hat{\mathbf{J}} + \mathbf{A} + \nabla\phi]\cdot\mathbf{n}\, dl = 0.$$

Choosing $C = l^x_{i-1/2,j,k} := (x_{i-1}, x_i) \times y_j \times z_k$, we have that

$$\int_{x_{i-1}}^{x_i} \left[\frac{\hat{J}^x_{i-1/2,j,k}}{b_{i-1/2,j,k}} + A^x_{i-1/2,j,k} + \frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{h^x_{i-1/2}}\right] = 0.$$

And so,

$$h^x_{i-1/2}\left(\frac{J^x_{i-1/2,j,k}}{b_{i-1/2,j,k}} + A^x_{i-1/2,j,k}\right) + \phi_{i,j,k} - \phi_{i-1,j,k} = 0. \tag{3.26}$$

Here we use a harmonic averaging scheme for the complex admittivity b. Since we have chosen a line in x, we average the quantity b over adjacent cells $V_{i-1,j,k}$ and $V_{i,j,k}$. Hence

$$b_{i+1/2,j,k} := h^x_{i+1/2}\left(\int_{x_i}^{x_{i+1}} \frac{1}{b(x,y_j,z_k)}\, dx\right)^{-1}.$$

Assuming b is constant across each finite volume, we have that

$$\begin{aligned}\int_{x_i}^{x_{i+1}} \frac{1}{b(x,y_j,z_k)}\, dx &= \int_{x_{i+1/2}}^{x_{i+1}} \frac{1}{b(x,y_j,z_k)}\, dx + \int_{x_i}^{x_{i+1/2}} \frac{1}{b(x,y_j,z_k)}\, dx; \\ &= b^{-1}_{i+1,j,k}x_{i+1} - b^{-1}_{i+1/2,j,k}x_{i+1/2} + b^{-1}_{i+1/2,j,k}x_{i+1/2} - b^{-1}_{i,j,k}x_i.\end{aligned}$$

We know that $b = b_{i,j,k}$ and $b = b_{i+1,j,k}$ in $V_{i,j,k}$ and $V_{i+1,j,k}$ respectively. This is shown qualitatively in figure 3.7. Hence,

$$\int_{x_i}^{x_{i+1}} \frac{1}{b(x,y_j,z_k)}\, dx = b^{-1}_{i+1,j,k}\left(x_{i+1} - x_{i+1/2}\right) + b^{-1}_{i,j,k}\left(x_{i+1/2} - x_i\right).$$

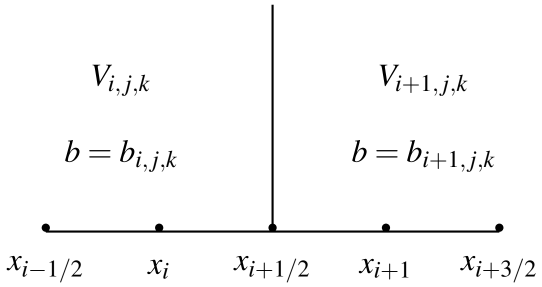


Figure 3.7: A depiction of $C = l^x_{i-1/2,j,k}$.

It follows that

$$b_{i+1/2,j,k} = h^x_{i+1/2}\left(\frac{h^x_i}{2b_{i,j,k}} + \frac{h^x_{i+1}}{2b_{i+1,j,k}}\right)^{-1}.$$

From (3.26), we obtain

$$\hat{J}^x_{i-1/2,j,k} = b_{i-1/2,j,k}\left(A^x_{i-1/2,j,k} + \frac{\phi_{i-1,j,k} - \phi_{i,j,k}}{h^x_{i-1/2}}\right),$$

where

$$b_{i-1/2,j,k} = h^x_{i-1/2}\left(\frac{h^x_i}{2b_{i,j,k}} + \frac{h^x_{i-1}}{2b_{i-1,j,k}}\right)^{-1}.$$

We use harmonic averaging instead of arithmetic averaging because it is more favourable when dealing with quantities that varies over several orders of magnitude [52].

For any surface $S \subset \Omega$, we have that

$$\int_S (\nabla \times \mathbf{A}) \cdot \mathbf{n}\, dS - i\omega \int_S \mu \mathbf{H} \cdot \mathbf{n}\, dS = \int_S \mathbf{J}_m \cdot \mathbf{n}\, dS.$$

By Stokes' Theorem,

$$\oint_{\partial S} \mathbf{A} \cdot \mathbf{n}\, dl - i\omega \int_S \mu \mathbf{H} \cdot \mathbf{n}\, dS = \int_S \mathbf{J}_m \cdot \mathbf{n}\, dS.$$

We follow the same approach as for (3.21a), except this integration is performed on the dual grid. As before, we perform the integration for one surface and state the results for the other surfaces. Thus, we choose $S = S^x_{i,j-1/2,k-1/2}$.

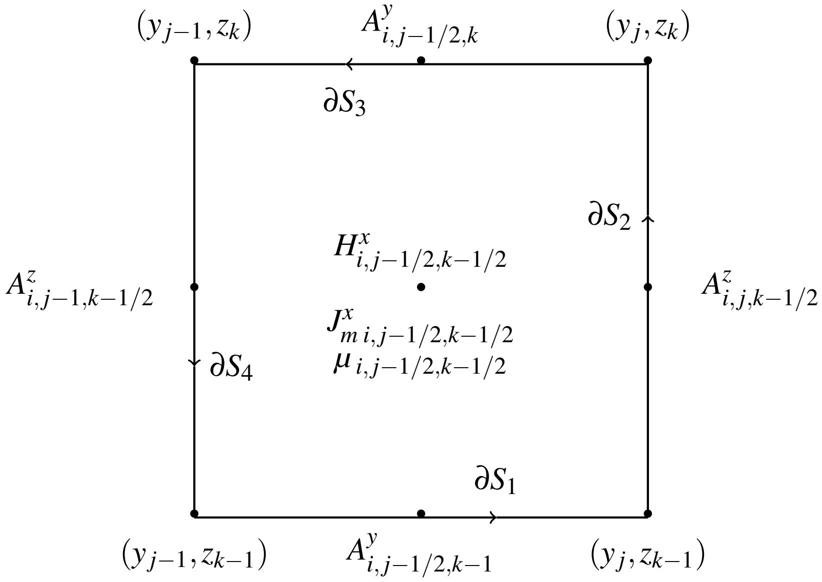


Figure 3.8: The surface $S = S^x_{i,j-1/2,k-1/2}$.

And so,

$$\oint_{\partial S} \mathbf{A} \cdot \mathbf{n}\, dl = \oint_{\partial S_1 \cup \ldots \cup \partial S_4} \mathbf{A} \cdot \mathbf{n}\, dl.$$

With (3.24), we have that

$$\oint_{\partial S} \mathbf{A} \cdot \mathbf{n}\, dl = h^z_{k-1/2} \left(A^z_{i,j,k-1/2} - A^z_{i,j-1,k-1/2} \right) - h^y_{j-1/2} \left(A^y_{i,j-1/2,k} - A^y_{i,j-1/2,k-1} \right).$$

In addition,

$$\begin{aligned} i\omega \int_S \mu \mathbf{H} \cdot \mathbf{n}\, dS &= i\omega \int_{y_{j-1}}^{y_j} \int_{z_{k-1}}^{z_k} \mu_{i,j-1/2,k-1/2} H^x_{i,j-1/2,k-1/2}\, dz dy; \\ &= i\omega h^y_{j-1/2} h^z_{k-1/2} \mu_{i,j-1/2,k-1/2} H^x_{i,j-1/2,k-1/2}. \\ \int_S \mathbf{J}_m \cdot \mathbf{n}\, dS &= \int_{y_{j-1}}^{y_j} \int_{z_{k-1}}^{z_k} J^x_{m\,i,j-1/2,k-1/2}; \\ &= h^y_{j-1/2} h^z_{k-1/2} J^x_{m\,i,j-1/2,k-1/2}. \end{aligned}$$

Therefore, we have that

$$\begin{aligned} i\omega h^y_{j-1/2} h^z_{k-1/2} \mu_{i,j-1/2,k-1/2} H^x_{i,j-1/2,k-1/2} &= -h^y_{j-1/2} h^z_{k-1/2} J^x_{m\,i,j-1/2,k-1/2} \\ + h^z_{k-1/2} \left(A^z_{i,j,k-1/2} - A^z_{i,j-1,k-1/2} \right) &- h^y_{j-1/2} \left(A^y_{i,j-1/2,k} - A^y_{i,j-1/2,k-1} \right). \end{aligned}$$

Similarly, for $S = S^y_{i-1/2,j,k-1/2}$ and $S = S^z_{i-1/2,j-1/2,k}$, we have that

$$\begin{aligned}
i\omega h^z_{k-1/2} h^x_{i-1/2} \mu_{i-1/2,j,k-1/2} H^y_{i-1/2,j,k-1/2} &= -h^z_{k-1/2} h^x_{i-1/2} J^y_{m\,i-1/2,j,k-1/2} \\
+h^z_{k-1/2}\left(A^z_{i,j,k-1/2} - A^z_{i-1,j,k-1/2}\right) &- h^x_{i-1/2}\left(A^x_{i-1/2,j,k} - A^x_{i-1/2,j,k-1}\right); \\
i\omega h^x_{i-1/2} h^y_{j-1/2} \mu_{i-1/2,j-1/2,k} H^z_{i-1/2,j-1/2,k} &= -h^y_{j-1/2} h^x_{i-1/2} J^z_{m\,i-1/2,j-1/2,k} \\
+h^y_{j-1/2}\left(A^y_{i,j-1/2,k} - A^z_{i-1,j-1/2,k}\right) &- h^x_{i-1/2}\left(A^x_{i-1/2,j,k} - A^x_{i-1/2,j-1,k}\right).
\end{aligned}$$

We have now derived all the discrete equations using finite volume integration on (3.21a-3.21e), with harmonic averaging on the complex admittivity b. The final step is to perform the analog on the permeability μ. Since the magnitude of μ doesn't vary extremely, we use standard arithmetic averaging to determine $\mu_{i,j-1/2,k-1/2}$, $\mu_{i-1/2,j,k-1/2}$ and $\mu_{i-1/2,j-1/2,k}$. Without loss of generality, we consider the edge parallel to the x axis, defining

$$\begin{aligned}
\mu_{i,j-1/2,k-1/2} &:= \frac{1}{h^y_{j-1/2} h^z_{k-1/2}} \int_{S^x_{i,j-1/2,k-1/2}} \mu(x_i,y,z)\;dydz; \\
&= \frac{1}{h^y_{j-1/2} h^z_{k-1/2}} \int_{z_{k-1}}^{z_k} \int_{y_{j-1}}^{y_j} \mu(x_i,y,z)\;dydz.
\end{aligned}$$

By observation of figure 3.9, we see that $\mu_{i,j-1/2,k-1/2}$ shares an edge with four neighbouring cells. Therefore, we consider averaging over these four cells. Coupled with the assumption of μ being constant in each cell, we have that

$$\begin{aligned}
\mu_{i,j-1/2,k-1/2} = (4h^y_{j-1/2} h^z_{k-1/2})^{-1} \Bigg[& \int_{z_{k-1}}^{z_{k-1/2}} \int_{y_{j-1}}^{y_{j-1/2}} \mu(x_i,y,z)\;dydz \\
& + \int_{z_{k-1/2}}^{z_k} \int_{y_{j-1}}^{y_{j-1/2}} \mu(x_i,y,z)\;dydz \\
& + \int_{z_{k-1/2}}^{z_k} \int_{y_{j-1/2}}^{y_j} \mu(x_i,y,z)\;dydz \\
& + \int_{z_{k-1}}^{z_{k-1/2}} \int_{y_{j-1/2}}^{y_j} \mu(x_i,y,z)\;dydz \Bigg].
\end{aligned}$$

And so,

$$\begin{aligned}
\mu_{i,j-1/2,k-1/2} &:= (4h^y_{j-1/2} h^z_{k-1/2})^{-1} \Big(h^y_{j-1/2} h^z_{k-1} \mu_{i,j-1,k-1} \\
&+ h^y_{j-1} h^z_k \mu_{i,j-1,k} + h^y_j h^z_{k-1} \mu_{i,j,k-1} + h^y_j h^z_k \mu_{i,j,k} \Big).
\end{aligned}$$

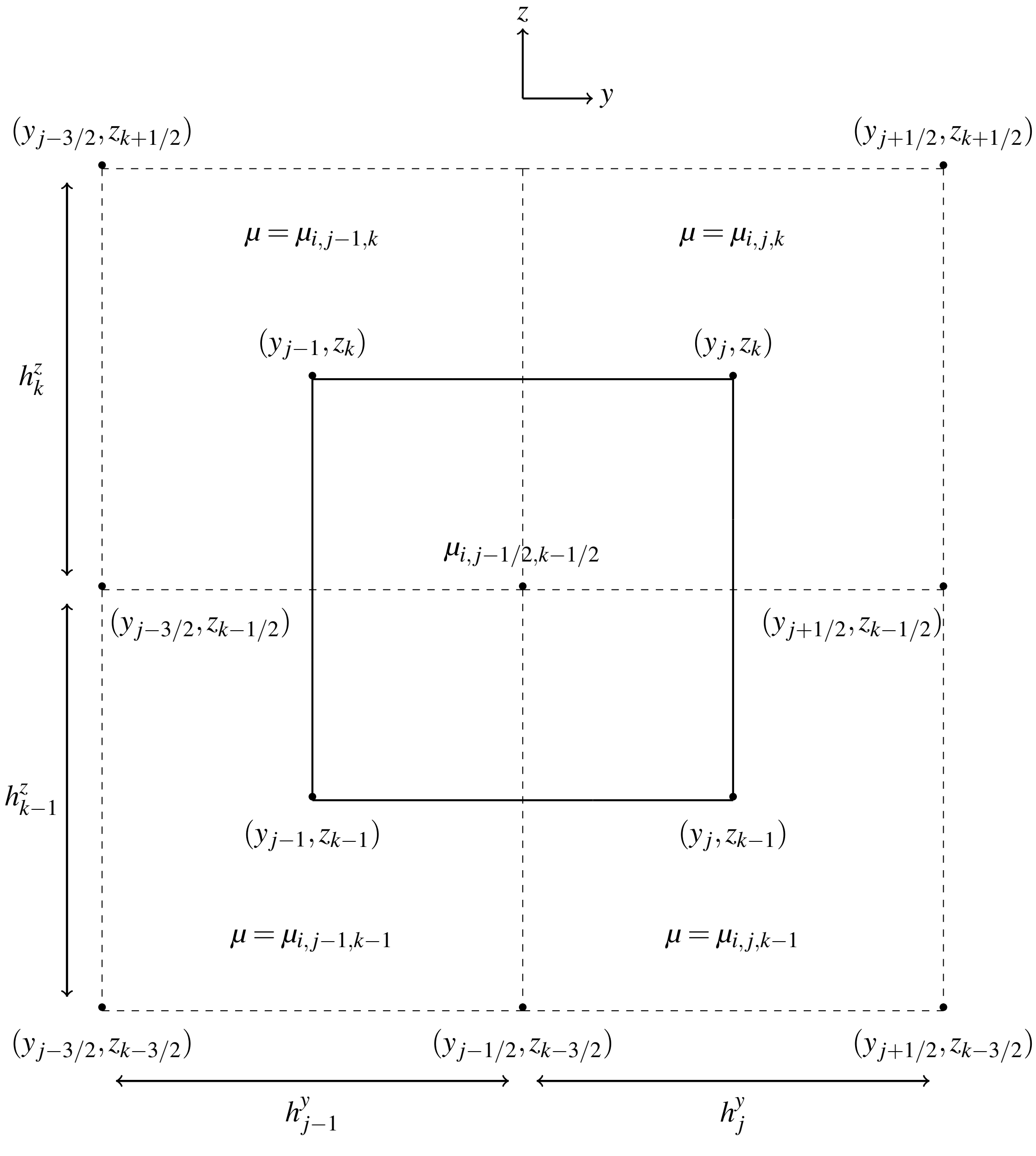


Figure 3.9: The surface $S = S^x_{i,j-1/2,k-1/2}$ and neighbouring surfaces.

Similarly,

$$
\begin{aligned}
\mu_{i-1/2,j,k-1/2} &:= \frac{1}{h^x_{i-1/2}h^z_{k-1/2}} \int_{S^y_{i-1/2,j,k-1/2}} \mu(x_i,y,z)\, dydz; \\
&= (4h^x_{i-1/2}h^z_{k-1/2})^{-1} \sum_{p,r=\pm 1/2} h^x_{i-1/2+p}h^z_{k-1/2+r}\mu_{i-1/2+p,j,k-1/2+r}; \\
\mu_{i-1/2,j-1/2,k} &:= \frac{1}{h^x_{i-1/2}h^y_{j-1/2}} \int_{S^z_{i-1/2,j-1/2,k}} \mu(x_i,y,z)\, dydz; \\
&= (4h^x_{i-1/2}h^y_{j-1/2})^{-1} \sum_{p,r=\pm 1/2} h^x_{i-1/2+p}h^y_{j-1/2+r}\mu_{i-1/2+p,j-1/2+r,k}.
\end{aligned}
$$

A discretization of (3.21a-3.21e) has now be obtained, with appropriate averaging techniques delegated to the discontinuous material coefficients between cells. The discrete system is the following:

$$
\begin{aligned}
\nabla^e_h \times \mathbf{H}^h - (i\omega)^{-1}\nabla_h \psi^h + \hat{\mathbf{J}}^h &= \mathbf{J}^h_e; \\
\nabla_h \cdot \hat{\mathbf{J}}^h &= \nabla_h \cdot \mathbf{J}^h_e; \\
\mu^h_c \psi^h - \nabla_h \cdot \mathbf{A}^h &= 0^h; \\
\hat{\mathbf{J}}^h + b^h(\mathbf{A}^h + \nabla_h \phi^h) &= \mathbf{0}^h; \\
\nabla^f_h \times \mathbf{A}^h - i\omega\mu^h_e \mathbf{H}^h &= \mathbf{J}^h_m.
\end{aligned}
$$

We can arrive at the discrete analog of the boundary value problem in (3.16) by eliminating $(\phi^h, \psi^h, \mathbf{H}^h)$ from the discrete system above. Therefore, we obtain

$$
\mathcal{A}^h x^h = b^h, \tag{3.28}
$$

where

$$
\mathcal{A}^h := \begin{pmatrix} (i\omega)^{-1}\Delta^h_\mu(\cdot) - b^h(\cdot) & -b\nabla_h(\cdot) \\ \nabla_h \cdot (b^h(\cdot)) & \nabla_h \cdot (b\nabla_h(\cdot)) \end{pmatrix};
$$

$$
\Delta^h_\mu(\cdot) := \nabla^e_h \times ((\mu^h_e)^{-1}(\nabla^f_h \times (\cdot))) - \nabla_h((\mu^h_c)^{-1}\nabla_h \cdot (\cdot)),
$$

$x^h := (\mathbf{A}^h\ \phi^h)^T$ and b^h represents the source distribution on Ω^h. Solving the linear system in (3.28) is implemented in the Python programming language and is the solver used in this thesis for numerical experiments involving the reconstruction schemes described in chapter 2. The software package is called "lsMax" and includes the forward solver, reconstruction schemes and line search criteria.

3.4

Analytical solution of Maxwell's equations

This section introduces an analytical method, with appropriate assumptions on the material parameters, for solving a specific form of Maxwell's equations. The analytical solution is used for validation of numerical solutions to (3.28) from the previous section. The derivation here closely follows that in [130].

The Python implementation for solving (3.28) has to be validated so that we are confident it is correctly implemented. In this section, we introduce an analytical solution that is derived for a harmonic dipole source in a homogeneous medium. Following that, we present numerical validation for the Python implementation using these analytical solutions. Recall that

$$\nabla \times \mathbf{E}(\mathbf{x}) - a(\mathbf{x})\mathbf{H}(\mathbf{x}) = -\mathbf{J}_m(\mathbf{x}); \tag{3.29a}$$

$$\nabla \times \mathbf{H}(\mathbf{x}) - b(\mathbf{x})\mathbf{E}(\mathbf{x}) = \mathbf{J}_e(\mathbf{x}); \tag{3.29b}$$

$$\nabla \cdot (\varepsilon \mathbf{E}) = \rho; \tag{3.29c}$$

$$\nabla \cdot (\mu \mathbf{H}) = 0, \tag{3.29d}$$

where we relabel the magnetic and electric sources as **J** with subscripts m and e respectively. Note that sign of the magnetic source is arbitrary. By linearity of Maxwell's equations in (3.29), we can decompose the total electromagnetic fields as

$$\begin{aligned} \mathbf{E} &= \mathbf{E}_m + \mathbf{E}_e; \\ \mathbf{H} &= \mathbf{H}_m + \mathbf{H}_e, \end{aligned}$$

where superscripts m and e denote magnetic and electric induced fields. Hence, for the pairs $[\mathbf{E}_e, \mathbf{H}_e]$, $[\mathbf{E}_m, \mathbf{H}_m]$ we have that $\mathbf{J}_m = 0$ and $\mathbf{J}_e = 0$ respectively. Therefore, we obtain the following set of equations:

$$\nabla \times \mathbf{E}_m = -\mathbf{J}_m - a\mathbf{H}_m; \tag{3.31a}$$

$$\nabla \times \mathbf{H}_m = b\mathbf{E}_m; \tag{3.31b}$$

$$\nabla \times \mathbf{E}_e = -a\mathbf{H}_e; \tag{3.31c}$$

$$\nabla \times \mathbf{H}_e = \mathbf{J}_e + b\mathbf{E}_e. \tag{3.31d}$$

Applying the divergence operator to (3.31a - 3.31d) yields

$$\begin{aligned}\nabla\cdot\mathbf{H}_m &= -b^{-1}\nabla\cdot\mathbf{J}_m; &\quad(3.32a)\\ \nabla\cdot\mathbf{E}_m &= 0; &\quad(3.32b)\\ \nabla\cdot\mathbf{H}_e &= 0; &\quad(3.32c)\\ \nabla\cdot\mathbf{E}_e &= -a^{-1}\nabla\cdot\mathbf{J}_e. &\quad(3.32d)\end{aligned}$$

The relationships in (3.32b -3.32c) suggest that $\mathbf{E}_m$ and $\mathbf{H}_e$ lie in the null space of the divergence operator. Therefore, we can let

$$\begin{aligned}\mathbf{E}_m &= -\nabla\times\mathbf{F}; &\quad(3.33a)\\ \mathbf{H}_e &= \nabla\times\mathbf{A}, &\quad(3.33b)\end{aligned}$$

where $\mathbf{F}$ and $\mathbf{A}$ are vector functions. We can say this is true with arbitrary sign because the curl operator lies in the null space of the divergence. These relationships can then be substituted into (3.31b - 3.31c) to obtain

$$\begin{aligned}\nabla\times\mathbf{H}_m &= -b\nabla\times\mathbf{F};\\ \nabla\times\mathbf{E}_e &= a\nabla\times\mathbf{A},\end{aligned}$$

respectively. Since the curl operator also has a non-trivial null space, we can do more than just equate both sides. Hence, we can introduce arbitrary scalar functions U and V such that

$$\begin{aligned}\mathbf{H}_m &= -b\mathbf{F}-\nabla U; &\quad(3.35a)\\ \mathbf{E}_e &= -a\mathbf{A}-\nabla V. &\quad(3.35b)\end{aligned}$$

By substituting (3.33a), (3.33b), (3.35a), (3.35b) into (3.31a) and (3.31d), we obtain

$$\begin{aligned}\nabla\times\nabla\times\mathbf{F} &= \mathbf{J}_m-ba\mathbf{F}-a\nabla U;\\ \nabla\times\nabla\times\mathbf{A} &= \mathbf{J}_e-ba\mathbf{A}-b\nabla V.\end{aligned}$$

Using the vector identity for the double curl operator, as described in (3.4), we obtain

$$\begin{aligned}\nabla(\nabla\cdot\mathbf{F})-\nabla^2\mathbf{F} &= \mathbf{J}_m-ba\mathbf{F}-a\nabla U;\\ \nabla(\nabla\cdot\mathbf{A})-\nabla^2\mathbf{A} &= \mathbf{J}_e-ba\mathbf{A}-b\nabla V.\end{aligned}$$

Recall that the scalar functions U and V are arbitrarily defined, therefore we can impose conditions on the divergence of vector functions $\mathbf{F}$ and $\mathbf{A}$. Thus, if we let

$$\begin{aligned} \nabla \cdot \mathbf{F} &= -aU; \\ \nabla \cdot \mathbf{A} &= -bV, \end{aligned}$$

then we obtain

$$\nabla^2 \mathbf{F} + k^2 \mathbf{F} = -\mathbf{J}_m; \tag{3.39a}$$

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = -\mathbf{J}_e. \tag{3.39b}$$

The relationships in (3.39a) and (3.39b) are the inhomogeneous Helmholtz equations for vector functions $\mathbf{F}$ and $\mathbf{A}$ respectively. The analytical solution described here is for a harmonic magnetic dipole source placed in an infinite medium. This means that the electrically induced fields $\mathbf{E}_e$ and $\mathbf{H}_e$ are zero. Therefore, the total fields are:

$$\mathbf{E} = -\nabla \times \mathbf{F}; \tag{3.40a}$$

$$\mathbf{H} = -b\mathbf{F} + a^{-1}\nabla(\nabla \cdot \mathbf{F}). \tag{3.40b}$$

Let there be a small loop of current I at a source location $\mathbf{x}_s = (x_s, y_s, z_s)$, which is z-directed (i.e. in the xy plane). This can be represented by an infinitesimal magnetic dipole with a moment $m = IdS$ where dS is the area of a small loop. Moreover, we choose the magnetization vector to be given as

$$\mathbf{Z} = m\delta(x_{sc} - x_s)\delta(y_{sc} - y_s)\delta(z_{sc} - z_s)\mathbf{e}_z.$$

Then, we have the corresponding Helmholtz equation:

$$\nabla^2 \mathbf{F} + k^2 \mathbf{F} = -i\omega\mu\mathbf{Z} = -\mathbf{J}_m. \tag{3.41a}$$

Since the magnetization vector has only a z component, this reduces (3.41a) to the Helmholtz operator acting on a scalar field F_z:

$$\nabla^2 F_z + k^2 F_z = -i\omega\mu m\delta(\mathbf{x}_{sc} - \mathbf{x}_s), \tag{3.42}$$

which can be written in the form $HF_z(\mathbf{x}_{sc}) = f(\mathbf{x}_{sc})$, where H is the Helmholtz operator.

Note that F_z can be found using Green's functions (the impulse response of a inhomogeneous linear differential equation with given boundary conditions). In our case, this is the inhomogeneous Helmholtz equation defined in an infinite space. Hence, we can find a Green's function G and subsequently F_z by using linearity properties of the Green's function.

Let us now define the triple spatial Fourier transform mappings, which will help find F_z:

$$\begin{aligned}
\mathfrak{F}[\mathbf{F}(x_{sc},y_{sc},z_{sc})] &= \tilde{\mathbf{F}}(k_x,k_y,k_z) \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathbf{F}(x_{sc},y_{sc},z_{sc})e^{-iq}d\mathbf{x}_{sc}; \\
\mathfrak{F}^{-1}[\tilde{\mathbf{F}}(k_x,k_y,k_z)] &= \mathbf{F}(x,y,z) \\
&= \frac{1}{(2\pi)^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{\mathbf{F}}(k_x,k_y,k_z)e^{iq}d\mathbf{k},
\end{aligned}$$

where $q = k_x x + k_y y + k_z z$. Now, let us use these pairings to transform (3.41a). We know that

$$\mathfrak{F}[\nabla^2\mathbf{F}] = (\mathfrak{F}[\nabla^2 F_x], \mathfrak{F}[\nabla^2 F_y], \mathfrak{F}[\nabla^2 F_z])^T,$$

where

$$F_k = \frac{1}{(2\pi)^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathfrak{F}[F_k]e^{-iq}d\mathbf{x}_{sc} \quad k = x,y,z.$$

Hence,

$$\frac{\partial^2 F_k}{\partial x^2} + \frac{\partial^2 F_k}{\partial y^2} + \frac{\partial^2 F_k}{\partial z^2} = (-k_x^2 - k_y^2 - k_z^2)F_k.$$

It follows that

$$\mathfrak{F}[\nabla^2 F_k] = (-k_x^2 - k_y^2 - k_z^2)\mathfrak{F}[F_k].$$

And so,

$$\mathfrak{F}[\nabla^2\mathbf{F}] = (-k_x^2 - k_y^2 - k_z^2)\mathfrak{F}[\mathbf{F}].$$

As a result, we arrive at the transformed equation:

$$\tilde{\mathbf{F}} = \tilde{G}\tilde{\mathbf{J}}, \tag{3.44}$$

where

$$\tilde{G} = \frac{1}{(k_x^2 + k_y^2 + k_z^2 - k^2)}. \tag{3.45}$$

The function $\tilde{G}$ is the Fourier transform of G in (3.42). Since (3.44) is a product in the Fourier space, it is a convolution in the inverse mapping; namely, the spatial domain. This is given by

$$\mathbf{F}(x_{sc}, y_{sc}, z_{sc}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(x_{sc}-x', y_{sc}-y', z_{sc}-z')\mathbf{J}(x', y', z')d\mathbf{x}'. \tag{3.46}$$

Hence, if we can compute the Green's function G, then we can obtain $\mathbf{F}$ given any source distribution $\mathbf{J}$. Thus, we evaluate (3.40a) and (3.40b) to obtain the electromagnetic fields $\mathbf{E}$ and $\mathbf{H}$. From (3.45), we can say that

$$(-k_x^2 - k_y^2 - k_z^2 + k^2)\tilde{G} = -1.$$

Applying the inverse Fourier operator results in

$$(\nabla^2 + k^2)G = -\delta(\mathbf{x}_{sc} - \mathbf{x}_s).$$

To find G, we evaluate the inverse Fourier operator so that

$$G(x_{sc}, y_{sc}, z_{sc}) = \frac{1}{(2\pi)^3}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{G}e^{iq}d\mathbf{k}. \tag{3.47}$$

Let us first consider the outermost integral, then we wish to compute

$$\hat{G} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \tilde{G}e^{ik_z z}dk_z.$$

From [101],

$$\hat{G} = \frac{e^{u|z|}}{2u},$$

where $u = (k_x^2 + k_y^2 - k^2)^{1/2}$. This brings (3.47) to the form

$$G(x_{sc}, y_{sc}, z_{sc}) = \frac{1}{8\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{u}e^{-u|z|}e^{i(k_x x + k_y y)}dk_x dk_y. \tag{3.48}$$

Now, let us compute the following mappings:

$$k_x = \lambda\cos\psi, \;\; k_y = \lambda\sin\psi;$$
$$x_{sc} = \rho\cos\phi, \;\; y_{sc} = \rho\sin\phi.$$

Then, since $0 \leq r < \infty$ and $0 < \psi \leq 2\pi$, (3.48) becomes

$$G(\rho, \phi, z) = \int_0^\infty \frac{e^{-(\lambda^2-k^2)^{1/2}|z|}}{(\lambda^2-k^2)^{1/2}} \int_0^{2\pi} e^{i\lambda\rho\cos(\psi-\phi)} d\psi \lambda d\lambda.$$

By using the integral representation in [131], we find that

$$\int_0^{2\pi} e^{i\lambda\rho\cos(\psi-\phi)} d\psi = 2\pi J_0(\lambda\rho).$$

Hence, for the integrand in (3.48), we obtain

$$G(\rho, z) = 2\pi \int_0^\infty \frac{\lambda e^{-(\lambda^2-k^2)^{1/2}|z|}}{(\lambda^2-k^2)^{1/2}} J_0(\lambda\rho) d\lambda,$$

where $\lambda^2 = k_x^2 + k_y^2$, $\rho^2 = x_{sc}^2 + y_{sc}^2$ and J_0 is the zeroth order Bessel function. From [101], we find that

$$G(\mathbf{x}_{sc}) = \frac{e^{-ikr}}{4\pi r},$$

where $r = |\mathbf{x}_{sc}| = [(x_{sc} - x_s)^2 + (y_{sc} - y_s)^2 + (z_{sc} - z_s)^2]^{1/2}$. Thus, (3.46) becomes

$$\mathbf{F}(\mathbf{x}_{sc}) = \int_{V'} \frac{e^{-ik(r-r')}}{4\pi(r-r')} \mathbf{J}(x', y', z') d\mathbf{x}'.$$

From (3.42), we have that

$$\mathbf{J}(\mathbf{x}_{sc}, \mathbf{x}_s) = i\omega\mu m \delta(\mathbf{x}_{sc} - \mathbf{x}_s)\mathbf{e}_z.$$

It follows that

$$\mathbf{F}(\mathbf{x}_{sc}) = \left(i\omega\mu m \int_{V'} \frac{e^{-ik(r-r')}}{4\pi(r-r')} \delta(\mathbf{x}' - \mathbf{x}_s) d\mathbf{x}' \right) \mathbf{e}_z.$$

With this definition, we can find $\mathbf{F}$ by using the sifting property of the Dirac Delta function:

$$\mathbf{F}(\mathbf{x}_{sc}) = i\omega\mu m \frac{e^{-ikr}}{4\pi r} \mathbf{e}_z.$$

We then find **E** and **H** using the expressions in (3.40), resulting in:

$$\begin{aligned}
\mathbf{E} &= \frac{i\omega\mu m}{4\pi r^2}(ikr+1)e^{-ikr}\left(\frac{z}{r}\mathbf{e}_x - \frac{y}{r}\mathbf{e}_y\right); \\
\mathbf{H} &= \frac{m}{4\pi r^3}e^{-ikr}\left[\left(\frac{xz}{r^2}\mathbf{e}_x + \frac{yz}{r^2}\mathbf{e}_y + \frac{z^2}{r^2}\mathbf{e}_z\right)(-k^2r^2+3ikr+3) + (k^2r^2-ikr-1)\mathbf{e}_z\right].
\end{aligned}$$

These electromagnetic solutions correspond to a isotropic homogeneous medium due to a harmonic magnetic dipole. For more information on analytical solutions to Maxwell's equations, on this and others, see [130].

3.5
Validation of numerical implementation

This section shows validation of the Python implementation which solves (3.28) numerically. In particular, we provide a comparison between the Python solver and the analytical solutions described in the previous section. We use the same grid for the analytical solution so that we can compare pointwise with the numerical method. Here, we choose to have an interior grid composed of $60 \times 60 \times 60$ cells. To mimic an unbounded domain outside this interior region, we apply a perfectly matched layer (PML) on its exterior. We use the following cell widths for our comparison:

$$\begin{aligned}
h_i^x &= 0.1[m]; \\
h_j^y &= 0.05[m]; \\
h_k^z &= 0.05[m],
\end{aligned}$$

for $i, j, k = N_p^{(x,y,z),l} + 1, N_p^{x,y,z,l} + 2, \ldots, I^{x,y,z}$, where $I^{x,y,z}$ is the total number of interior cells in the x, y, z directions and $N_p^{(x,y,z),l}$ denotes the total number of PML cells in the x, y, z directions from the left. This grid size, $6(m) \times 3(m) \times 3(m)$, is chosen as it is a realistic spatial domain for large boxes or cargo containers. For the exterior cells (i.e. inside the PML), we have that

$$h_i^x = \begin{cases} (N_p^{x,l}+1-i)h & i = 1,2,\ldots,N_p^{x,l}. \\ (i-I^x)h & i = I^x+1, I^x+2,\ldots,N^x. \end{cases} \tag{3.50a}$$

$$h_j^y = \begin{cases} (N_p^{y,l}+1-j)h & j = 1,2,\ldots,N_p^{y,l}. \\ (j-I^y)h & j = I^y+1, I^y+2,\ldots,N^y. \end{cases} \tag{3.50b}$$

$$h_k^z = \begin{cases} (N_p^{z,l}+1-k)h & k = 1,2,\ldots,N_p^{z,l}, \\ (k-I^z)h & k = I^z+1, I^z+2,\ldots,N^z. \end{cases} \tag{3.50c}$$

Here, we compute relative errors for the x, y components of the electric field $\mathbf{E}$ and the magnetic field $\mathbf{H}$ in the interior cells. We use standard arithmetic averages for the errors. Therefore,

$$\Delta E_x = \frac{1}{n_v}\sum_{k=N_p^{z,l}}^{I^z}\sum_{j=N_p^{y,l}}^{I^y}\sum_{i=N_p^{x,l}}^{I^x}\frac{\|E^{x,a}_{i-1/2,j,k}-E^{x,n}_{i-1/2,j,k}\|_2}{\|E^{x,n}_{i-1/2,j,k}\|_2}; \tag{3.51a}$$

$$\Delta E_y = \frac{1}{n_v}\sum_{k=N_p^{z,l}}^{I^z}\sum_{j=N_p^{y,l}}^{I^y}\sum_{i=N_p^{x,l}}^{I^x}\frac{\|E^{y,a}_{i,j-1/2,k}-E^{y,n}_{i,j-1/2,k}\|_2}{\|E^{y,n}_{i,j-1/2,k}\|_2}; \tag{3.51b}$$

$$\Delta H_x = \frac{1}{n_v}\sum_{k=N_p^{z,l}}^{I^z}\sum_{j=N_p^{y,l}}^{I^y}\sum_{i=N_p^{x,l}}^{I^x}\frac{\|H^{x,a}_{i,j-1/2,k-1/2}-H^{x,n}_{i,j-1/2,k-1/2}\|_2}{\|H^{x,n}_{i,j-1/2,k-1/2}\|_2}; \tag{3.51c}$$

$$\Delta H_y = \frac{1}{n_v}\sum_{k=N_p^{z,l}}^{I^z}\sum_{j=N_p^{y,l}}^{I^y}\sum_{i=N_p^{x,l}}^{I^x}\frac{\|H^{y,a}_{i-1/2,j,k-1/2}-H^{y,n}_{i-1/2,j,k-1/2}\|_2}{\|H^{y,n}_{i-1/2,j,k-1/2}\|_2}, \tag{3.51d}$$

where superscripts a and n denote analytical and numerical solutions respectively and $n_v = I^xI^yI^z$[2]. For the validation, we choose to have $N_p^{x,y,z,l,r} = 10$, $f = 1\text{kHz}$, $\sigma = 0.1$ and $I = 0.1\text{A}$. The PML used here makes cells progressively larger in the exterior cells to mimic the infinity limit in an unbounded domain. This is achieved by applying the length scales in (3.50) to the PML cells.

Instead of computing those in (3.51), we compute a subset of them. We dismiss two cells deep from the PML in our error estimate. These cells are irrelevant in our error estimate because they will not be used to measure data. The difference in error with and without these cells is marginal.

Let us now show a surface plot of the relative errors shown in (3.51), for all (i, j) with k

---

[2] Note that we have no error estimate for E_z due to this quantity being 0, since the source is z directed

fixed. We fix k such that it is close to the source.

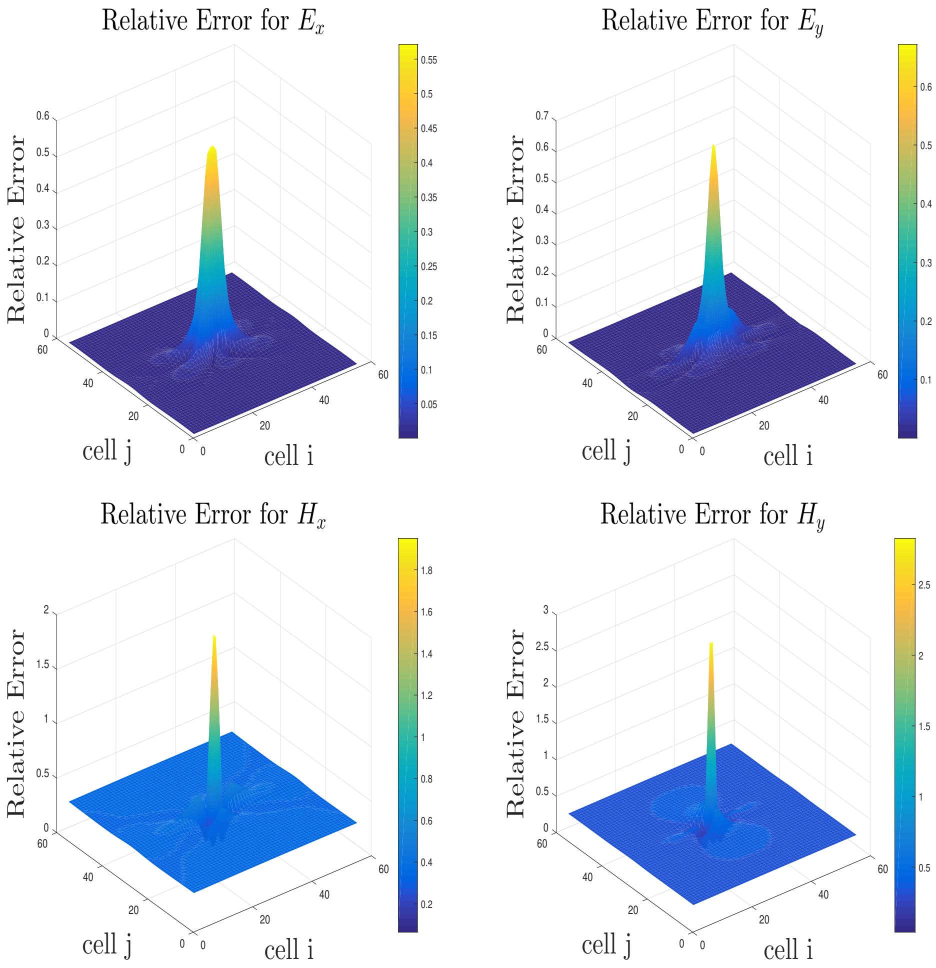


Figure 3.10: Surface plots for error estimates $\Delta E_x, \Delta E_y, \Delta H_x$ and ΔH_y for $i, j = N_p^{x,y,l} + 2, \ldots, N^{x,y} - 2$, k fixed.

Figure 3.10 shows the relative error for the electric and magnetic fields. We see that the relative error peaks at the centre of our grid. This is expected as the source location is placed at the centre in (i, j), so we should obtain greater error in this location as we try to approximate the dipole source by a small rectangular loop whose dimensions are $h_i^x \times h_j^y$. Therefore, we expect that near the source the numerical method has inaccuracies. Even though this occurs near the source, propagation of error for the rest of our domain is minimal; as can be observed in the total relative error. In the numerical experiments, we apply a filter nearby sources so that these inaccuracies of the forward model do not penalize the inversion.

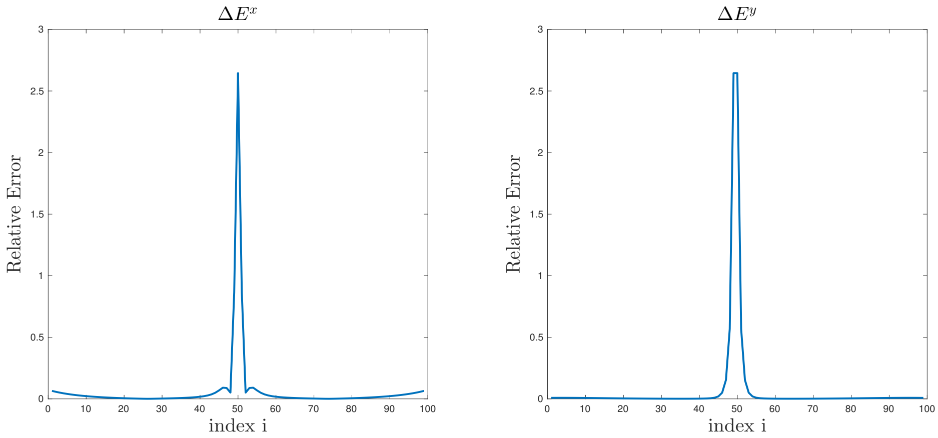


Figure 3.11: Relative error plot showing significant peak at source cell.

Figure 3.11 shows the middle row of the electric field for $i = j$ and fixed k. We find that the error begins to creep up at the boundaries due to the placement of a PML region outside the interior cells. Explicitly, the PML discrete functions are coupled with some of the interior ones, therefore the error propagates into the interior domain.

Quantity	Percentage
ΔE_x	2.149%
ΔE_y	2.133%
ΔH_x	8.632%
ΔH_y	8.588%
ΔH_z	13.803%

Table 3.1: Error percentages in electromagnetic quantities **E** and **H**.

Table 3.1 shows the total relative error across the entire grid. The relative error of our numerical solution with the harmonic magnetic dipole analytical solution seems to be reasonable. The error percentages in the magnetic field are noticeably higher because they are computed using numerical differentiation. This is because **H** is post processed using (1.6a). Though all these errors in the table above can be improved upon by employing an adaptive grid in the neighbourhood of source cells. We note here that accuracy of the forward model is important in context of the inverse problem (discussed in chapter 2). In particular, the electromagnetic quantities measured at the receiver locations must be accurate, as they underpin the gradient function (shown in 2.43). As discussed in chapter 2, the gradient function forms a crucial component in the optimization process for finding solutions to the inverse problem described in this thesis. In our experiments we use electric field loops as the electromagnetic measurements, meaning the post-processed **H** quantities are irrelevant in context of the inverse problem considered here. We also note here that if a joint inversion was to be considered as part of future work, involving both electrical conductivity and magnetic permeability, then we would require greater accuracy in **H**, as it would form the gradient function with respect to the magnetic permeability. For more information on error convergence and analysis of the FVM presented in this chapter, see [52].

CHAPTER 4

Numerical Results

FOLLOWING validation of the numerical solver for Maxwell's equations in the frequency domain, we are now in a position to find numerical solutions of the reconstruction schemes described in chapter 2. This chapter provides new numerical results of the novel reconstruction schemes described in chapter 2. The inversion routines are also written in Python alongside the forward solver.

This chapter is split into two main components. The first, section 4.1, shows numericals results for three inversions, each using different regularization schemes from chapter 2. These are: single level set inversion (algorithm 5), traditional $\mathcal{L}^2$-based inversion (algorithm 1) and sparsity regularized pixel-based inversion (algorithm 2). Some results in section 4.1 are also found in the publication written in section 1.3. The second component, beginning in section 4.2, shows results also for three inversions. These are: color level set inversion (algorithm 6), single level set inversion (algorithm 5) and traditional $\mathcal{L}^2$-based inversion (algorithm 1). Some of these results are also found in the submitted paper listed second in section 1.3. All numerical experiments here are designed for the near-field electromagnetic imaging problem described in section 1.2, though the reconstruction schemes themselves are not restricted to this setup.

4.1

Sparsity and single level set experiments

In this section, we describe the setup for numerical experiments involving algorithms 1, 2 and 5. We model a cube-shaped container by a domain $\Omega^h = [3] \times [3] \times [3]\mathrm{m}^3$. The domain Ω^h is divided into $[20] \times [20] \times [20]$ uniform rectangular grid cells, each of them having dimension $[0.15] \times [0.15] \times [0.15]\mathrm{m}^3$. Sources and receivers are located along two planes with constant z coordinates, opposing each other at both sides of the box (see figure 1.1). There are no sources or receivers facing the other four sides of the domain. We equip Ω^h with a homogeneous background conductivity, shielded walls on all six sides, and some unknown embedded object(s). The shielded walls have dimensions $[3] \times [3] \times [0.30]\mathrm{m}^3$ on the two sides parallel to the xy plane, and the same but shifted dimensions for the corresponding coordinates at the other four sides. These shielded walls are composed of material with conductivity $b = 0.1$. We choose to use two distinct phantoms; the first consists of two rectangular small cuboid inclusions embedded in the interior of our domain of interest, and the second is a horizontal torus shaped inclusion. We choose both phantoms to be composed of material with conductivity $b = 0.5$. The remaining space (not occupied by those inclusions) is filled with air, which is assumed here to have a homogeneous background conductivity $b_0 = 1 \times 10^{-8}$. A summary of chosen conductivity values for these numerical experiments is shown in Table 4.1.

Table 4.1: Conductivity parameters used for numerical experiments in section 4.1.

Domain Characteristic	b
Background	1×10^{-8}
Unknown Inclusions	0.5
Shields	0.1

Moreover, a single frequency $f = 1\mathrm{MHz}$ is used. We consider $(n_s, n_r) = (16, 100)$, where the distribution of these sources and receivers follow that in figure 1.1. Each source has dimension $[0.60] \times [0.60] \times [0.15]\mathrm{m}^3$ and each receiver has dimension $[0.15] \times [0.15] \times [0.15]\mathrm{m}^3$. Furthermore, we excite each source (electric wire loop) with an electric current $I_j = I = 0.1\mathrm{A}$.

The discretization of the Maxwell model in (1.7a, 1.7b) follows the scheme described in [6, 7, 51, 52] and is implemented in the programming language Python. See those references or the brief overview in chapter 3 for more details. The scheme discretizes a vector potential formulation of (1.7a, 1.7b) and integrates over finite volumes to find a discrete Maxwell system. We then solve the discrete Maxwell system iteratively using the BiCGSTAB algorithm, with tolerance 10^{-4}. A simple post-processing step is then performed to obtain the electromagnetic fields $\mathbf{E}$ and $\mathbf{H}$ from the vector potential.

In all numerical experiments considered here, including the next section, we use synthetically generated data using the same forward modelling scheme, but add 1% white Gaussian noise to the data before its use as data in the inversion process. Therefore, the true electromagnetic data $\mathbf{d}_j$ admits a decomposition

$$\mathbf{d}_j = \mathcal{M}_j \hat{\mathbf{E}}_j(\mathbf{x}) \left(1 + c\varepsilon_j\right),$$

where $\hat{\mathbf{E}}_j(\mathbf{x})$ is the electric field generated by the true phantom, $\varepsilon_j \sim \mathcal{N}(0,1)$, $c \in \mathbb{R}$ is a scaling parameter that dictates the noise level and $\mathcal{M}_j$ is defined in (2.22). The first set of numerical experiments assume the computational setup described above, and should be assumed for others unless otherwise stated. The numerical experiments in this section are structured as follows:

1. LK-Single level set reconstructions using algorithm 5;
2. LK-Pixel (traditional $\mathcal{L}^2$-based) reconstructions using algorithm 1;
3. LK-Sparsity reconstructions using algorithm 2.

For the pixel-based reconstructions, we display cross-sections of the recovered conductivity. For the level set reconstructions, we consider cross-sections and three-dimensional surface plots. Such surface plots are available for this technique as the interior conductivity is single-valued by definition of the level set method. Therefore, the recovered conductivity admits two values; the background and target conductivity. We use the line search criteria for the LK-Sparsity scheme once for each true phantom and use the resultant parameter τ_0 for all sparse reconstructions, with respect to each experiment. Moreover, we will show corresponding plots for the evolution of the step sizes τ_s, $\overline{N}$, if applicable, against the sweep number s and a pseudo cost measurement which we will discuss. We also provide comparisons of the three methods afterwards.

In more technical aspects, for the single level set inversion, we have chosen in these numerical experiments the intervals $I_{target} = [N_{inf}, N_{sup}]$ where $N_{inf} = 0$ and $N_{sup} = 25$, $I'_{target} = [\hat{N}_{low}, \hat{N}_{high}]$ where $\eta = 1/5$, $S = 200$, $N_{low} = \frac{1}{4} N_{sup}$ and $N_{high} = \frac{3}{4} N_{sup}$. These are our parameters for the dynamic intervals described in (2.64, 2.65). Bounds of the admissible interval range I_{target} are displayed in block black lines and bounds of the desired region I'_{target} are shown with dashed green lines.

The level set representation does not include the shielded regions. These are added to the conductivity profile once it has been defined by ϕ_1, for both data generation and when the level set functions are updated. We initialize ϕ_1 as an ellipsoid in the centre of the domain. This initial guess isn't optimized, since using other inversion techniques may give inference as to where initial guesses should be placed. But for our purpose, a centred initial guess for ϕ_1 suffices. This setup also applies to each level set function in the color level set regime described in section 4.2.1 onwards.

4.1.1

LK-Single level set reconstruction with correctly chosen contrast value

Shown in figure 4.1 are 2D slices of 3D LK-Single level set reconstructions for two separate phantoms. The top two rows display the true phantom and the reconstructed image, where the true phantom in this case resembles two isolated objects. The bottom two rows display the same, but in this case the true phantom resembles a torus. By observation, the level set scheme has managed to distinguish the two isolated objects and has recovered parts of the torus, demonstrating that the level set method is successful in this scenario. This is more apparent in the 3D surface plots shown in figure 4.2.

Figure 4.3 shows evolution of the average number of voxels $\overline{N}$ that change per sweep and its relationship with the step size τ_s, where in this case τ_s denotes the initial step size for the input to the backtracking line search function for each source update in the sth sweep. The algorithm attempts to keep $\overline{N} \in I'_{target}$ for each sweep s by adjusting τ_s according to the line search scheme discussed in section 2.1.3.2.

In the LK scheme considered here, we have immediate access to the residuals $\mathcal{R}_j[b_j]^s$ where s denotes the quantity at sweep s. However, to calculate the true cost after one complete LK sweep we need to compute $\sum_{j=1}^{n_s} \mathcal{R}_j[b_{n_s}]^s$, meaning we would have to compute $n_s - 1$ additional forward solves. To avoid these computational costs, we approximate the actual residual $\mathcal{R}[b]^s$ with a pseudo-residual. Mathematically speaking

$$\mathcal{R}[b]^s = \sum_{j=1}^{n_s} \mathcal{R}_j[b_{n_s}]^s \approx \tilde{\mathcal{R}}[b]^s = \sum_{j=1}^{n_s} \mathcal{R}_j[b_j]$$

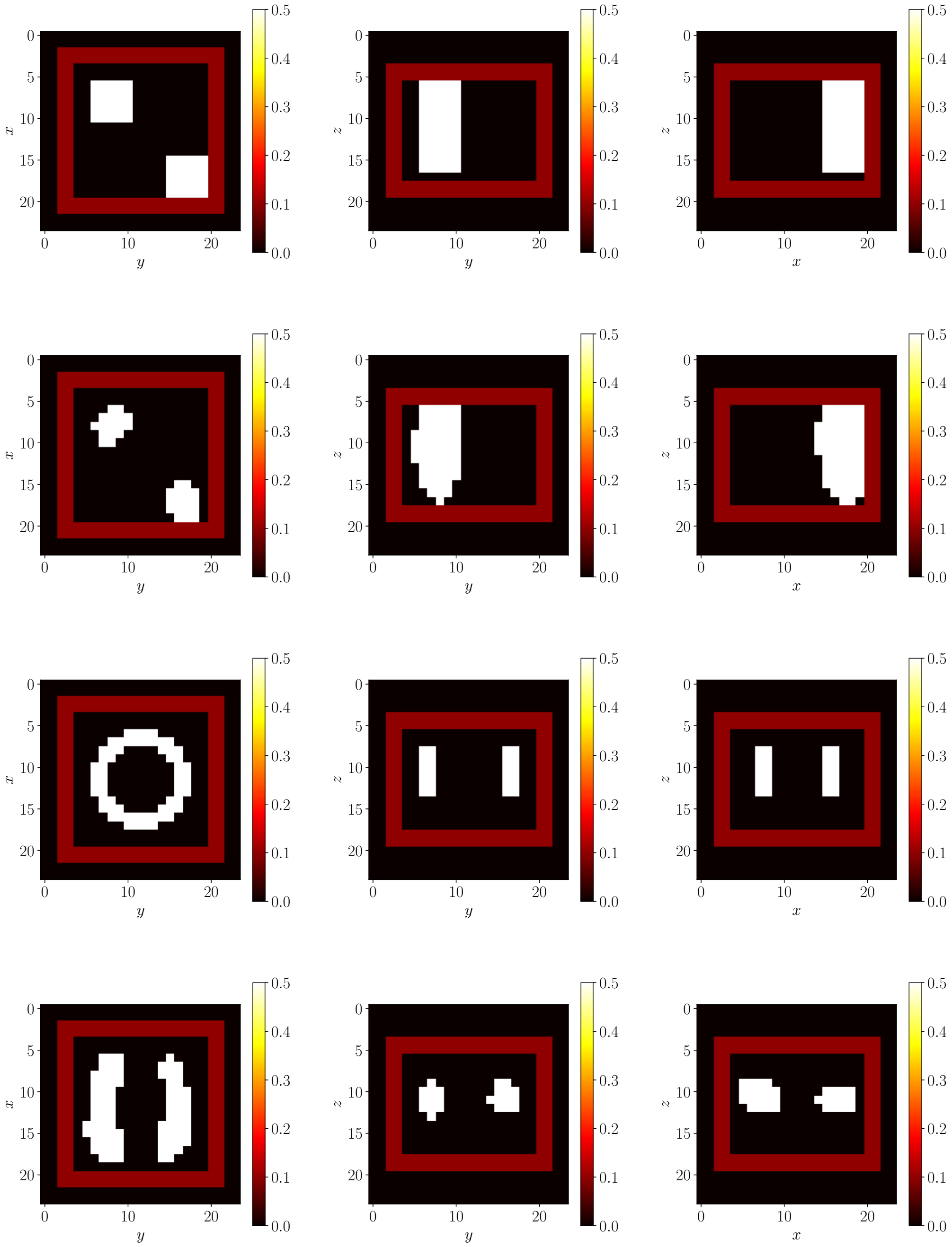


Figure 4.1: 2D cross-sections through two types of 3D LK-Single level set reconstructions.
Top row: 2D cross-sections through first true phantom. Left $z = 44$, middle $x = 17$, right $y = 26$.
Second row: The corresponding 2D cross-sections through the level set reconstructions at sweep number $s = 100$.
Third row: 2D cross-sections through second true phantom. Left $z = 40$, middle $x = 18$, right $y = 17$.
Fourth row: The corresponding 2D cross-sections through the level set reconstructions at sweep number $s = 100$.

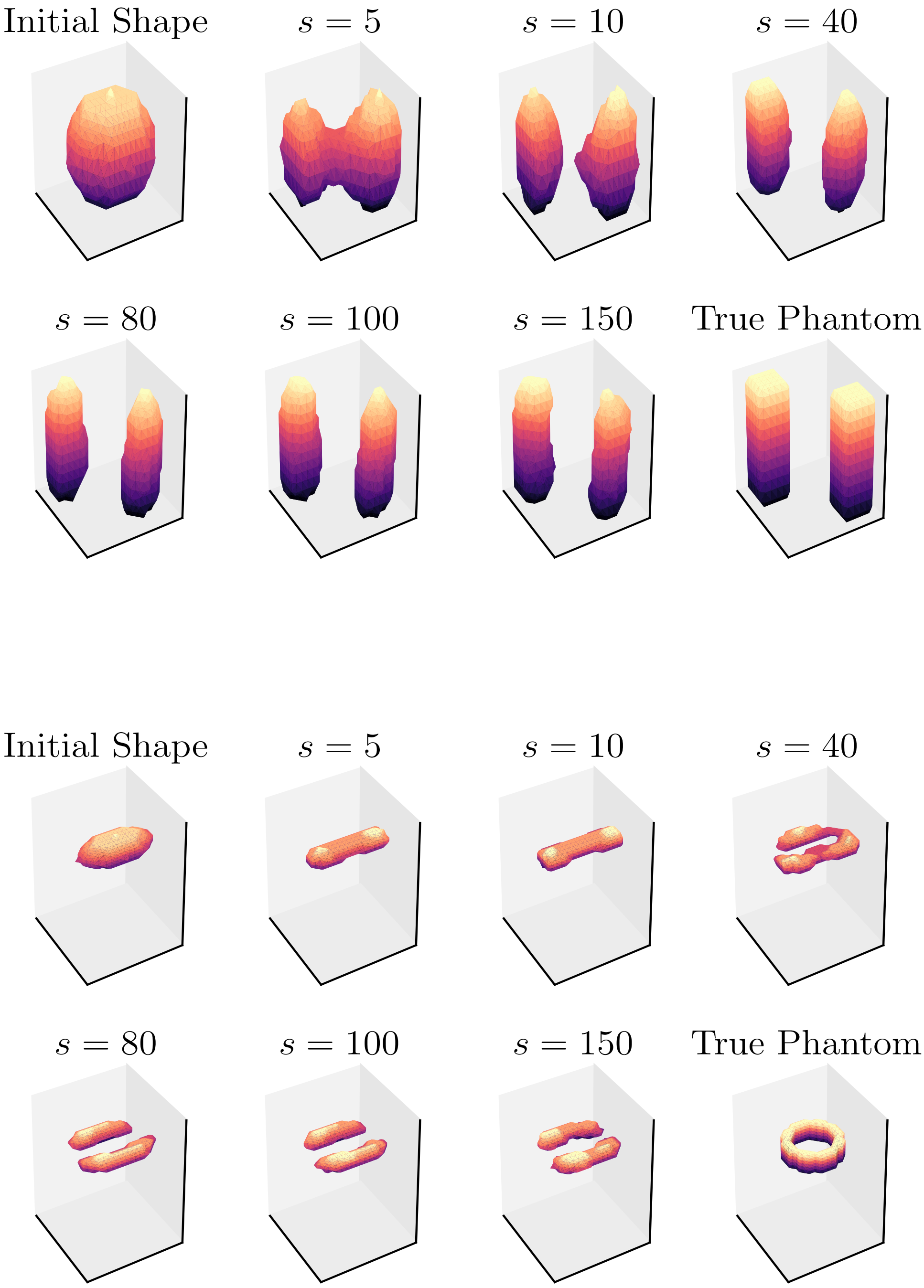


Figure 4.2: Surface plots of 3D shape evolution. Shown are (from left to right) the initial shape, snapshots at iteration numbers $s = 5, 10, 40, 80, 100, 150$, and the true phantom. Top two rows: for first true phantom; Bottom two rows: for second true phantom.

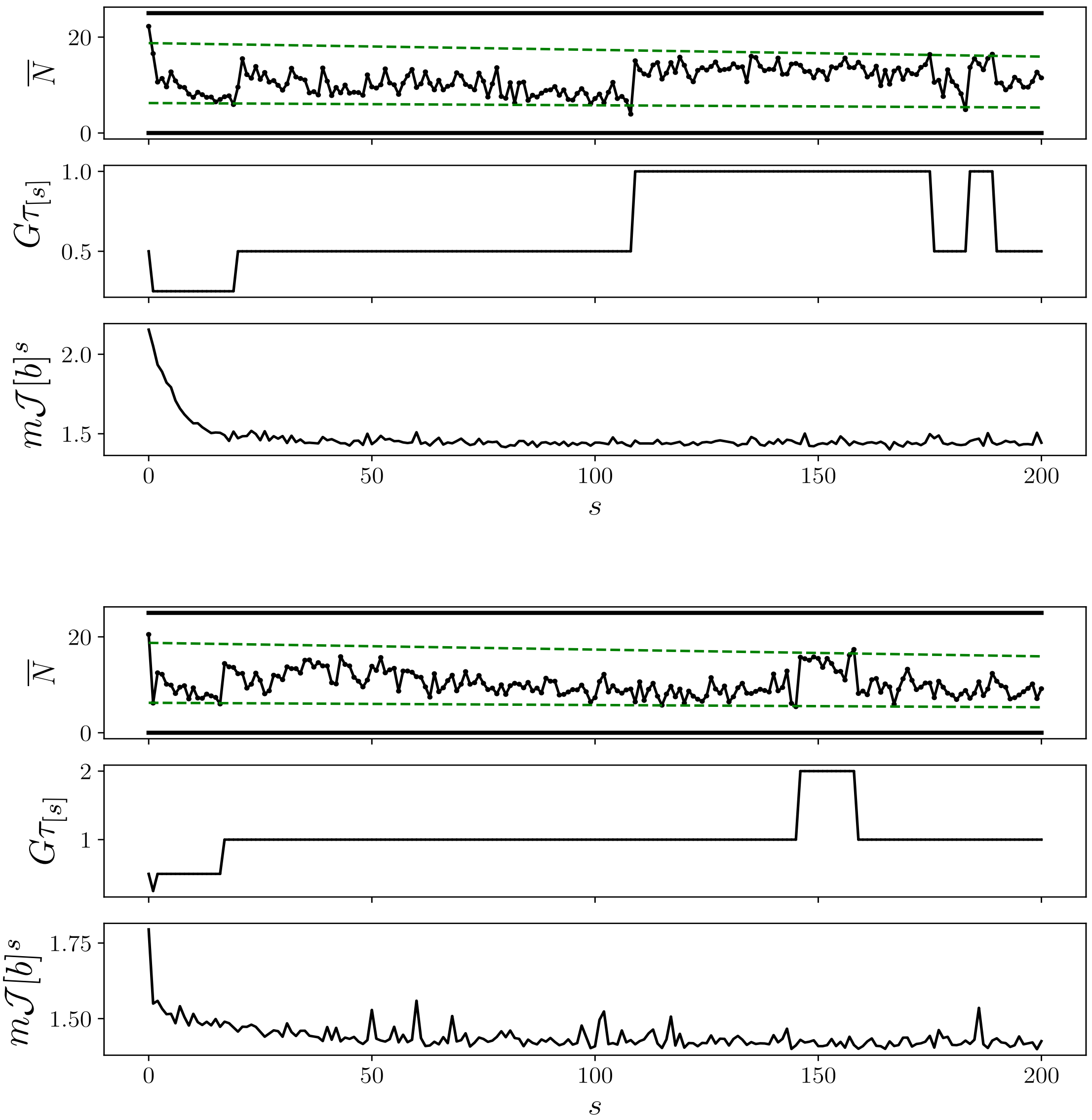


Figure 4.3: Evolution of line search parameters $\overline{N}, G\tau_s$ (Giga τ_s) and of the pseudo-cost $m\mathcal{J}[b]^s$ (part per thousand) against iteration number. Top: for true phantom (1); Bottom: for true phantom (2).

resulting in a pseudo-cost functional:

$$\tilde{\mathcal{J}}[b]^s = ||\tilde{\mathcal{R}}[b]^s||^2_{\mathcal{L}^2(\Omega)}. \tag{4.1}$$

This cost be computed *on the fly* per sweep, and is used throughout the numerical experiments as the cost measurement. From experiments, the true cost tends not to differ significantly to the quantity in (4.1) and can be viewed as a lagged cost measurement. For that reason, we label the cost plot using $\mathcal{J}$, when in reality we are actually computing that in (4.1). By observation of figure 4.3, we see that the pseudo-cost converges as the sweeps progress.

4.1.2

LK-Single level set reconstruction with incorrectly chosen contrast value

In the first two numerical experiments described in figures 4.1-4.3, the level set method performs well in recovery of location and shape under the assumption that interior and exterior values of conductivity are known. Let us now test how stable level set reconstructions are when some *a priori* information is incorrect in this application. We consider a level set representation of the conductivity whose contrast values are incorrectly prescribed. We model this as:

$$\tilde{b}(\phi)(\mathbf{x}) = wb^i(1-H(\phi)) + b^e H(\phi), \tag{4.2}$$

where $w \in \mathbb{R}^+$ indicates the deviation factor from the correct value of the internal conductivity. Firstly, we generate electromagnetic data for the true phantom with conductivity b^i for objects and b^e for the background. Then, we apply the LK-Single level set algorithm with an incorrectly chosen interior value wb^i throughout the reconstruction. We refer to $w > 1$ and $w < 1$ as overestimating and underestimating respectively.

Figure 4.4 shows behaviour of the LK-Single level set scheme under an incorrectly chosen interior value for both true phantoms considered in the previous section. For this experiment, we choose the level set representation given in (4.2) with $w = 1/2$, to represent the conductivity profile. By observation of this 3D surface plot representation of the conductivity shown in figure 4.4 we see that the algorithm responds to the modelling error of underestimating b^i by recovering the inclusions with a similar shape as the correct one, but slightly 'inflated', i.e. covering a larger volume. This is still acceptable in situations where the main goal is to identify and locate certain target objects, such as threats hidden in boxes or containers or certain objects hidden in caves or rooms.

Figure 4.5 show behaviour of the 3D LK-Single level set reconstruction scheme with an overestimating parameter $w = 2$. By observation, we see a reversal of behaviour compared with $w = 1/2$. In particular, the algorithm recovers smaller 'deflated' objects in response to an incorrectly overestimated interior conductivity value. Though the general location of embedded objects were recovered correctly. However, due to smaller reconstructions and in compensation to the incorrectly assumed higher internal conductivity value, much of the shape (material) is missing.

In summary, the LK-Single level set algorithm performs well when the interior conductivity value is approximately consistent with that of the true phantoms. The location and shape are reasonably accurate, given shielding and low frequency f. From this, it is clear that the algorithm can detect enough information in the electromagnetic data to reconstruct such images. We have demonstrated that it can reconstruct multiple objects with complex topologies. Furthermore, the algorithm still performs well under slightly incorrect assumptions on the interior conductivity values. The chosen scaling parameter w was small and constant but it is encouraging that the algorithm has robustness in these situations. Whilst this is encouraging for stability in single level set reconstruction schemes, we demonstrate that this behaviour can lead to further complications when considering a more realistic *a priori* conductivity profile.

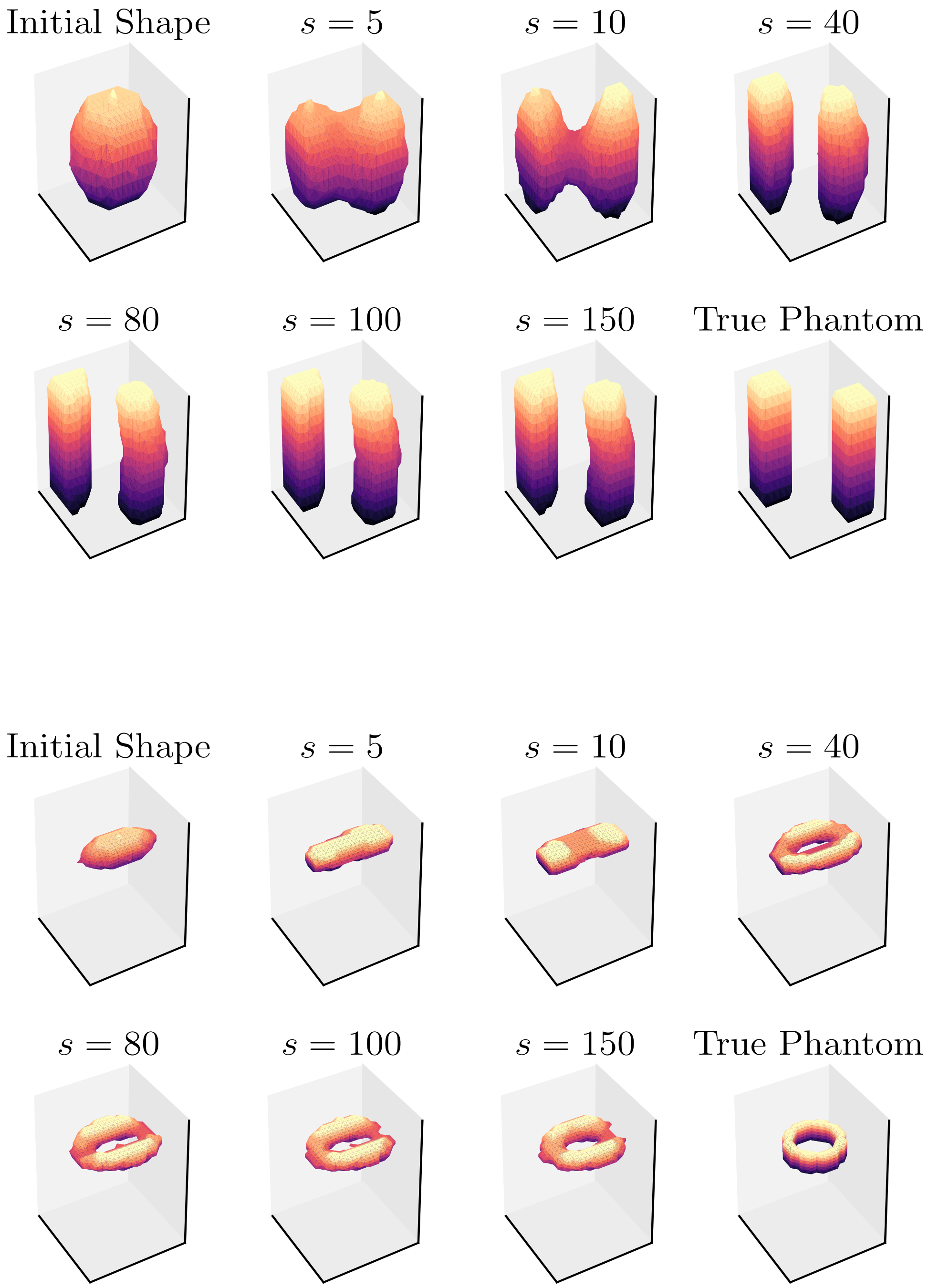


Figure 4.4: Surface plots of 3D shape evolution with $w = 0.5$. Here the internal conductivity value is assumed to be $b = 0.25$ during the reconstruction instead of the correct value $b = 0.5$. Shown are (from left to right) the initial shape, snapshots at iteration numbers $s = 5, 10, 40, 80, 100, 150$, and the true phantom. Top two rows: first experiment; Bottom two rows: second experiment.

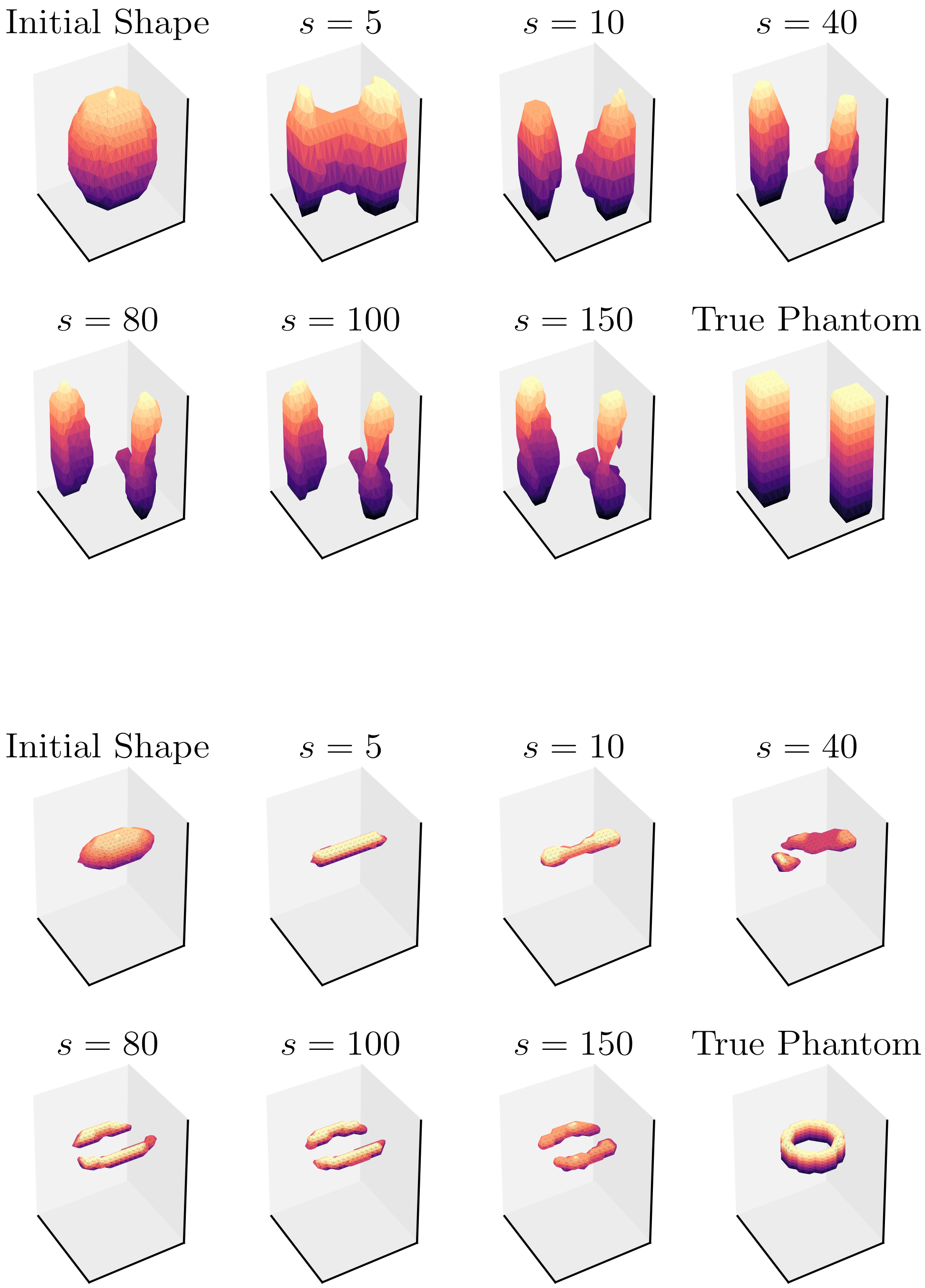


Figure 4.5: Surface plots of 3D shape evolution with $w = 2$. Here the internal conductivity value is assumed to be $b = 1.0$ during the reconstruction instead of the correct value $b = 0.5$. Shown are (from left to right) the initial shape, snapshots at iteration numbers $s = 5, 10, 40, 80, 100, 150$, and the true phantom. Top two rows: first true phantom; Bottom two rows: second true phantom.

4.1.3
Pixel-based $\mathcal{L}^2$-reconstruction of conductivity profile

Figures 4.6 and 4.7 show $\mathcal{L}^2$-based reconstructions of both true phantoms using algorithm 1. By observation, even though they provide some indication of the presence and location of the two isolated objects in one of the experiments, the reconstructions are worse when compared with the LK-Single level set reconstruction. However, the level set approach uses stronger assumptions on the nature of conductivity profile inside the domain of interest, which is rewarded with better estimates of shapes.

Figures 4.8 and 4.9 show LK-Sparsity reconstructions of both true phantoms using algorithm 2. We note that the gradient of the conductivity between neighbouring cells in non zero regions increases as one reduces the threshold parameter. This behaviour has its drawbacks; as can be seen in the recovery of the torus. Due to compounded discardment of the gradient, we see that topology of the reconstruction has changed (i.e. no holes). In both reconstructions, as we reduce the thresholding term, the reconstructions become increasingly similar to results of the standard $\mathcal{L}^2$-based approach (shown in figure 4.6). This is to be expected, as LK-Sparsity regularizes the pixel-based problem. Lowering the threshold parameter puts greater emphasis on minimizing the data misfit term in the cost functional. Insisting on a sparse conductivity profile can allow the recovered shape to be distorted, therefore care must be taken when choosing the threshold parameter. Figure 4.10 shows two comparisons. The first is between the data misfit and the sparsity level Λ and the second each reconstruction scheme and their data misfits. As we reduce the sparsity thresholding term, we see that the data misfit is shifted by a factor downward. This is due to the scheme putting less emphasis on minimizing the second term in the sparsity cost functional. In comparison with the reconstructions in figures 4.8 and 4.9, this is consistent; as increasing the threshold resulted in the reconstruction resembling the true phantom less. As we lower the threshold parameter, the cost curve shifts closer to the traditional $\mathcal{L}^2$-based cost.

The comparison between reconstruction schemes is interesting, as different regularization have produced similar data misfit values. For example, LK-Single level set and LK-Pixel have an almost identical data misfit value for the torus, even though the techniques and reconstructions are wildly different.

At the bottom of figure 4.10 shows a line plot of the diagonal in each reconstruction at the stated z. It highlights the behaviour of each reconstruction scheme compared to the true phantom. By observation of the line plots for the first true phantom, we can see peaks

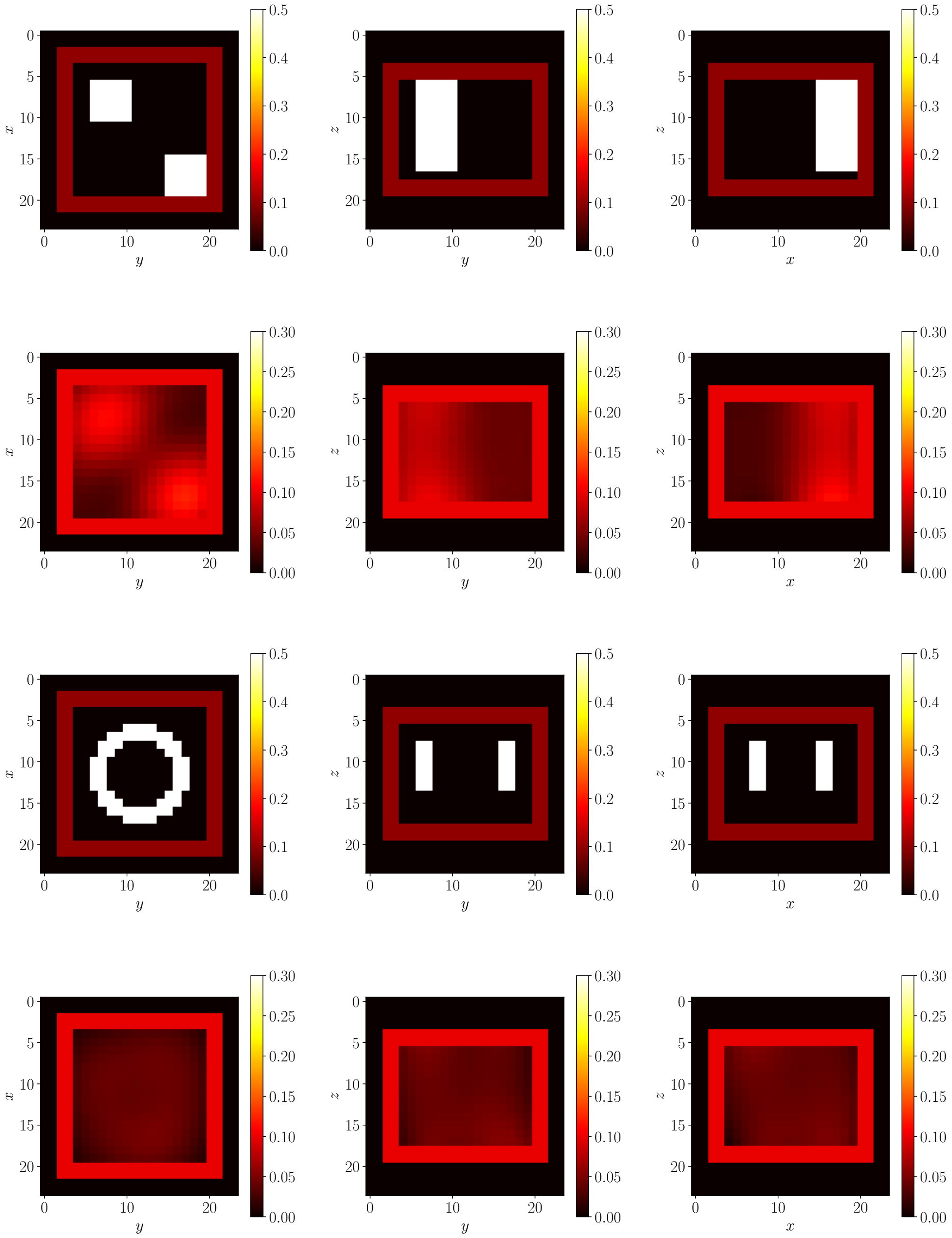


Figure 4.6: 2D cross-sections of 3D $\mathcal{L}^2$-based reconstructions for two true phantoms.
Top row: 2D cross-sections of first true phantom. Left $z = 44$, middle $x = 18$, right $y = 27$.
Second row: 2D cross-sections of $\mathcal{L}^2$-based 3D reconstruction for first true phantom, at sweep $s = 100$.
Third row: 2D cross-sections of second true phantom. Left $z = 40$, middle $x = 18$, right $y = 17$.
Fourth row: 2D cross-sections of $\mathcal{L}^2$-based 3D reconstruction for second true phantom, at $s = 100$.

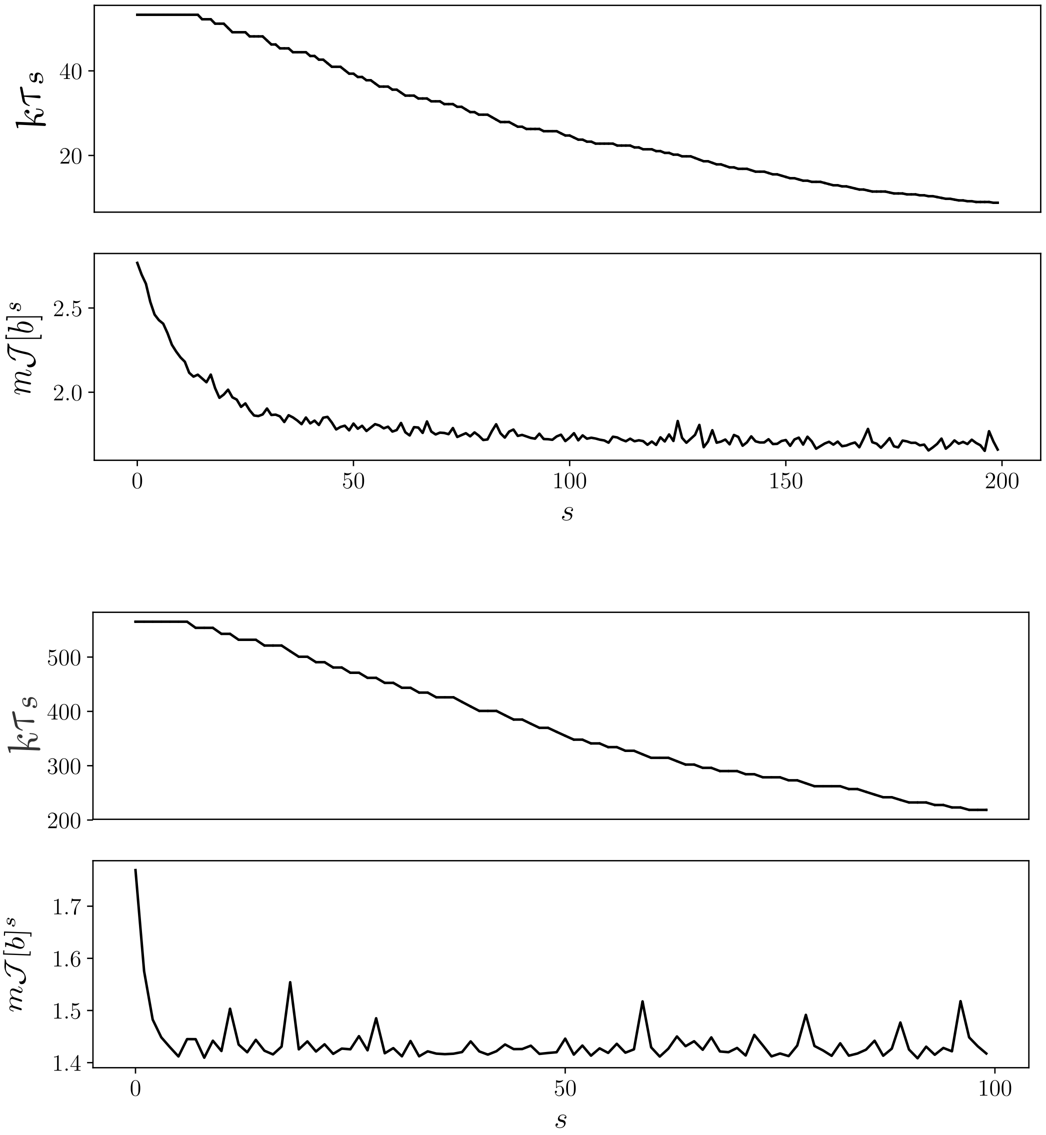


Figure 4.7: Evolution of line-search parameter $k\tau_s$ (kilo τ_s) and the pseudo-cost $m\mathcal{J}[b]^s$ against sweep number s for the $\mathcal{L}^2$-based reconstruction. Top: for first true phantom; Bottom: for second true phantom.

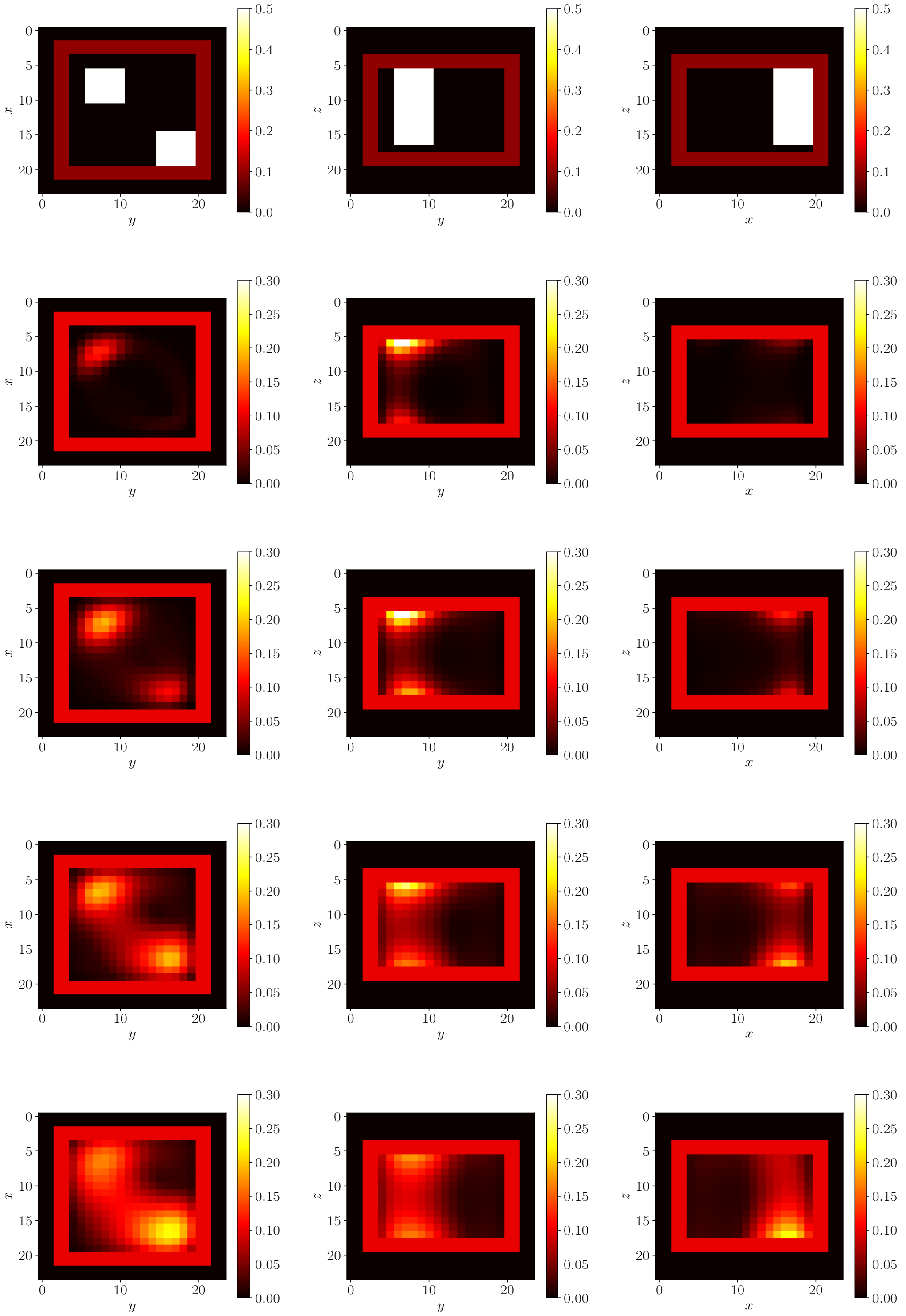


Figure 4.8: Top: 2D cross-sections of first true phantom. Left $z = 44$, middle $x = 17$, right $y = 26$. Second - Fifth row: 2D cross-sections of a LK-Sparsity reconstruction, at $s = 100$ for various (τ, d_1).
Second Row: $(\tau, d_1) = (7.7413 \times 10^5, 1.20)$ Third Row: $(\tau, d_1) = (7.7413 \times 10^5, 0.95)$ Fourth Row: $(\tau, d_1) = (7.7413 \times 10^5, 0.80)$ Fifth Row: $(\tau, d_1) = (7.7413 \times 10^5, 0.60)$.

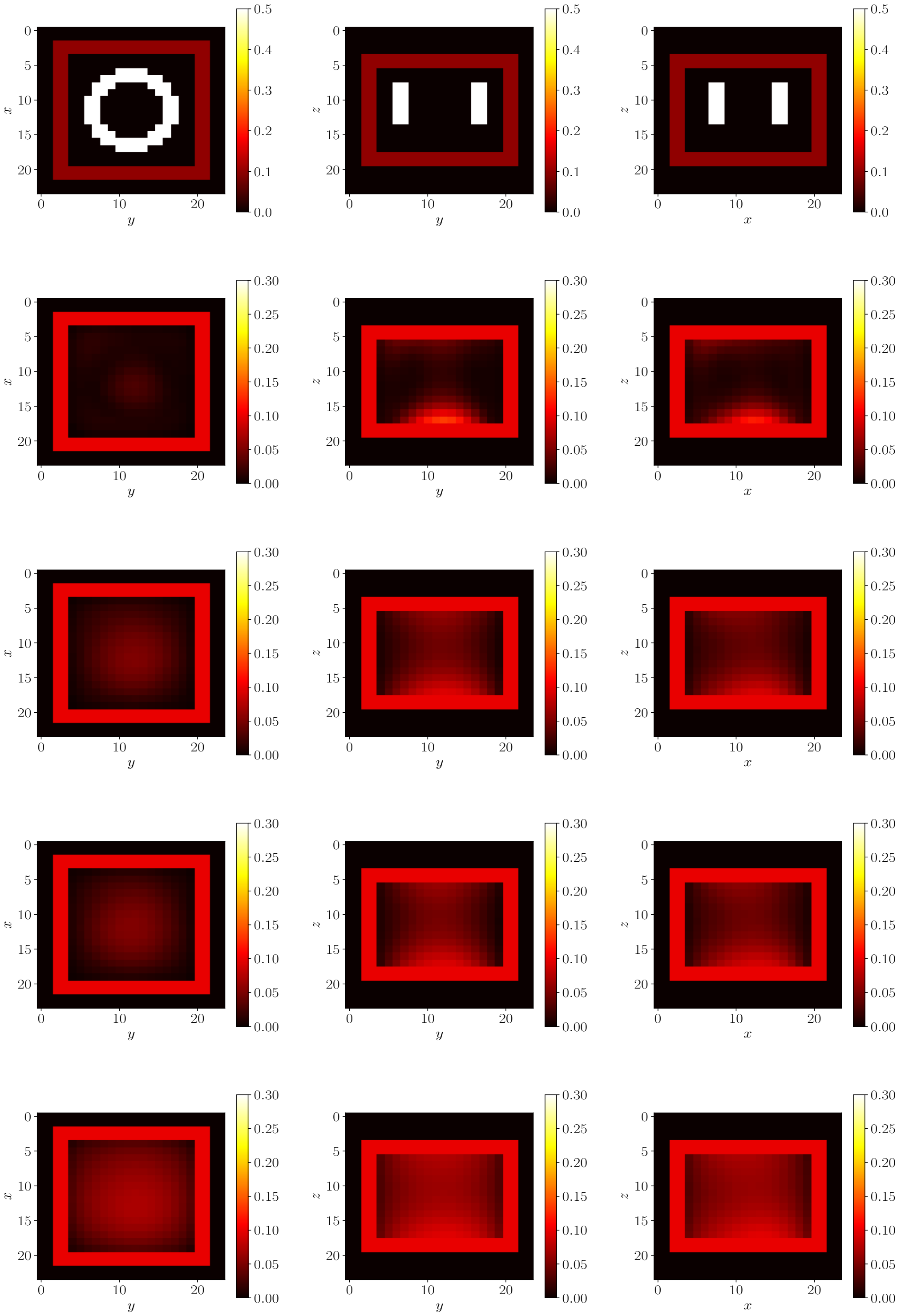


Figure 4.9: Top: 2D cross-sections of second true phantom. Left $z = 40$, middle $x = 18$, right $y = 17$. Second - Fifth row: 2D cross-sections of a LK-Sparsity reconstruction, at $s = 100$ for various (τ, d_1).
Second Row: $(\tau, d_1) = (8.1413 \times 10^4, 0.60)$ Third Row: $(\tau, d_1) = (8.1413 \times 10^4, 0.40)$ Fourth Row: $(\tau, d_1) = (8.1413 \times 10^4, 0.20)$ Fifth Row: $(\tau, d_1) = (8.1413 \times 10^4, 0.10)$.

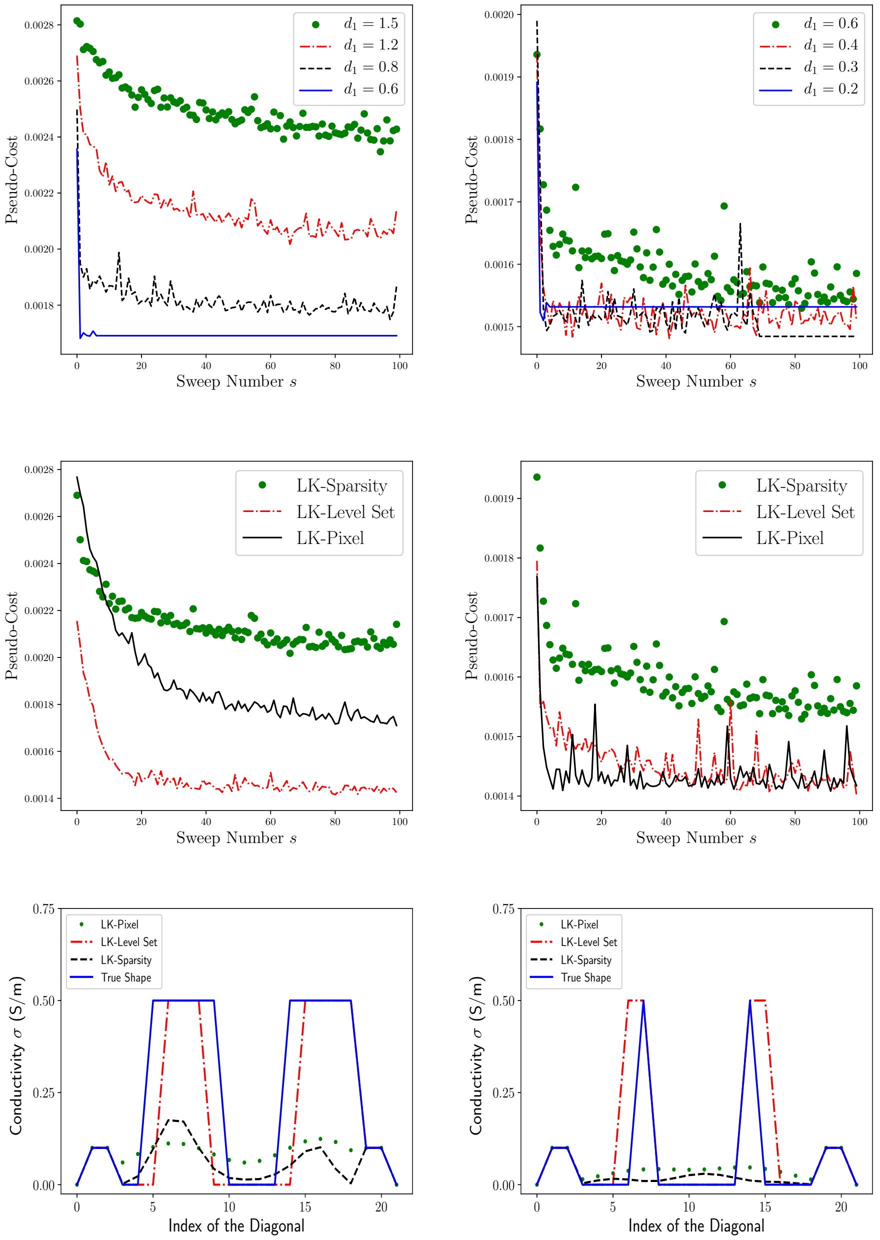


Figure 4.10: Comparison of LK-Single level set, LK-Pixel and LK-Sparsity methods.
Top Row: Evolution of the pseudo-cost for both experiments (Left: isolated objects, Right: torus) with varied sparsity level d_1.
Second Row: Evolution of the pseudo-cost for methods LK-Sparsity, LK-Single level set and LK-Pixel for both true phantoms (Left: isolated objects, Right: torus).
Bottom Row: Line plots of the diagonal for constant z for both experiments (Left: isolated objects ($z = 44$), Right: torus ($z = 40$).

in the same region as the true phantom across all three reconstruction schemes, with LK-Sparsity attempting to zero fill where there is no object. The smooth behaviour of the LK-Pixel scheme is also clear to see. Recovering the torus is more difficult for all three reconstruction schemes, with both LK-Pixel and LK-Sparsity struggling to recover where the true object is located. We also see LK-Sparsity being strictly lower than LK-Pixel in this line plot.

In this section, we have shown two numerical experiments involving two new reconstruction algorithms for imaging shielded containers using near-field electromagnetic data. A comparison of the performance for our two proposed reconstruction schemes was made with an established scheme. The first new algorithm solves a shape-based EM inverse problem by adapting general ideas described in [35, 39] for application in this new challenging situation. The second one solves a pixel-based inverse problem but promotes sparsity in the solution through regularization. The method used here follows that proposed in [100] (for a different 2D imaging modality) and extends it to 3D imaging from near-field electromagnetic data.

The performances of all three algorithms are promising for constructing a general purpose algorithm. However, when *a priori* knowledge is available, indicating that inclusions with small compact support and significantly higher internal conductivity value are present in the domain of interest, then both the sparsity regularized and the level set based shape evolution approach perform better than the standard $\mathcal{L}^2$-based scheme. This is true even in situations where the $\mathcal{L}^2$-based scheme reduces the least squares data misfit functional to the same extent as the other two schemes. Due to the additional constraints included in the sparsity and level set based algorithms, reconstructions are more likely to yield results which are in agreement with such prior information.

4.2

Color level set experiments

The experiments outlined in the previous section are encouraging for building a general purpose algorithm in this application aswell as motivating research into more complicated scenarios. In this section, we perform numerical experiments using the novel reconstruction scheme outlined in algorithm 6.

Here, we introduce the setup for numerical experiments involving the color level set regime as introduced in chapter 2.We use the LK-Color level set scheme, as described in algorithm 6, for the near-field electromagnetic imaging problem described in section 1.2. We choose to model such a region as cube-shaped by a domain $\Omega^h = [3] \times [3] \times [3] m^3$. We then divide this region into uniform rectangular cells; each with dimension $[0.15] \times [0.15] \times [0.075] m^3$. Sources and receivers are located in the same manner as the first set of numerical experiments. Here we use algorithm 6 for two different experiments, both of which involve placing two objects with differing constant conductivities in a background conductivity profile resembling air. The first is an object embedded inside another, whereas the second resembles two isolated objects. In both numerical experiments, we choose the unknown inclusions to be shielded by a cage with conductivity $b = 0.1$. Moreover, we consider $n_s = n_r = 16$ with the source and receiver distributions following the setup in figure 1.1 and a frequency $f = 1$MHz. Each source and receiver has dimension $[0.6] \times [0.6] \times [0.075] m^3$, with the sources being excited with an electric current $I_j = I = 0.1A$. The conductivities used in the two numerical experiments are shown in Table 4.2. In context of the imaging problem, b^3 is considered to represent conductivity of air and b^1, b^2 represent bodies of water with differing conductivities. This particular setup, for example, could resemble soil and biological material (e.g. meat) enclosed in wooden boxes. For other high-contrast situations, such as in industrial process control, where you want to monitor pipes containing various liquids and gases with differing conductivities, color level set methods as introduced here can be adapted to represent these situations. For more information on those applications, see [79, 128].

Table 4.2: Conductivity parameters used for numerical experiments in section 4.2.

	b^1	b^2	b^3
Numerical Experiment 1	0.5	10.0	1×10^{-8}
Numerical Experiment 2	0.5	1.0	1×10^{-8}

The conductivity values shown in table 4.2, as well as those described in table 4.1, represent high-contrast conductivity values. Both scenarios are particularly challenging as the high-contrast values distort the sensitivity functions. Moreover, due to the smooth behaviour of the $\mathcal{L}^2-$based gradient (which is composed of sensitivity functions), it is difficult to build a high-contrast interface with a finite number of smooth updates. For further analysis, we refer to [37] and for other imaging applications involving color level set inversion for high-contrast multi-phase materials see for example [3].

4.2.1

LK-Color level set inversion for Numerical Experiment 1

In the first numerical experiment we have chosen $I^m_{target} = [0, 26]$, and in the second $I^m_{target} = [0, 117]$. Note that these numbers for N^m_{sup} are not arbitrary; they are chosen as a percentage of the total number of interior cells. Furthermore, we choose $\eta_m = 1/5$, $S = 200$ and $[N^m_{low}, N^m_{high}] = [\frac{1}{4}N_{sup}, \frac{3}{4}N_{sup}]$ for both experiments. Note that these intervals are constant with respect to m, but other choices are also possible. These are the parameters that define the dynamic intervals in (2.64, 2.65) for the line search criteria. Along with images of the color level set reconstruction, we also display information on the line search criteria. We display the admissible interval range $\hat{I}^m_{target}$ as block black lines and bounds of the desired region $\hat{I}'^m_{target}$ are shown in dashed green lines. Figure 4.11 shows 2D cross-sections of a 3D LK-Color level set inversion involving two level set functions for Numerical Experiment 1. Shown are various snapshots of the shape evolution from the top (initial guess) to the bottom (true phantom). In addition to figure 4.11, figure 4.12 shows a 3D view of the reconstruction at various sweep snapshots. From both of these figures, we observe that as the algorithm converges (by the pseudo-cost metric), the reconstruction manages to capture the main topological characteristics of the true phantom. This is quite impressive considering that each of the three conductivities present in the true phantom are high-contrast with respect to each other. Moreover, we have used a small amount of data in comparison to the grid size (the number of data points is approximately 1% of the total number of unknowns), and only one fixed frequency. This could possibly be improved by using information from a broadband of frequencies for the reconstruction task. This would involve incorporating an additional loop over a set of frequencies into the reconstruction schemes.

Figure 4.13 shows reconstruction information of Numerical Experiment 1. In the top figure, we observe the average number of voxels that change each sweep for both level set functions (yellow dotted line: ϕ_1, black dotted line : ϕ_2). The top black line descends as the algorithm progresses, making the admissible interval smaller. This induces $\hat{I}'^m_{target}$ to also be smaller, meaning acceptance of an update is more difficult. We see that the average number of voxels changing hands reduces over the lifespan of the reconstruction scheme. This is a consequence from the descending linear lines which form from the dynamic acceptance interval.

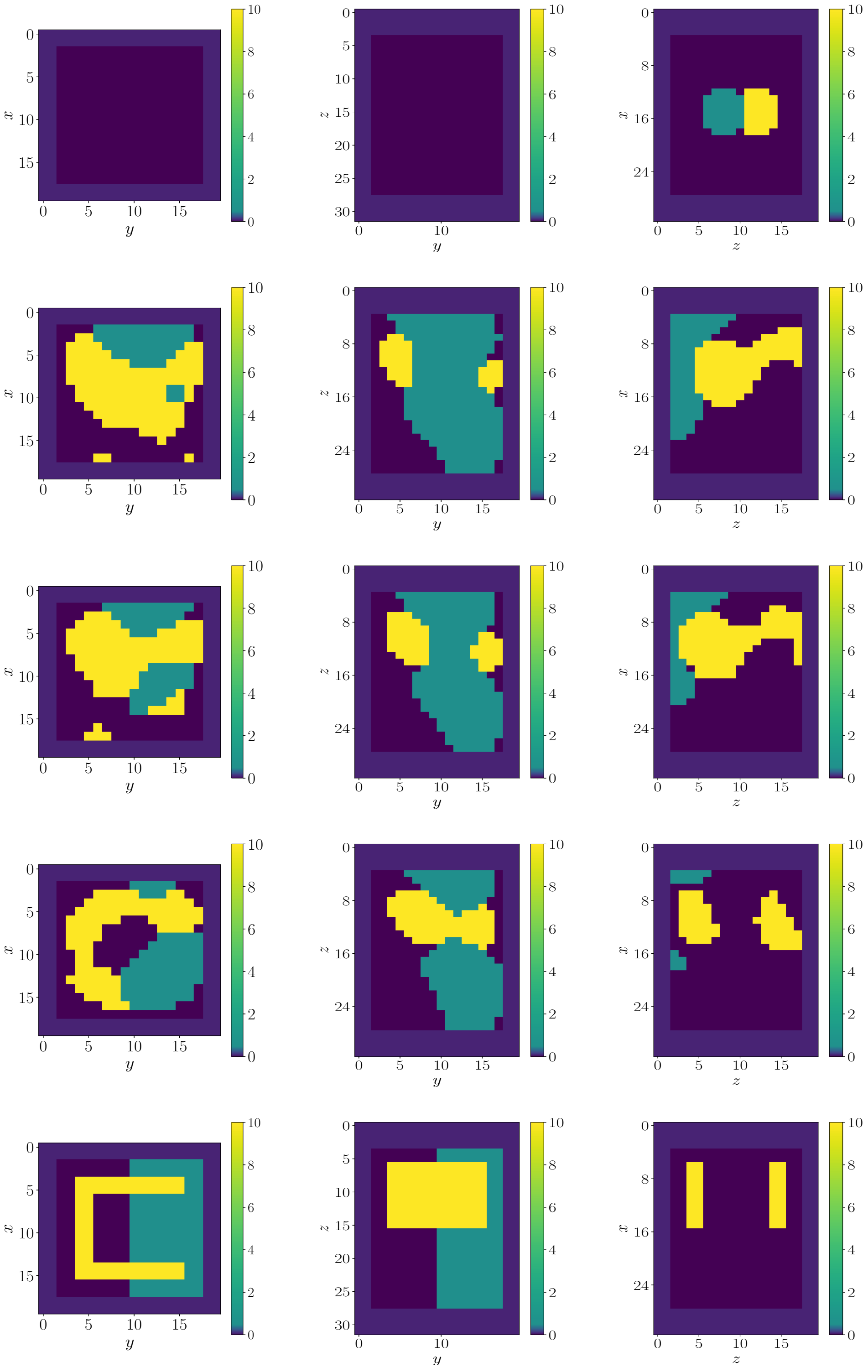


Figure 4.11: 2D cross-sections through a 3D LK-Color level set reconstruction for Numerical Experiment 1.
Each row: Left $z = 25$, middle $x = 15$, right $y = 18$.
1st row: Initial guess, 2nd row: $s = 60$ sweeps, 3rd row: $s = 125$ sweeps, 4th row: $s = 500$ sweeps and 5th row: true phantom.

Analogous to that in (4.1), we use a pseudo-cost which is cheap to compute. The pseudo-cost in the level set case is defined as

$$\tilde{\mathcal{J}}[\Phi] = \sum_{j=0}^{n_s-1} \mathcal{J}_j[\Phi_j], \quad \mathcal{J}_j[\Phi] = ||\mathcal{T}_j(\Phi)||^2_{\mathcal{L}^2(\Omega)},$$

and can be viewed as a lagged cost. It is significantly less expensive to compute than the actual cost after each sweep, since it can be calculated during the inversion. Significant differences between the pseudo and true cost have not been observed in the level set regime when choosing to calculate both in experiments. In this particular example, observing the middle figure, we see that the algorithm has managed to find a step size which allows $\overline{N}^m \in \hat{I}'^m_{target}$ for the majority of sweeps. The bottom figure shows the pseudo-cost each sweep, which decreases as the algorithm progresses, meaning the error between data generated by the reconstructed shape and the true shape is becoming smaller.

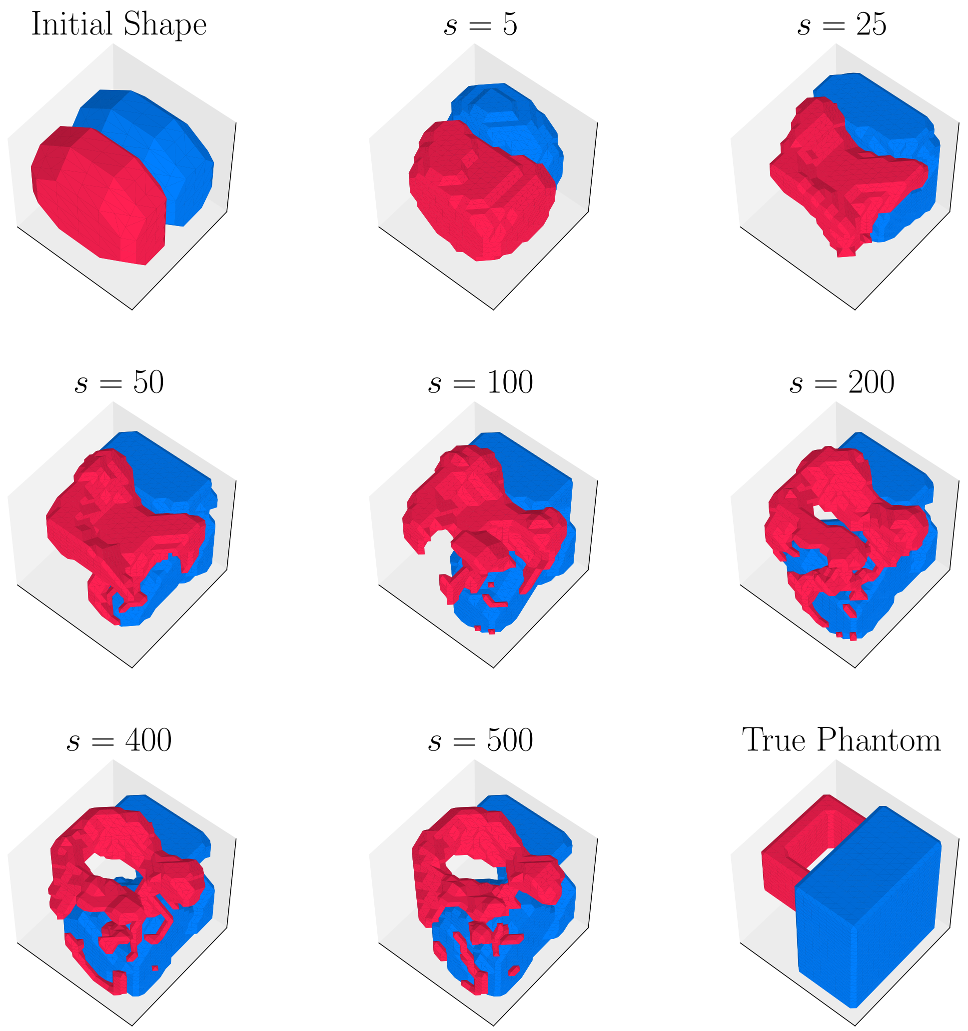


Figure 4.12: Surface plots of 3D shape evolution when using LK-Color level set reconstruction scheme with two level set functions for Numerical Experiment 1. Shown are the initial shape, snapshots at sweep numbers $s = 5, 25, 50, 100, 200, 400, 500$ and the true phantom. The colour red indicates the shape of the conductivity b^2 and blue indicates the shape of the conductivity b^1.

It appears the level set inversion algorithms compensate for false interior values by growing or shrinking the corresponding interior shape, depending on whether the false values were an under or over estimation of the true interior conductivity. The experiments in [58], also presented in section 4.1.3, suggest that a small object with high conductivity is roughly equivalent to a large object with low conductivity in the data misfit. We will also observe this behaviour when using the single level set inversion for Numerical Experiment 1. It is tricky to say whether this occurs in the color level set regime since this scheme can adjust multiple domains to fit the data.

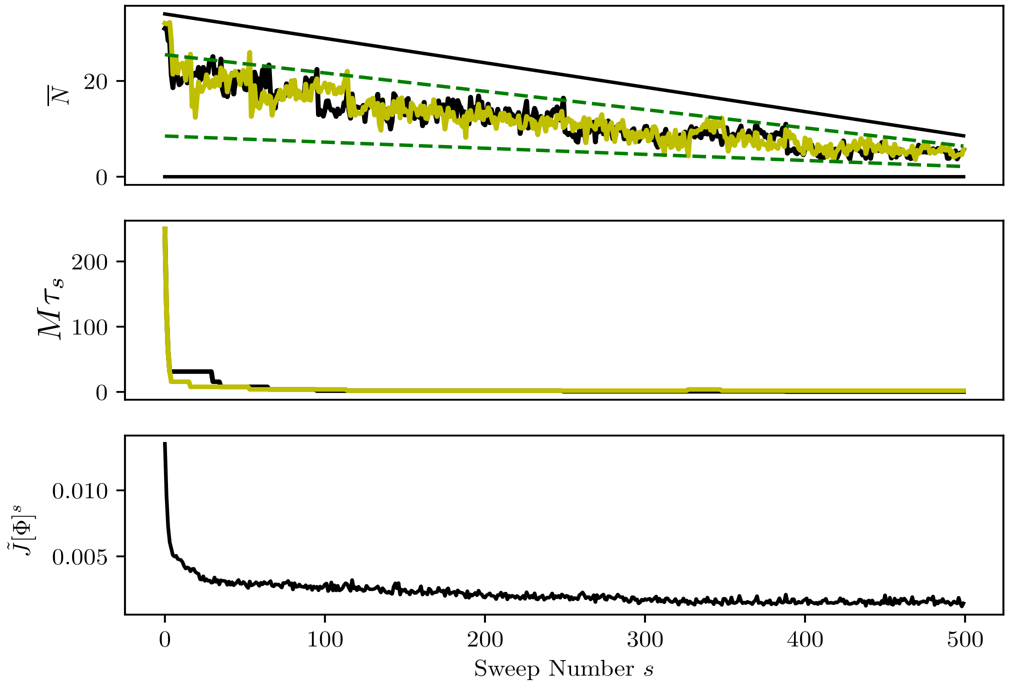
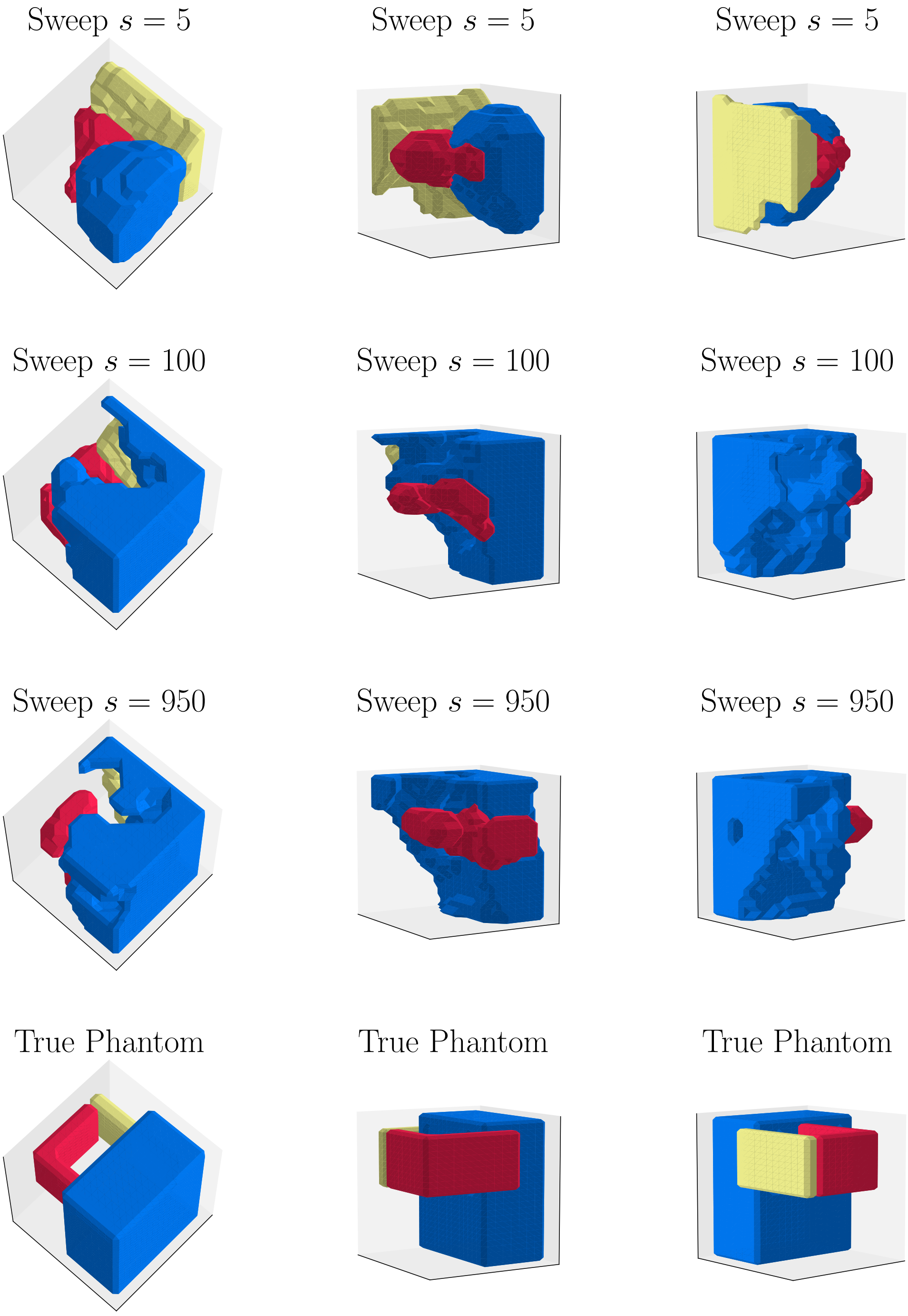


Figure 4.13: Line search analysis of LK-Color level set reconstruction for Numerical Experiment 1. Top Row: Evolution of the average number of voxels $\overline{N}$ as a function of the sweep number s. The dotted yellow line represents $\overline{N}_s^1$ and the black line represents $\overline{N}_s^2$. Middle Row: Evolution of the step sizes τ_s^1 and τ_s^2 each denoted by yellow and black dotted lines respectively. Bottom Row: Evolution of pseudo-cost $\tilde{\mathcal{J}}[\Phi]^s$ as a function of the sweep number s.

Figure 4.14 shows surface plots from a LK-Color level set inversion involving three level set functions for a slightly altered version of the true phantom in Numerical Experiment 1, where $b^1 = 0.5$, $b^2 = 10.0$, $b^3 = 50.0$ and $b^4 = 1 \times 10^{-8}$. The level set formulation, which in this case involves three level set functions, follows the setup described in (2.70). We introduce an additional conductivity to make it more difficult for the algorithm to recover the present high contrast materials. From observation of the snapshots, the inversion scheme manages to recover locations of b^1, b^2 and b^4 reasonably well. However, b^3 has shrunk significantly in the final reconstruction. Whilst some of the shape is still present, b^1 is larger than in the true phantom therefore it could be that this has a similar misfit to the data as having a larger shape associated with b^3, for reasons already discussed.



Sweep $s = 5$ Sweep $s = 5$ Sweep $s = 5$

Sweep $s = 100$ Sweep $s = 100$ Sweep $s = 100$

Sweep $s = 950$ Sweep $s = 950$ Sweep $s = 950$

True Phantom True Phantom True Phantom

Figure 4.14: Surface plots of 3D shape evolution using LK-Color level set reconstruction scheme with three level set functions for altered Numerical Experiment 1. Shown are sweep snapshots at iteration numbers $s = 5, 100, 950$ and the true phantom, for three orientations. The colour red indicates the shape of the conductivity b^2, blue indicates the shape of the conductivity b^1 and yellow indicates the shape of the conductivity b^3.

4.2.2

Comparison of regularization schemes

Figure 4.15 shows a comparison of the color level set inversion scheme with other reconstruction algorithms for Numerical Experiment 1. In particular, we compare the LK-Color level set scheme with two variants of a single level set inversion and a traditional $\mathcal{L}^2$-pixel based scheme. The two single level set inversions differ in their *a priori* information; the first assumes that the interior conductivity value in the single level set inversion, which we label b^i, is an arithmetic average of the two present in the true phantom, whereas the second assumes that $b^i = b^2$. The exterior background, b^e, is assumed to be equal to b^3 in both these cases.

The traditional $\mathcal{L}^2$-based scheme performs reasonably well given that the true phantom has relatively high contrast between conductivity domains. The scheme itself promotes smooth conductivity profiles whereas in contrast the level set inversion schemes promote non-smooth conductivity profiles with sharp edges. Given correct *a priori* information on the conductivity value, level set inversion can resolve shape more clearly. However, in this case, when false *a priori* assumptions exist, the traditional $\mathcal{L}^2$-pixel scheme resolves shape better. This is quite impressive considering that the *a priori* information given to the pixel-based schemes assumes less than that of the level set inversion methods. In part, this happens because the single level set inversion has been penalized with false assumptions on the interior conductivity values. More complicated scenarios such as the true phantom for Numerical Experiment 1 do not lend themselves well to the single level set inversion, since fundamentally the method can only recover shape of two parameter domains. Therefore, we are forced to provide false *a priori* information on the conductivity profile, since in some sense the single level set inversion has incompatibility in these scenarios.

As shown in [58], and in section (4.1.1), level set inversion tends to grow and shrink domains to fit the data depending on whether *a priori* information on the conductivity is lower or higher than what is present in the true phantom. We observe this phenomenon again in the LK-Single level set reconstruction displayed in 4.15. When we take the interior conductivity value to be an average of the two present in the true phantom, we observe an inflated conductivity profile as it is tries to grow the shape to fit the data. We can compare this situation to the second scenario, whereby we take the interior conductivity value to be the highest interior value present in the true phantom. In comparison, we observe a smaller interior shape since the interior conductivity is almost double that of the averaged interior values.

The LK-Color level set inversion algorithm performs well in attempting to recover the

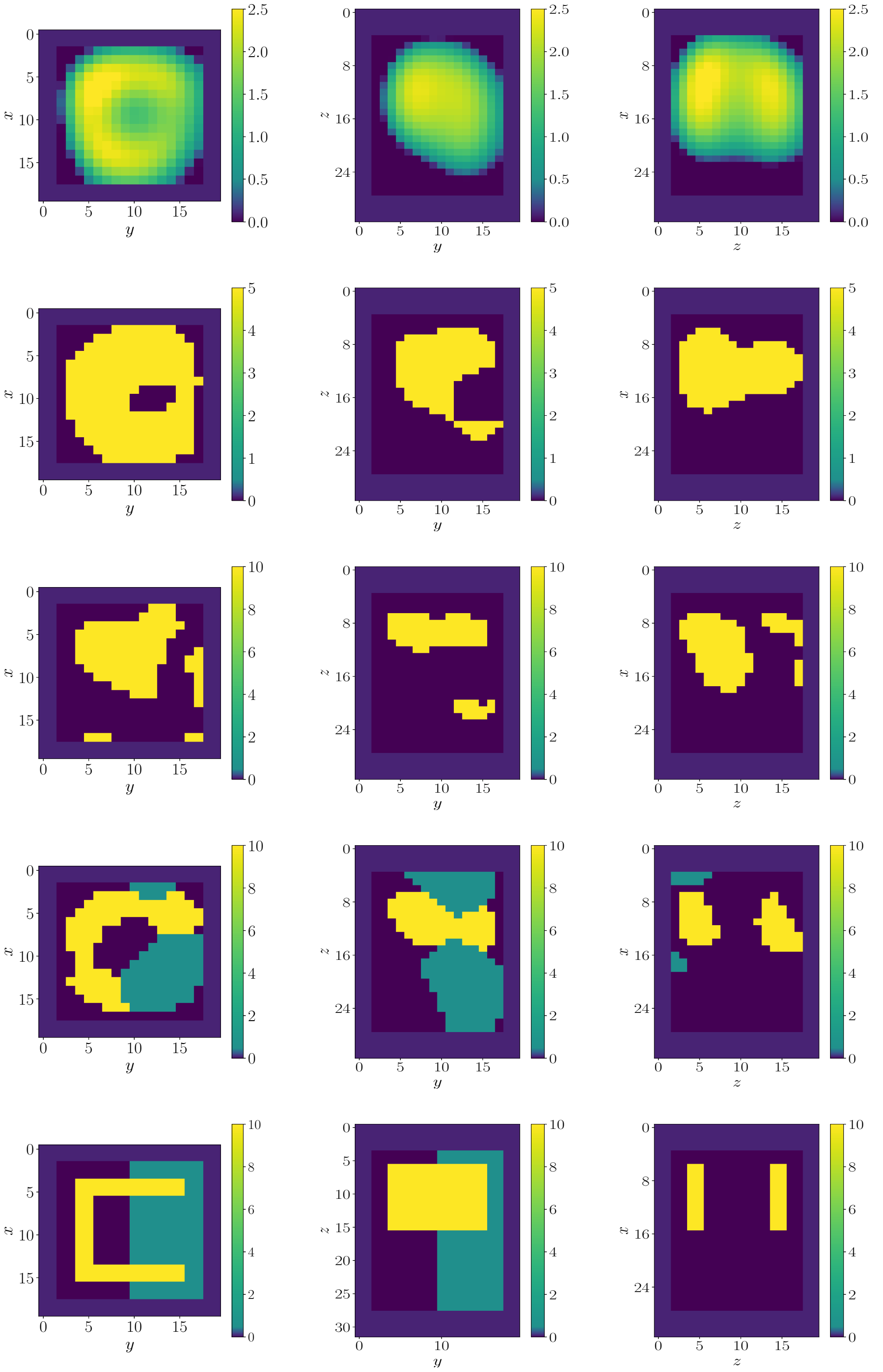


Figure 4.15: 2D cross-sections through 3D reconstructions of different regularization schemes for Numerical Experiment 1.
Each row: Left $z = 25$, middle $x = 15$, right $y = 18$.
1st row: LK-Pixel , 2nd row: LK-Single level set $b^i = \frac{1}{2}\left(b^1 + b^2\right)$, 3rd row: LK-Single level set $b^i = b^2$, 4th row: LK-Color level set and 5th row: true phantom.

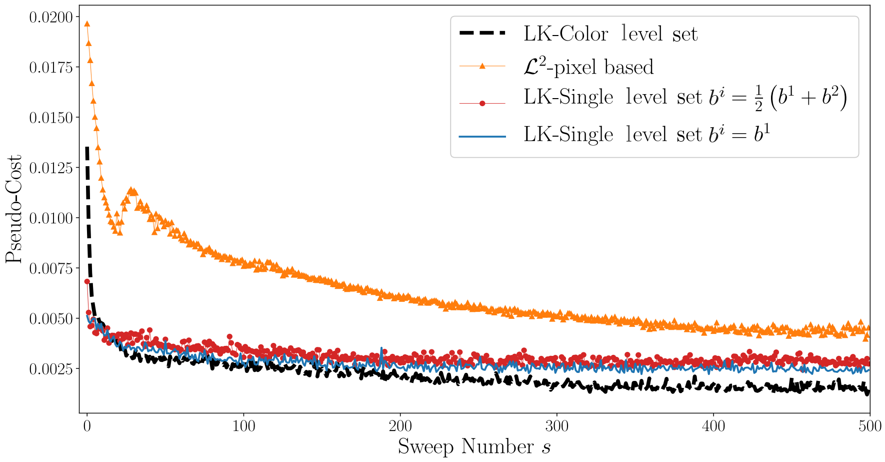


Figure 4.16: Pseudo-cost for four different regularization schemes; LK-Color level set , two variants of LK-Single level set, and a $\mathcal{L}^2$ pixel-based scheme.

true phantom, since the main characteristics of the true phantom are present in the reconstruction. For example, the scheme has managed to recover a conductivity profile that has the same distinctive separation features (i.e. a portion of b^3 is trapped in between b^1 and b^2). In comparison, the pixel-based scheme has recovered a smooth conductivity profile over the entire domain, whereas the single level set scheme has recovered an inflated shape of the true phantom. The latter is partially due to an undershooting of the true conductivity profile in the *a priori* information for single level set inversion and that it tends to struggle when trying to recover more complicated phantoms.

Figure 4.16 shows a comparison of the pseudo-cost for each reconstruction scheme shown in figure 4.15. The traditional $\mathcal{L}^2$-pixel based scheme has the highest data misfit value, whereas the data misfit of both single level set schemes are relatively close to the color level set scheme despite a considerably worse reconstruction of the true phantom. The relatively high value for the traditional $\mathcal{L}^2$-based scheme is a little bit surprising, but this could be due to slow convergence of the LK scheme. Displayed is the pseudo-cost value after $s = 500$ sweeps, therefore it is entirely possible that further reduction of this quantity could be achieved by running the scheme for significantly longer (which however would be impractical in realistic applications). Furthermore, it may also be due to the smoothing effect of the pixel-based reconstruction schemes, which makes it difficult to correctly fit data obtained with a high contrast model from reconstructions of low contrast. The two-value level set approach performs better here, but still cannot match the final cost value of the LK-Color level set reconstruction scheme in this example. This is certainly plausible, as the color level set model agrees best of all with the correct setup of the true phantom and incorporates optimally the available prior information.

4.2.3

Variability in the true conductivity profile

We can test the numerical stability of the color level set regime by increasing the realism of the true conductivity profile inside the container. In this case, we assume that the background conductivity b^1 (other than air which is b^3) has some variability. Here, we assume that the true conductivity profile admits the following decomposition:

$$b(\Phi^*)(\mathbf{x}) = \begin{cases} \xi(\mathbf{x}) & \text{in } S_1 \text{ where } \phi_1^*(\mathbf{x}) \leq 0 \\ b^2 & \text{in } S_2 \text{ where } \phi_1^*(\mathbf{x}) > 0 \text{ and } \phi_2^*(\mathbf{x}) \leq 0 \\ b^3 & \text{in } S_3 = \Omega \backslash (S_1 \cup S_2) \text{ where } \phi_1^*(\mathbf{x}) > 0 \text{ and } \phi_2^*(\mathbf{x}) > 0. \end{cases} \tag{4.3}$$

where $\xi(\mathbf{x})$ is drawn from a probability distribution and the '$*$' superscript denotes a level set function which accurately depicts the true conductivity profile. Here we will consider a comparison between two variations of a normal distribution N and a uniform distribution U. Therefore, in the two cases, $\xi(\mathbf{x}_i)$ admits the decompositions

$$\xi(\mathbf{x}_i) \sim N(\beta b^1, \alpha b^1); \tag{4.4a}$$

$$\xi(\mathbf{x}_i) \sim U(b^1 - \delta, b^1 + \delta), \tag{4.4b}$$

where $\alpha, \beta, \delta \in \mathbb{R}^+$ are chosen parameters. Note that the two normal distribution variants considered here come from choosing two different values of α.

Let us now apply the color level set inversion scheme to this new challenging situation whereby the true phantom has greater non-triviality. Numerically speaking, we now assign each cell in S^1 to a single draw from a probability distribution without replacement. We consider three examples; the first where $\xi(\mathbf{x})$ is drawn from a normal distribution such that the conductivity associated with S^1 is now slowly varying around b^1, the second where $\xi(\mathbf{x})$ is drawn from a uniform distribution whose mean is b^1 and finally where $\xi(\mathbf{x})$ is drawn from a normal distribution with mean $2b^1$.

Figures 4.17, 4.18 and 4.19 depict LK-Color level set reconstructions of a true phantom where the true S^1 is created by using draws from both normal and uniform distributions. All three figures show a LK-Color level set reconstruction of a true phantom whereby S^1 has

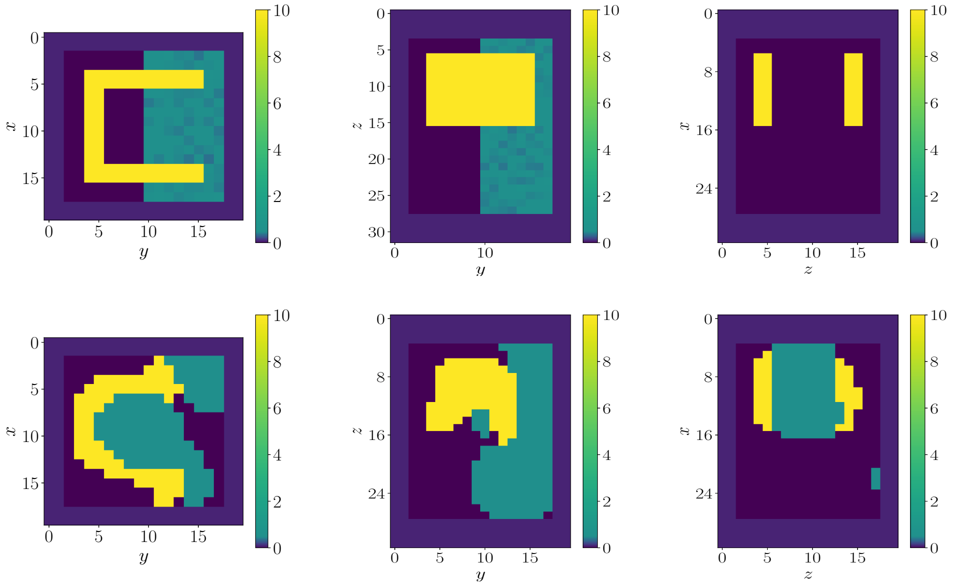


Figure 4.17: 2D cross-sections through a 3D LK-Color level set reconstruction for altered Numerical Experiment 1 (using the formulation in (4.3) for the true phantom). Here, $\xi(\mathbf{x})$ is drawn from a normal distribution with parameters $\beta = 1, \alpha = 10$.
Each row: Left $z = 23$, middle $x = 15$, right $y = 18$ Top: True Image, Bottom: LK-Color level set reconstruction at sweep $s = 1000$.

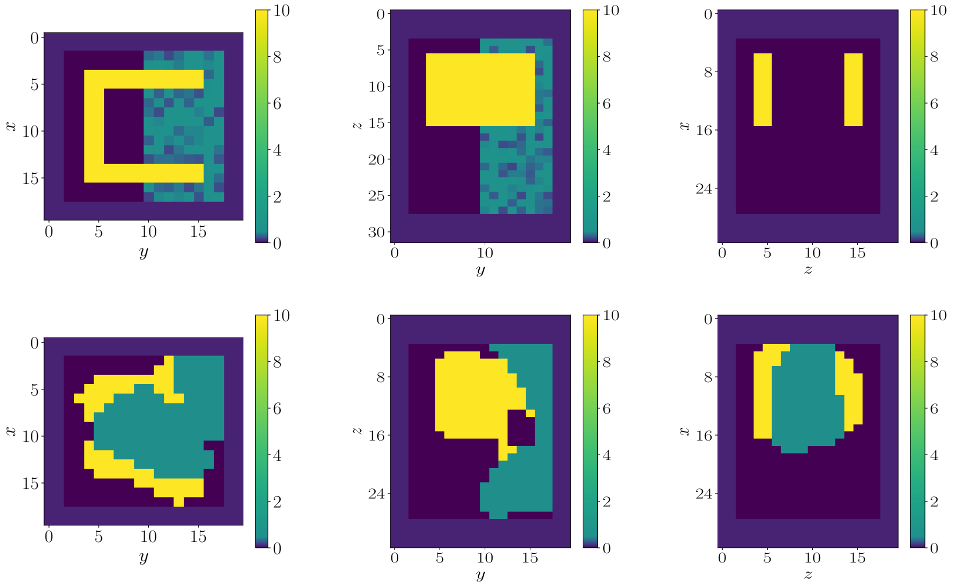


Figure 4.18: 2D cross-sections through a 3D LK-Color level set reconstruction for altered Numerical Experiment 1 (using the formulation in (4.3) for the true phantom). Here, $\xi(\mathbf{x})$ is drawn from a uniform distribution with parameter $\delta = 0.3$.
Each row: Left $z = 23$, middle $x = 15$, right $y = 18$ Top: True Image, Bottom: LK-Color level set reconstruction at sweep $s = 1000$.

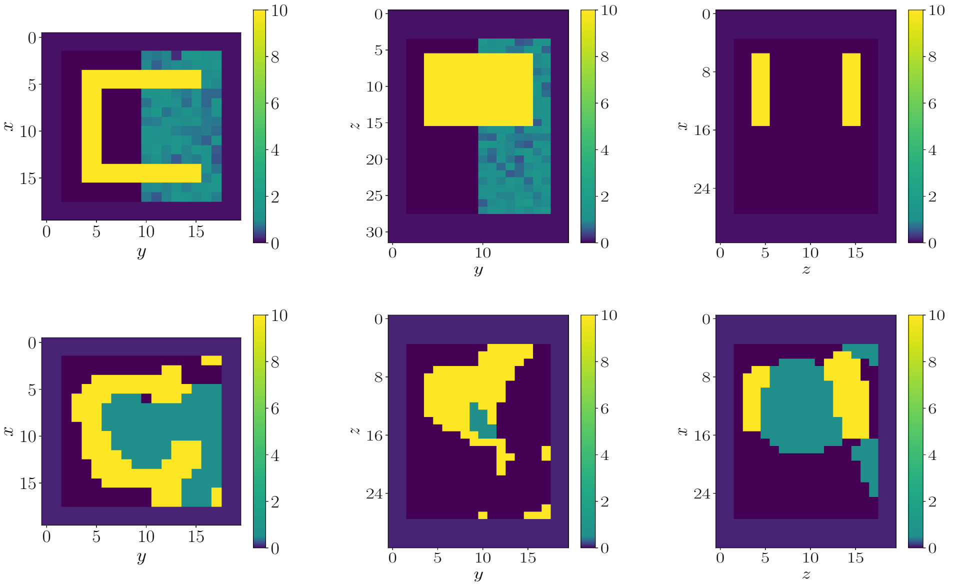


Figure 4.19: 2D cross-sections through a 3D LK-Color level set reconstruction for altered Numerical Experiment 1 (using the formulation in (4.3) for the true phantom). Here, $\xi(\mathbf{x})$ is drawn from a normal distribution with parameters $\beta = 2, \alpha = 2$.
Each row: Left $z = 23$, middle $x = 15$, right $y = 18$ Top: True Image, Bottom: LK-Color level set reconstruction at sweep $s = 1000$.

a small variability of conductivity associated with it. In comparison to the reconstructions shown in figures 4.12 and 4.15, we see that S^3 is no longer between S^1 and S^2 (resembling a hole in the true phantom). However, S^2 still has resemblance with the true phantom and most characteristics remain from the reconstruction. All three reconstructions are impressive considering many of the conductivity values associated with S^1 are not known *a priori* in the LK-Color level set reconstruction scheme. Figure 4.19, for example, shows a LK-Color level set reconstruction of a true phantom where the true conductivity profile associated with S^1 has large variability and has mean around twice the value used in the color level set inversion. Although the reconstruction scheme has been heavily penalized with wrong *a priori* information, the recovery is quite similar to the first two, which is impressive.

Figure 4.20 shows the samples which were generated for the conductivity associated with S^1 for the inversions depicted in figures 4.17, 4.18 and 4.19, from left to right respectively. In all three figures, the red line denotes the value of conductivity b^1 which is used in the color level set inversion scheme. Note that the samples shown in the left and middle figures do not seem to inhibit the LK-Color level set inversion in its recovery of the shape, since values in S^1 fall either side of b^1. One possibility of this behaviour is that the underestimation and overestimation of the conductivity associated with S^1 in the color level set inversion

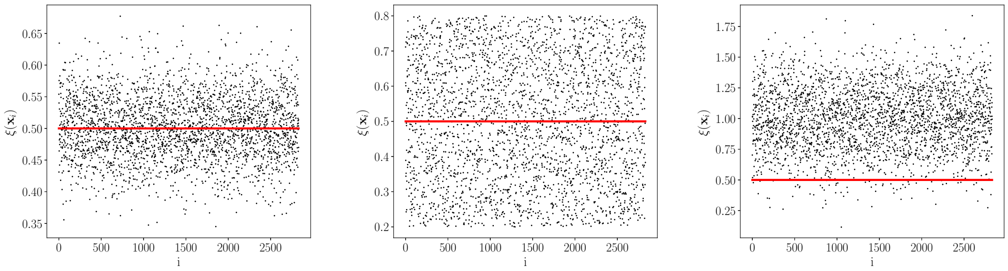


Figure 4.20: Left: Sample of conductivity profile in S^1 using probability distribution $N(b^1, 0.05)$, Middle: Sample of conductivity profile in S^1 using probability distribution $U(0.2, 0.8)$, Right: Sample of conductivity profile in S^1 using probability distribution $N(2b^1, 0.25)$.

effectively cancel each other out when observing the data (the growing and shrinking effect as discussed earlier), since both these reconstructions are similar to that shown in figure 4.12.

On the contrary, the conductivity values used for the color level set inversion in the right subplot of figure 4.20 does not capture the true values well. Impressively, the reconstruction overall captures the true behaviour of the conductivity profile almost as well as the other two better informed inversions. Figure 4.21 shows a comparison of the pseudo-cost for each sampling technique. Despite significantly different incorrect *a priori* information in the color level set formulation for each inversion, all converge to similar data misfit values.

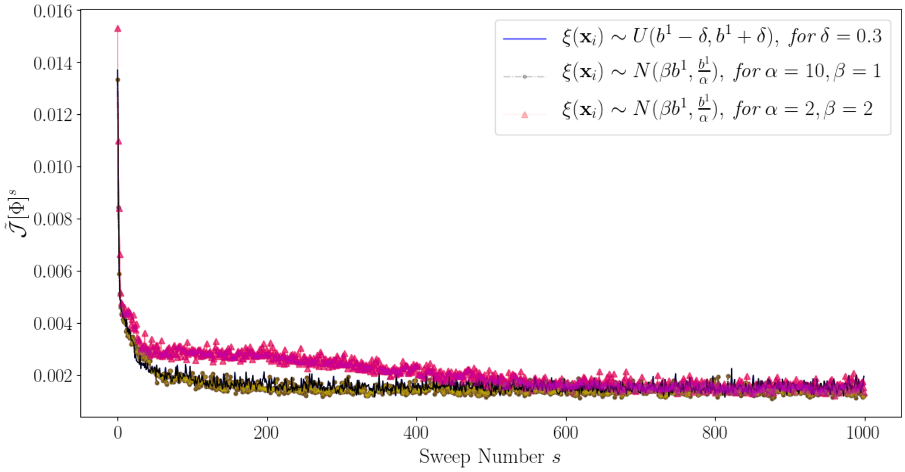


Figure 4.21: Pseudo-cost comparison between different sampling methods for creating conductivity profile associated with S^1.

4.2.4

LK-Color level set inversion for Numerical Experiment 2

The top plot in figure 4.22 shows various stages of a 3D LK-Color level set reconstruction for Numerical Experiment 2. By observation, the algorithm performs well in firstly locating objects inside the imaging domain, but incorrectly classifies the correct conductivity. This is problematic as we are recovering the incorrect image. Also notice that incorrect classification has resulted in the algorithm providing compensation for the data misfit by shrinking or growing each conductivity domain. Also shown in figure 4.22 is the line search data. Notice how some spikes in the pseudo-code match with increases in the step sizes (though these spikes are reduced in subsequent iterations).

Without showing its result, if the initial guess is flipped, in the sense that $\phi_1^{(0)}$ was changed to $\phi_2^{(0)}$ and vice-versa, we recover correct classification and location. Perhaps, this means that different initial guess' in the color level set regime provide convergence towards, or in the neighbourhood, of different local minima.

Much like single level set inversion, color level set inversion also relies on a narrowband approximation to the Dirac Delta function, meaning that the level set grows and shrinks locally. It appears that the LK descent directions send the level set functions towards a neighbourhood of local minima that resemble the nearest objects in the mapping of level set function to conductivity. From observation, it appears that once ϕ_1 and ϕ_2 have converged in the vicinity of a local minima, their zero level sets grow and shrink to try and compensate for their incorrect classifications. Though this is also possible before convergence. Unfortunately, it is difficult to say whether correct or incorrect classification is a better recovery, since in practice we do not have access to the true phantom. Moreover, it appears the compensation yields a similar data misfit value, therefore trying to use the data misfit term to see this incorrect recovery in practice is not useful. However, improving this scenario would be encouraging for future design of tailor-made algorithms where this same situation arises. We see more on this in section 4.2.5.

Therefore, in an additional run for Numerical Experiment 2, we choose a portion of the imaging domain to be randomly selected for placement of an artificial object. In this numerical experiment, we choose each artificial object to have length 6 cells in each direction, resembling a cube seed. This represents 3% of the container contents. To compare with figure 4.22, the same initial guess is used for this variant of a LK-Color level set inversion. For theoretical details on artificially placing objects inside the imaging domain see section 2.1.3.4.

The top plot in figure 4.23 shows various stages of a 3D LK-Color level set reconstruction with an initial seeding phase, where $\gamma = 15$. The reconstruction is an improvement on that in figure 4.22, in both classification and location. Although the seeding process has corrected the reconstruction, it is not immediately clear how successful the process is, since it is stochastic. Therefore, many simulations need to be performed to see its success (this will be the focus of section 4.2.5). Performing topological perturbations between sweeps in the inversion can be seen as a pseudo-initial guess and by experiment they seem to give the algorithm greater chance of recovering conductivity in the correct location. In addition to its use case in color level sets with classification, we found that the seeding technique was also useful in single level set inversion. Without going into details, we found the same behaviour as that in [127]; the seeded single level set was able to recover additional objects that were present in the true phantom which were otherwise not recovered in single level set inversion without seeding. Figure 4.24 shows a pseudo-cost comparison between two color level set inversions with and without seeding. Notice how the costs are indistinguishable even though the reconstructions are different. One has reconstructed conductivity in the correct location and the other has not. This is problematic as in practice there is no way to decide which is correct without having access to the true phantom. We discuss this in greater detail in section 4.2.5. These experiments serve as a proof-of-concept style for the seeding process as a valid tool to aid level set reconstruction schemes in recovering the correct conductivity distribution, or at least collect a set of local minima for the inverse problem.

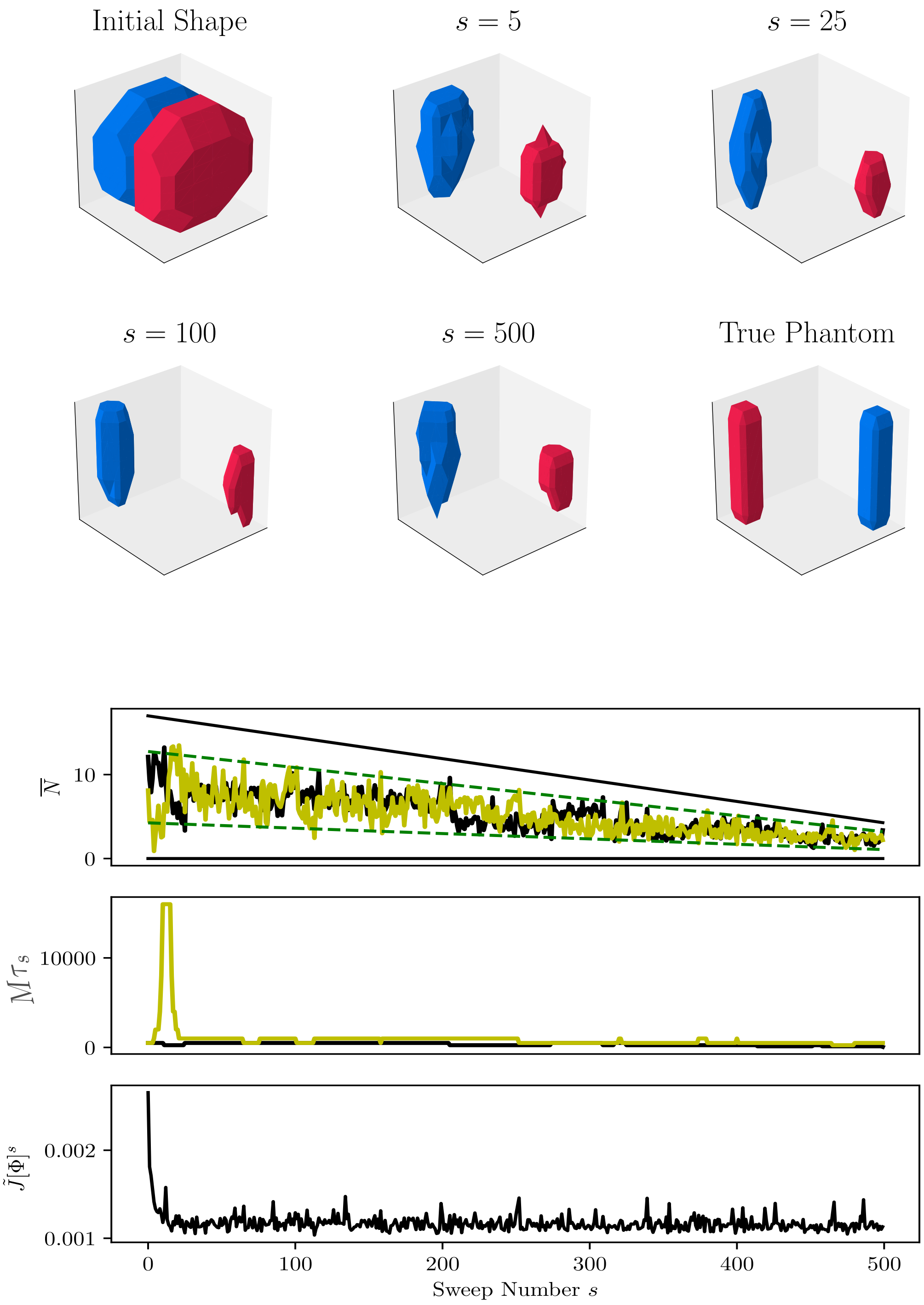


Figure 4.22: Top: Surface plots of 3D shape evolution for Numerical Experiment 2. Shown are the initial shape and snapshots at iteration numbers $s = 5, 25, 100, 500$ and the true phantom. The colour red indicates shape of the conductivity b^2 and blue indicates shape of the conductivity b^1. Bottom: The top row shows evolution of $\overline{N}$ as a function of the sweep number s. The dotted yellow line represents $\overline{N}_s^1$ and the black line represents $\overline{N}_s^2$. The middle row shows evolution of step sizes τ_s^1 and τ_s^2 each denoted by yellow and black dotted lines. The bottom row shows evolution of $\tilde{\mathcal{J}}[\Phi]^s$ as a function of the sweep number s.

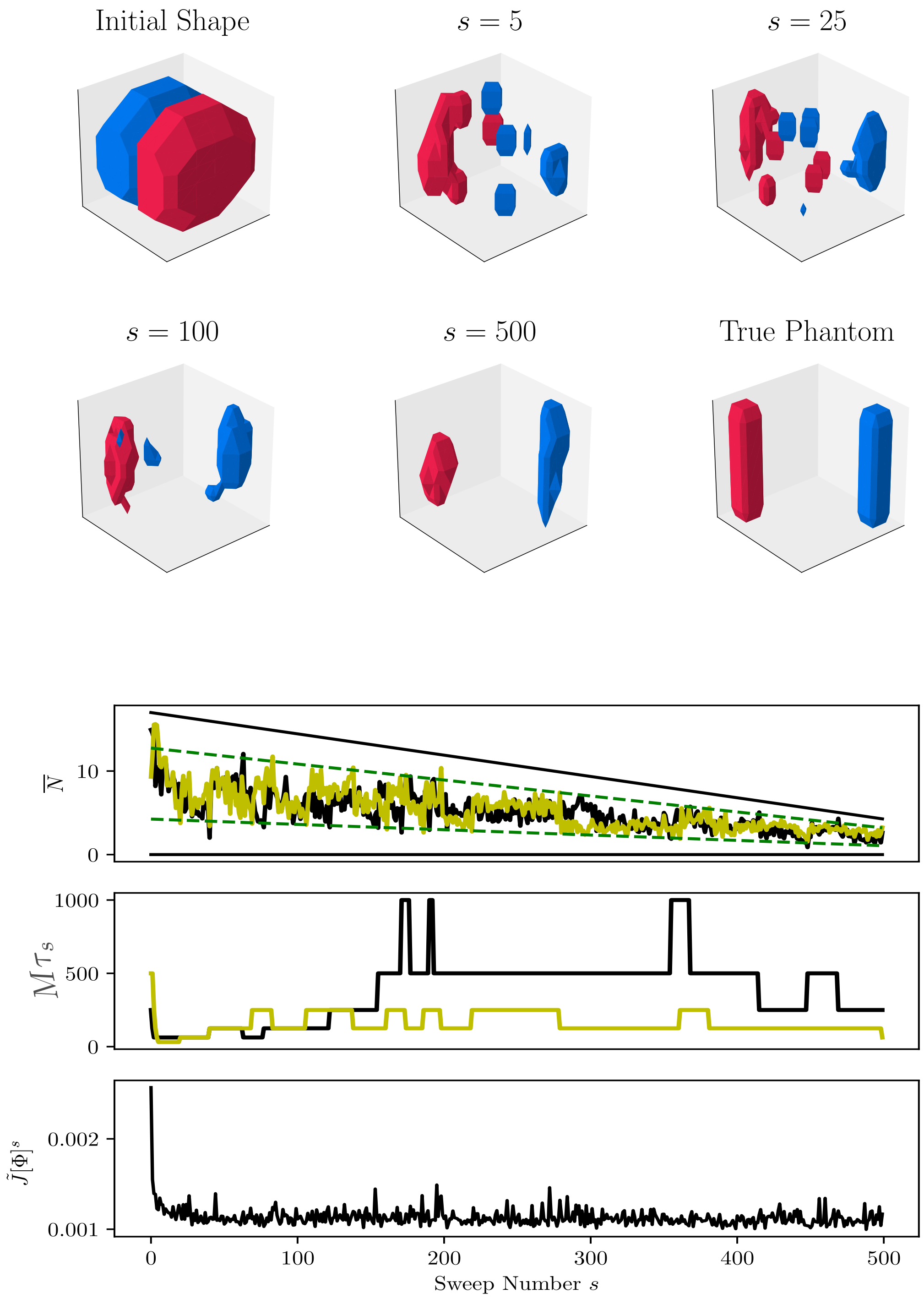


Figure 4.23: Top: Surface plots of 3D shape evolution for Numerical Experiment 2 with an initial seeding phase where $\gamma = 15$. Shown are the initial shape and snapshots at iteration numbers $s = 5, 25, 100, 500$ and the true phantom. The colour red indicates shape of the conductivity b^2 and blue indicates shape of the conductivity b^1. Bottom: The top row shows evolution of $\overline{N}$ as a function of the sweep number s. The dotted yellow line represents $\overline{N}_s^1$ and the black line represents $\overline{N}_s^2$. The middle row shows evolution of step sizes τ_s^1 and τ_s^2 each denoted by yellow and black dotted lines. The bottom row shows evolution of $\tilde{J}[\Phi]^s$.

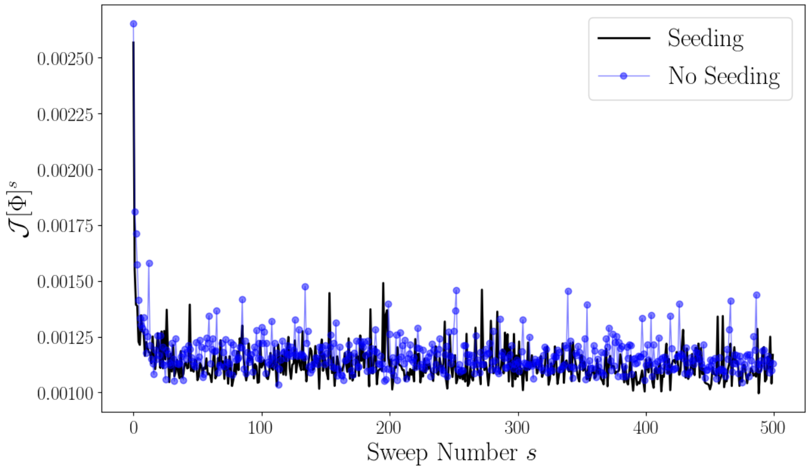


Figure 4.24: Comparison of pseudo-cost $\mathcal{J}[\Phi]^s$ for LK-Color level set inversion with seeding and LK-Color level set inversion with no seeding.

4.2.5
Seeding study

The results in the previous section suggest that it is possible to probe the imaging domain with artificial objects in order to escape unwanted local minima. Since we are unaware of what is contained in the imaging domain, locations for placing artificial objects must be chosen according to some random rule.

In our setup, minimizing (2.71) is a stochastic process, as Ω_2 is typically chosen at random. Though if other information becomes available, suggesting that some locations are more favourable, then those locations should be used. However, in general, the seeding process is stochastic. The result in figure 4.23 is encouraging, although we cannot expect to recover this result for each instance of algorithm 6, given that $\gamma > 0$. To quantify how successful the stochastic process is, we consider N samples of the color level inversion for the true phantom in Numerical Experiment 2 with the same initial guess and $\gamma = 15$. As mentioned, it is not always guaranteed to find the correct classification as in figure 4.23. In fact, we may end up with distinct reconstructions which resemble local minima of the cost functional. We hope to capture this set of distinct local minima (i.e. those that give different classifications, locations, etc) such that we can infer from this information. Here we provide a proof-of-concept study, where the true phantom is known and can be compared with the reconstructions. We firstly consider a stochastic sampling of the color level set inversion whose initial guess was chosen such that it would incorrectly classify conductivity.

As observed in figure 4.24, the pseudo-costs associated with reconstructions with and without seeding (i.e. correct vs incorrect classification) are indistinguishable. It seems using the pseudo-cost to determine whether a reconstruction has correct conductivity in the correct location is infeasible. We test this by considering multiple color level set inversions. We introduce some measurements which help in the analysis. These are:

$$\overline{\tilde{\mathcal{J}}[\Phi]} = \frac{1}{S}\sum_{s=1}^{S}\tilde{J}[\Phi]^s;$$

$$V_s = g(b_s - b^*), \ \ \overline{V} = \frac{1}{S}\sum_{s=1}^{S} V_s,$$

where $\overline{\tilde{\mathcal{J}}[\Phi]}$ is the average pseudo-cost of an inversion after S sweeps, b_s is the conductivity profile at sweep s, b^* is the true conductivity and $g(\cdot)$ is a function which counts the number of non zero entries in its input. Hence, V_s is the total number of incorrect voxels at sweep s and $\overline{V}$ is the average incorrect voxel count of an inversion after S sweeps. We expect the average cost and the average incorrect voxel count to be positively correlated for obvious reasons. Ideally, we can separate inversions into clusters resembling local minima and compare them with another scenario which seems to arrive at a similar reconstruction each time - regardless of stochastic seeding. Of those classified into correct and incorrect reconstructions, the ideal scenario is that the average average pseudo-cost is greater in the incorrect group than correct group. If this proves to be incorrect then we can conclusively say that the problem cannot distinguish between these minimizers, by experiment. On the other hand, if it is proved to be correct, then we can conclude that there is scope for further analysis into why the pseudo-cost is larger, possibly by collecting a bigger sample for greater confidence in the result.

Let us begin with Numerical Experiment 2. We decide to compute $N = 125$ color level set inversions with an initial seeding phase ($\gamma = 15$). Figure 4.25 shows some statistics of the color level set inversions. The left subplot shows a frequency chart of the inversions and their average incorrect voxel count. We see that the majority of inversions have above $\sim$ 100 incorrect voxels on average. This is just as pronounced in the right subplot of figure 4.25, where we see a cluster of inversions to the right. Unfortunately, whilst the average pseudo-cost and average incorrect voxel count are correlated, the spread of the cluster is large. In fact, many of the inversions to the right (which we suspect to be incorrect reconstructions) have an indistinguishable cost value when compared to those of less than $\sim$ 100 incorrect voxels.

Employing an expert to classify the reconstructions as correct, ambiguous and incorrect results in figure 4.26. Those that were defined as mostly correct resembled the true phantom

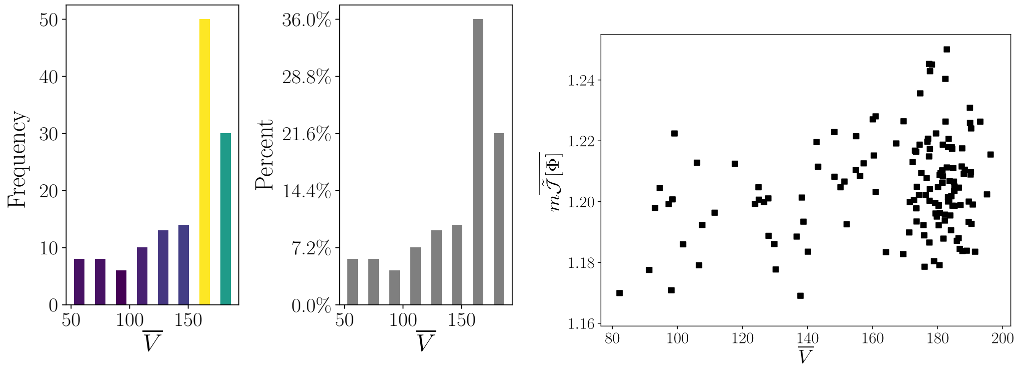


Figure 4.25: Left : Histogram plot of average incorrect voxels $\overline{V}$.
Right : Relationship between pseudo-cost and $\overline{V}$. Each point represents one inversion.

with possible minor defect contributions coming from the other internal conductivity, those defined as mostly incorrect had conductivity domains that were mostly in the wrong place and those defined as ambiguous had conductivity domains that were mixed together in both locations. The classification was performed by looking at one slice in the *xy* plane (generally sufficient). Figure 4.27 shows examples of each classification group.

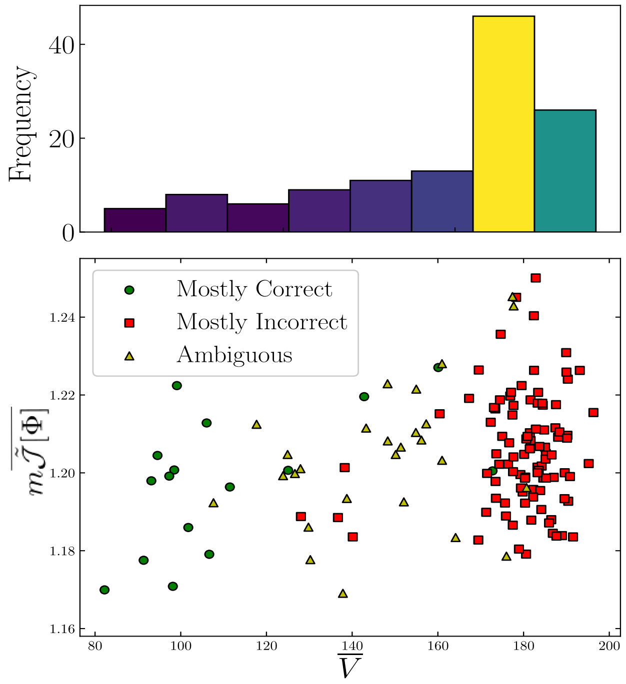


Figure 4.26: Classifying LK-Single level set inversions with bias initial guesses.

As expected, there is a strong correlation between classification and incorrect voxels.

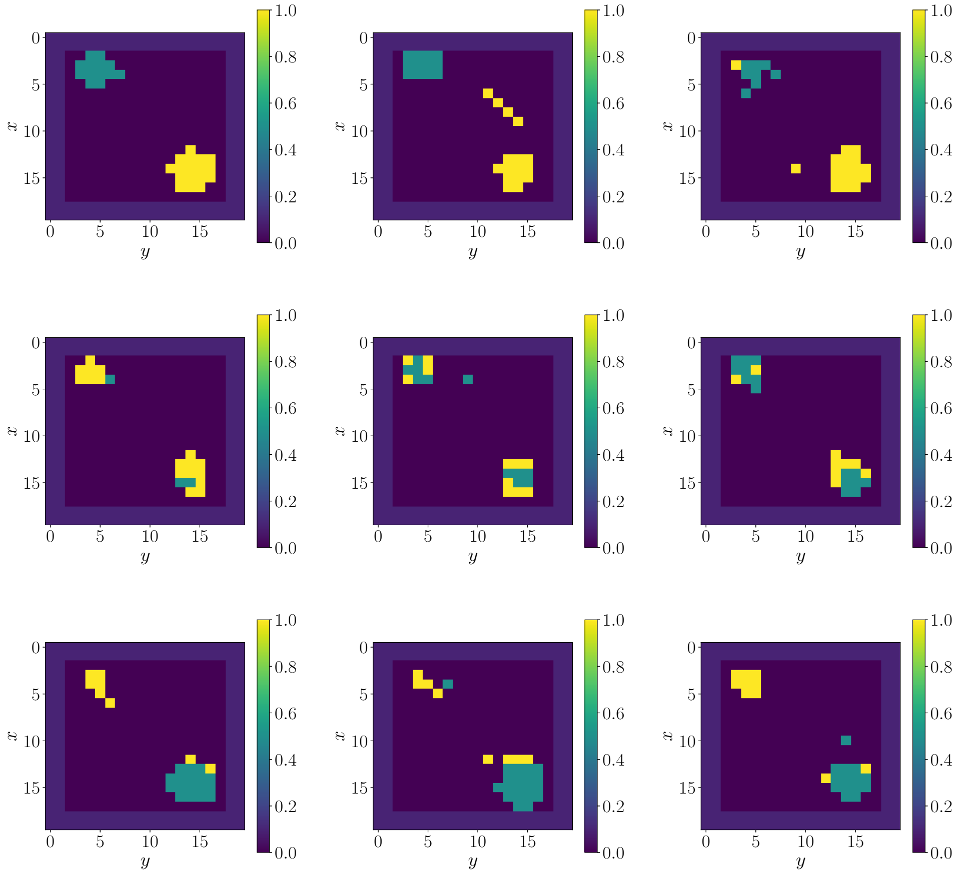


Figure 4.27: Some examples in each classification group.
1st row: Classified as mostly correct, 2nd row: Classified as Ambiguous, 3rd row: Classified as mostly incorrect.

What is more interesting, however, is the ambiguity that some inversions provide. In this problem, it seems that the majority of inversions converge to two distinct local minima (see figures 4.22 and 4.23). However, some fail to either grow or shrink the artificial objects (which may be placed on top of each other). Subsequently, they join together forming ambiguous results, with respect to the true phantom.

Figure 4.28 applies the same classification as that in figure 4.26, but this time with a non-bias initial guess. In this case, rather than using the initial guess from Numerical Experiment 2, only the seeding phase is used as an initial guess. In comparison with the bias initial guess, the non-bias case favours neither local minimum; hinting that the eventual reconstruction is dependent on the initial guess in this scenario.

Figure 4.29 shows a joint plot of inversions for both bias and non bias initial guesses. Note that those with the bias initial guess have on average a higher pseudo-cost than those which are random guesses.

Figure 4.30 shows a scatter plot of inversions for Numerical Experiment 1. The inversions use random initial guesses generated from an initial seeding phase. A closer observation of the reconstructions themselves show they appear similar to those in figures 4.11 and 4.12. This is also reflected by the classification in figure 4.30, where the green colour denotes mostly correct. In Numerical Experiment 1, the true phantom fills the majority of the imaging domain and is intertwined. Because of this scenario, it could be possible that no other distinct minimizers in the color level set regime exist for this scenario. For example, the algorithm may not be able to grow or shrink conductivity domains to arrive at a similar data misfit. Therefore, the data in this case may point to similar local minima in the color level set regime.

Since level set approaches provide local updates in the corresponding minimization process, it appears that when conductivity domains are not embedded with each other, then possible ambiguities can occur and more than one distinct solution is common. This observation is based on the statistical data for both numerical experiments considered in this section. The inversion related to the conductivity domains embedded within each other offer a similar form of solution when applying seeding from the onset. However the true phantom in Numerical Experiment 2 offers several solutions which fit the data in the color level set regime. We cannot be sure that topology of the true phantom is its causation, however it is a reasonable possibility.

Figures 4.28 and 4.29 show no significant distinction of the pseudo-cost for correctly and incorrectly classified reconstructions. To say with greater certainty, more samples need to be collected. However, due to the computational expense of such inversions, this is delegated to future research. The idea of classifying reconstructions motivates the classification task in appendix A, which focuses on classification of reconstructions using an autonomous approach. The classification employed in this section is an ad-hoc technique to try and understand why some minimization problems behave differently to others.

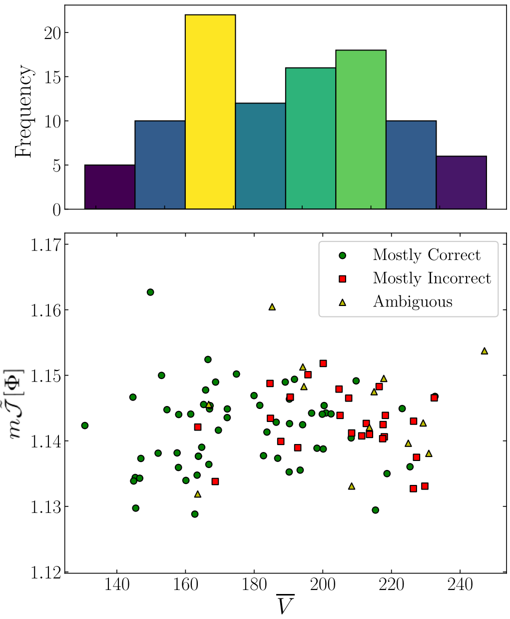


Figure 4.28: 2D plot of average residual $\overline{m\tilde{\mathcal{J}}[\Phi]}$ against average incorrect voxels $\overline{V}$.

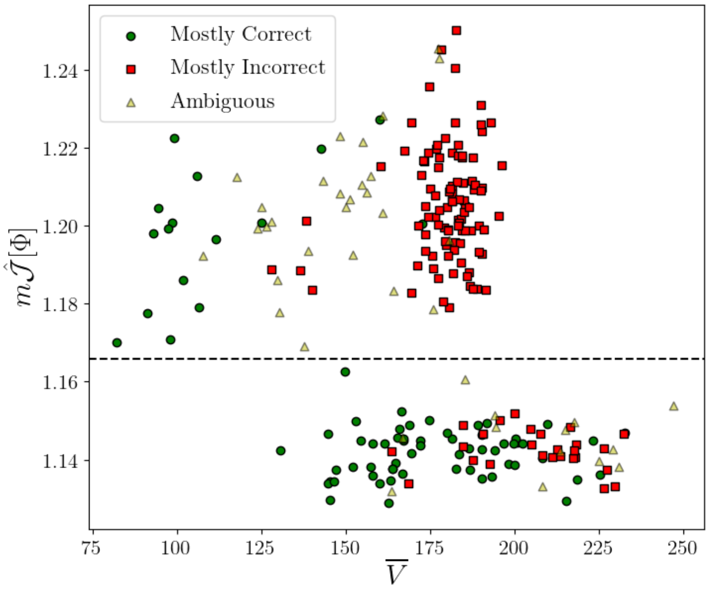


Figure 4.29: Joint 2D plot of average residual $\overline{m\tilde{\mathcal{J}}[\Phi]}$ against average incorrect voxels $\overline{V}$ for bias and non bias initial guesses. Above dotted black line are bias inversions and below are non bias.

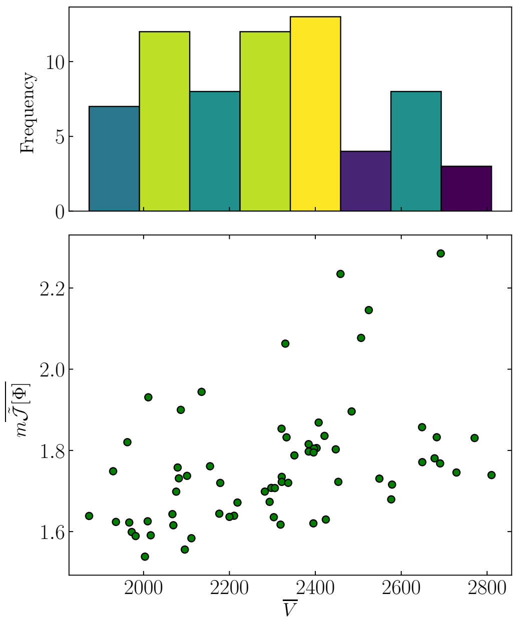


Figure 4.30: Relationship between pseudo-cost and average incorrect voxel count for Numerical Experiment 1. Note that all these inversions have converged to similar conductivity shapes and are not distinctly different as in Numerical Experiment 2.

CHAPTER 5

Conclusion and Future Work

The main goal of this thesis was to design novel reconstruction schemes for screening objects enclosed inside electromagnetic shields using electromagnetic data measured externally. In these situations, high frequency electromagnetic waves cannot penetrate the shields due to the skin depth. However, low frequency electromagnetic waves can, meaning information about the contents can elude the shielding and appear in electromagnetic data measured outside these enclosures.

Novel reconstruction schemes, as outlined in chapter 2, were designed to recover information about the conductivity profile inside the imaging domain (contents of electromagnetic enclosures) from electromagnetic data. These schemes were either pixel-based or shape-based. The former meaning no assumptions were made on the conductivity profile, whereas the latter assumed the conductivity profile was known but its shape not. Following a review of existing methods to solve Maxwell's equations in chapter 3, we provided numerical results for all new algorithms and compared them against an existing pixel-based scheme. Whilst low frequency electromagnetic waves can penetrate electromagnetic enclosures, resolution in the reconstruction images suffer. The pixel-based schemes, whilst providing some information about location of objects and some characteristics, consistently underestimated the conductivity value and the profile was typically smoothly splayed out in the imaging domain. On the other hand, the shape-based schemes generated impressive reconstructions for the situation, particularly when the true contents were chosen to be non-trivial. However, some ambiguities arose in the shape-based regime. For example, in the single level set case, where conductivity profiles are composed of two known values, ambiguities arise when one of the two values are slightly incorrect. The experiments in chapter 4 confirm that in these scenarios, the algorithm can still correct identify locations of objects, but the shape may be 'deflated' or 'inflated' based on whether the incorrect value is above or below the true value. It seems this behaviour also filters into the color level set regime. The general location of objects tend to be recovered with an appropriate initial guess, though the conductivity at some or all locations can be incorrect and are clearly dependent on the initial guess (as discovered in chapter 4). To improve this issue, a method which places artificial objects inside the imaging domain was introduced. All these results are shown in chapter 4, some are published in [58] and others are present in the submitted article listed in section 1.3.

The first new algorithm solves a shape-based electromagnetic inverse problem by adapting general ideas described in [35, 39] for application to this new challenging situation. The second one solves a pixel-based electromagnetic inverse problem and promotes sparsity in the solution through regularization. The method used here follows that proposed in [100] (for a different 2D imaging modality) and extends it to 3D imaging. When *a priori* knowledge is available, indicating that inclusions with small compact support and significantly higher internal conductivity value are present in the domain of interest, both sparsity regularized and level set based approaches perform better than the standard $\mathcal{L}^2$-based scheme. This is true even in situations where the $\mathcal{L}^2$-based scheme reduces the least squares data misfit functional to the same extent as the other two schemes. Due to the additional constraints included in the sparsity and level set based algorithms, reconstructions are more likely to yield results which are in agreement with such prior information.

The third new algorithm is the LK-Color level set reconstruction scheme. Shown in chapter 4 is a comparison with alternative regularization schemes to see how well it fared against other new schemes introduced here and existing methods. The scenario considered in Numerical Experiment 1, as discussed in section 4.2, is more realistic and challenging than those previously considered for this application (as in section 4.1). Increasing realism of the problem by choosing a conductivity profile with more than two constant domains as the true phantom, has led to improved results but also to additional challenges. In the LK-Color level set regime, we found that the increased complexity of the model comes with some new types of local minima. Some of those local minima resemble incorrect classification of an object's conductivity when compared with the true phantom. Here, the seeding process can try to circumvent some of those issues, or at least give a set of local minima to infer from by other means. On the other hand, the increased complexity of the color level set method, compared to the single level set method, can provide improved results in situations where the correct content of a shielded box contains multiple conductivity values, as shown in Numerical Experiment 1. In fact, in such situations color level set methods seems to outperform both the single level set approach and the traditional pixel-based approach. However, in situations where two or more isolated objects have different conductivity values and are located inside the shielded box or container, the color level set method does not provide improved reconstructions when compared with the other methods. Basic characteristics of the overall content can be inferred from all three of them, but what the color level set inversion converges to depends greatly on the initial guess and the overall shape evolution. This may result in the final reconstruction being misleading. This behaviour is highlighted in section 4.2.4 when the seeding process was employed to avoid local minima. However, even such a scheme is not able to arrive at the correct final characteristics of the individual objects with significantly increased certainty.

Appendix A presents a methodology for *classifying* reconstructions generated from some of the schemes presented in chapter 2. The scheme presented there is entirely disjoint to the reconstruction schemes and can be seen as a post processing technique. The scheme is trained using the EMNIST dataset, which is a large collection of images of handwritten digits and letters [24]. The scheme is then applied to images which result from trying to reconstruct 3D phantoms whose cross-sections resemble digits or letters. The results in appendix A show that the scheme can correctly identify certain shapes inside the imaging domain and flag them up as specific letters. This methodology can no doubt be extended to more realistic scenarios where the task, for example, could be to identify sharp objects and/or dangerous equipment. Since the technique is disjoint to the reconstruction scheme, the identification will become better as existing reconstruction schemes improve with time. The bottleneck for correct identification of objects is the reconstruction scheme rather than the trained model for identification, as the latter is dependent on the amount of data available in the training set whereas the former is restricted to physics of the problem. As touched upon at the beginning of appendix A, ideally we will move towards classifying electromagnetic data rather than images which result from specific reconstruction algorithms.

There are many possible avenues of research which can follow from the work presented in this thesis. The inverse problem described in chapter 2 is set up for a single parameter inversion involving recovery of the conductivity profile inside the imaging domain from electromagnetic data. For a more realistic approach it would be beneficial to develop a multi inversion scheme which incorporates recovery of both magnetic permeability and electrical permittivity alongside conductivity, since in practice these characteristics may have an impact on the electromagnetic data. Adjustments to the forward problem may need to be made to incorporate inhomogeneous electrical permittivity, so a starting point may be a joint inversion of conductivity and magnetic permeability as the finite volume formulation accounts for inhomogeneity in magnetic permeability. This is currently not a well explored topic and it would be interesting to see results for a multi inversion in this application. Moreover, regardless of whether the inversion is a single or multi parameter inversion, it would be beneficial to research other regularization schemes which could be appropriate for this application.

Another possibility is to consider real data rather than synthetic data. This may be difficult, as the equipment and setup of the physical experiments has to have agreement with the forward solver. Nevertheless, pursuing real data inversion is the gold standard in inverse problems. The machine learning approach presented in appendix A has many future directions. One is to create 3D datasets for the training set in the classification problem so that 3D objects can be identified. Another is to use ML techniques to learn the forward/adjoint operator to reduce computational expensive of solving Maxwell's equations. Amongst the lower hanging fruits, synthetic noise could be generated from alternative forward solvers, different source and receiver set ups could be chosen such that sensitivity functions inside the imaging domain are optimized and synthetic data could be generated at different grid sizes to resemble more realistic electromagnetic data. The future research directions outlined above are by no means an exhausted list. The sheer amount of possible avenues are a clear indication that the field is in its infancy as there are many things that can be done.

APPENDIX A

Object Classification

In this appendix, we describe a classification task for electromagnetic images which result from some of the algorithms discussed in the main body of the thesis. This application hopefully provides incentive for a classification task which involves mapping electromagnetic data to shapes in the classification, as such a scheme would be significant in the context of screening applications.

Since the 1950s, a subset of Artificial Intelligence (AI), called Machine Learning (ML), has revolutionized many fields. This is particularly so in more recent years, as many ML problems are computationally expensive by design. But with the advent of improving computer systems, these bottlenecks have reduced and an explosion of innovations have happened in image classification, object detection, speech recognition, medicine and biology, cancer detection, self-driving cars, amongst many other fields, see [9, 71, 86, 111] for more detail on some of those. Another motivation of the increasing innovation in ML are pop culture surrounding famous victories in iconic ancient games. For example, in 1997 the reigning chess world champion, Garry Kasparov, was defeated by Deep Blue, an AI created by researchers at IBM [19]. More recently, in 2016, an AI named AlphaGO, created by a team at DeepMind, defeated the world champion Lee Sedol of the ancient chinese game GO. Famously, one of the moves made by the AI, turn 37, was unexpected and by commentators was seen to be novel, creative and innovative when observing the state of play retrospectively [113].

In fact, this turn led many human players to change their strategies following this match. Since then, DeepMind have gone one step further in 2018 and designed a new AI called AlphaGO Zero to compete against AlphaGO. The subtle difference between these AIs was that AlphaGO learnt how to play GO through human players whereas AlphaGO Zero was self-taught and based on AI play. AlphaGO Zero beat AlphaGO 100-0 in 100 games [114]. Many of these famous victories in iconic games, which require strategy and creativity rather than brute force, have inspired the next generation of researchers to innovate further.

Here, we focus on one of these exciting applications. In particular, we use object detection algorithms to determine *certain shapes* which appear in images resulting from the novel reconstruction schemes. These *certain shapes* can be separated into so-called classes, with the goal being to train an AI to observe images and place them into their correct classes. This is known as a classification task, and many algorithms have been designed to address this problem. Some of the most popular methods include Support Vector Machines (SVMs) [63], Neural Networks [73], Decision Trees [70], k-Nearest Neighbor [27], amongst others. We use one of these methods for the classification task.

Classifying images which result from solutions to inverse problems using ML is a relatively recent approach, but it has been studied extensively in other imaging modalities, such as medical imaging [86, 108, 111]. In that case, for example, one is interested in identifying possible tumours, fractures, damaged bones, amongst other conditions from the reconstructed image. In the past and still prevalent today, experts observe data produced by an imaging modality (whether that be x-ray imaging, ultrasound imaging, ...) and diagnose patients based on what they observe. More recently, with the advancement of ML algorithms, this process is becoming more automated. There has been alternative developments of ML algorithms in inverse problems too. For example, there has been research into developing data-driven forward operators [115], solving ill-posed inverse problems with neural networks [2] and learning regularization terms [5]. We will not apply any of these in this thesis, but it is useful to highlight what could be applied in future research. Here, we consider classifying shapes which appear in electromagnetic images resulting from the novel reconstruction schemes shown in chapter 2.

In chapter 4, we 'classified' inversions using an ad-hoc method. Here, we would like to demonstrate an automation of the expert observation. The utility of such a method is quite clear for security screening applications. For example, if a certain shape was flagged during an inversion, then the container of interest could be inspected. Here, we develop a classification scheme which is *real time*, in the sense that, we apply the classification after each sweep of the inversion process. In this way, we quantify the behaviour of each inversion technique

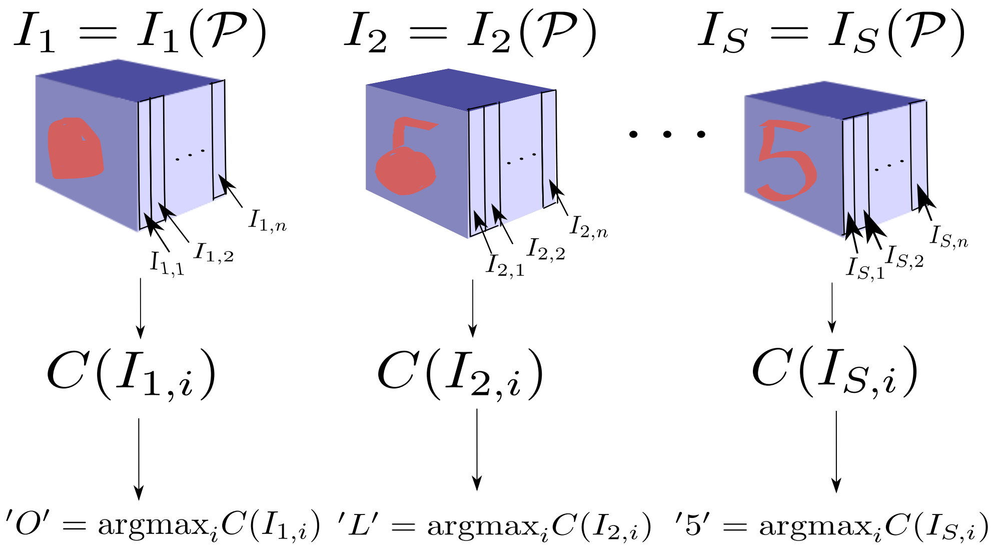


Figure A.1: A schematic of pseudo-3D classification approach. From left to right shows 3D arrays I_k which are segmented by one axis such that the classification operator C acts on a slice-by-slice basis. Note that I_k is a function of some method P which generates the array at sweep k.
Once the classification of each slice has been computed, we then take the mode classifier as the resulting classification of the 3D object. In this example, a number 5 has been found to be the mode classifier.

as they progress. Note that the classification scheme introduced here is not restricted to the inversion routines in this thesis. Success of the classification is clearly dependent on the inversion scheme as they dictate the input space of the classification function. For example, when developments of reconstruction schemes for low frequency electromagnetic inverse problems occur, they can be used here. The same can be said for the classification scheme. Currently, rich datasets of 3D objects are not readily available, therefore we train a classification scheme with 2D images then apply a pseudo-3D classification scheme which classifies images on a slice-by-slice basis. Figure A.1 shows a general outline of how we intend to classify 3D reconstructions of conductivity.

A.1

Formulating the classification task

This thesis has been concerned with recovering conductivity from electromagnetic data measured at receiver locations (see section 1.2 for the imaging problem and chapter 4 for numerical results). Therefore, in this framework, we formulate a corresponding classification task which identifies certain shapes in conductivity images which result from the reconstruction schemes introduced in chapter 2. In general, the reconstruction task can be formulated as follows:

$$\mathcal{P}(D) = [\sigma_{\mathcal{P},1}, \sigma_{\mathcal{P},2}, \ldots, \sigma_{\mathcal{P},S}]^T \tag{A.1}$$

where D is data measured at the receiver locations, $\mathcal{P}$ is the reconstruction scheme, whether that be a level set, pixel-based or an unknown inversion method, and $\sigma_{\mathcal{P},i}$ is the corresponding conductivity profile at the ith sweep which the reconstruction scheme $\mathcal{P}$ generates. We design a classification algorithm which applies itself to each entry of the vector in (A.1). The classification problem can be stated as follows:

Problem 1: Electrical Conductivity Classification

Given the vector $\mathcal{P}(D)$, for some method $\mathcal{P}$ and data D, apply a trained classification function C to each entry of $\mathcal{P}(D)$ such that $C(\sigma_{\mathcal{P},i}) = c_i$ for $i = 1, 2, \ldots, S$ for some class label c_i.

Note that this problem has two interpretations; we can either apply the classifier as entries of $\mathcal{P}(D)$ are available (i.e. *real time* reconstruction) or the classifier can be a post-processing procedure. We solve Problem 1 for multiple methods $\mathcal{P}$.

To solve Problem 1, we must be satisfied with two components. The first is the reconstruction task generating the vector in (A.1) and the second is building/training an algorithm competent at classifying shapes. Generating the vector in (A.1) for various $\mathcal{P}$ has been the main topic of the work here. For more details on the reconstruction methods see chapters 2 and 4. Here we assume that we already have access to $\mathcal{P}(D)$ and focus completely on creating C.

Training the classification function C in our case involves minimizing the distance between $C(x_i)$ and c_i, where x_i are inputs and c_i are the correct classification of those inputs. Having classification labels accompany inputs in the training set means that the training process is *supervised*. *Supervised learning* forms one of the most common types of ML algorithms, as well as *unsupervised* and *reinforcement learning*. We will only consider one *supervised learning*

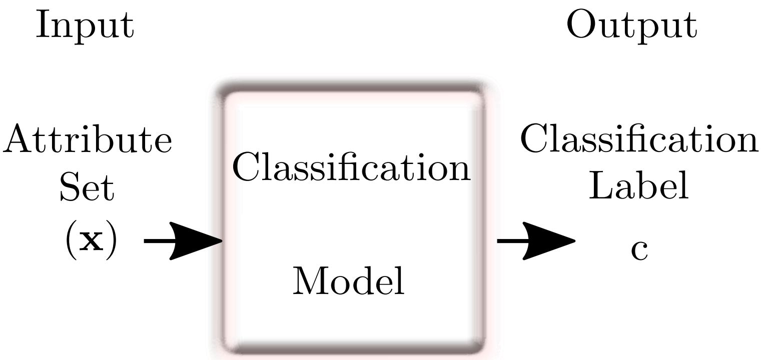


Figure A.2: Schematic of classification function C.

algorithm here. For more information on the main differences between these methods see [49]. Comparisons with other learning algorithms, as well as applying both reinforcement and unsupervised learning methods are delegated to future research.

The framework of training C, as described here, follows that in [57]. To begin *training* a classification function, we need a training set X whose elements contain inputs x_i and their corresponding classification labels c_i [1]. The set X is assumed to contain n elements, each consisting of a 2-tuple; one input and one corresponding classification label. Mathematically speaking,

$$X = \left\{(x_1,c_1),(x_2,c_2),\dots,(x_n,c_n)\right\}.$$

A reasonable condition on C is that it should be able to project inputs $x_i \in X$ towards their corresponding classification labels $c_i \in X$, i.e. $C(x_i) \approx c_i$. Figure A.2 shows a qualitative view of the classification process.

Within this framework, we seek to minimize the following cost:

$$L = \sum_{j=1}^{n} ||C(x_i) - c_i||^2_{\mathcal{L}^2(\Omega)}. \tag{A.2}$$

Although minimizing the cost function in (A.2) seems reasonable, since we want our function C to create a valid input-classification map of the set X, it does not generalize well. The minimizer of (A.2) is quite clear, setting $C(x_i) = c_i$ for each i yields the minimum of L. However, once C observes other inputs not contained in the training set X, it will not perform well as it's over fitting to the training data. The idea of *training* a function C is so that if C evaluates an input it hasn't seen before, say $\hat{x}$, which has the same classification label as an input x,

---

[1] For example, the inputs could be images of dogs or cats and the classification labels could be $0 =$ dog and $1 =$ cat.

then C in most cases should output the same classification label. In some cases, inputs may be slightly ambiguous and be subsequently misclassified, but in general they should output the same result. For example, two images of the same digit with different handwriting should classify as the same digit, given that the classification function has been trained to recognize digits.

To fix the generalization problem, we restrict the choice of C to a family of functions $\mathcal{F}$ which contains functions that are parameterized by weights. In particular, we choose a nonlinear function parameterized by weights $\mathbf{w}$ such that its range is large with respect to tuning of the weight parameters. We desire the range of the function to be large and the input-output map to be nonlinear such that for some $\mathbf{w}^*$, the function captures the correct behaviour in our training set. Making the function have a large range gives the optimization process a better chance of finding a suitable solution. The new goal is to minimize:

$$L(\mathbf{w}) = \sum_{j=1}^{n} ||C(x_i;\mathbf{w}) - c_i||^2_{\mathcal{L}^2(\Omega)}. \tag{A.3}$$

with respect to the weights $\mathbf{w}$.

In general, once the classification function has been adequately trained by minimizing (A.3) using the training set X, accuracy of the function is measured by using inputs from a test set $\hat{X}$ where

$$\hat{X} = \left\{(x_{n+1}, c_{n+1}), (x_{n+2}, c_{n+2}), \dots, (x_{n+m}, c_{n+m})\right\},$$

whose input-output pairs were not used in the training procedure. One must be cautious in minimizing (A.3), as we do not want to overfit the data present in X. Our hope is that minimizers of (A.3) work well for other datasets such as $\hat{X}$. Minimizing (A.3) with respect to the data set X as much as possible will result in similar but less severe problems as minimizing (A.2).

The conductivity profile $\sigma_{\mathcal{P},i}$ is a 3D array, therefore if we were to solve Problem 1 as stated, we would have to train C on 3D images. Whilst this is achievable, 3D datasets are not as readily available. However, we can employ a 2D classification function on planar images in $\sigma_{\mathcal{P},i}$ and take the mode classification of these planes as the classifier, invoking a pseudo-3D technique (see figure A.1).

As an example case, we use a dataset called 'EMNIST' to train our classification function. The dataset consists of images which are handwritten upper-case letters, lower-case letters and digits. Images from the EMNIST dataset will be used as the training set X. For more information on how the EMNIST dataset was created, see [24] for more details. Note that using this dataset is without loss of generality for the classification scheme and should be viewed as a proof-of-concept.

A.2

Parametric form of the classification function

Choosing the parametric form of the function C with respect to the weights is largely determined by which classification scheme is used. The focus here is to create an algorithm capable of classifying planar images of entries in $\mathcal{P}(D)$. Whilst a comparison between classification methods may provide insights into the algorithms themselves (which has also been covered in the literature, see [76, 110] for more information) we choose a method which has been shown to work well for our demonstrative purposes (see [93] for more details).

We choose to use Convolutional Neural Networks (CNNs) for parameterizing the classification function C. The inspiration and makeup of this parameterization comes from the origin of the perceptron, whose model is based on connectivity of biological neurons in the brain, forming a neural network. The models based on this are called artificial neural networks (ANNs). An ANN is made up of multiple *layers*, which are connected through weights and biases. In each layer, every neuron outputs a real number which is subsequently fed into the next layer as an input. The output of each neuron is a composite function of the previous input, weights and biases. Input from the previous layer undergoes an affine transformation using both weights and biases. The result is then fed into an 'activation' function, which generates a real number as output for that particular neuron. The affine transformation component is linear, meaning if the nonlinear behaviour in the input-output map is captured, the activation function must be nonlinear. Figure A.3 shows an example of an ANN with three layers. The inputs x_i and outputs c_i in the training process are known quantities, and the goal is to tweak the weights such that the composite function of x_i matches with its desired output. Each arrow in the figure is a multiplier of the previous input with a weight associated with that arrow. Each arrow pointing to a node is a weight that contributes to the overall activation of that particular neuron. Each neuron shown in figure A.3 computes an affine transformation using the weights and inputs connected to that particular neuron. An activation map is then applied to the affine transformation, resulting in a real number as output for that neuron. The overall activation map of a general ANN

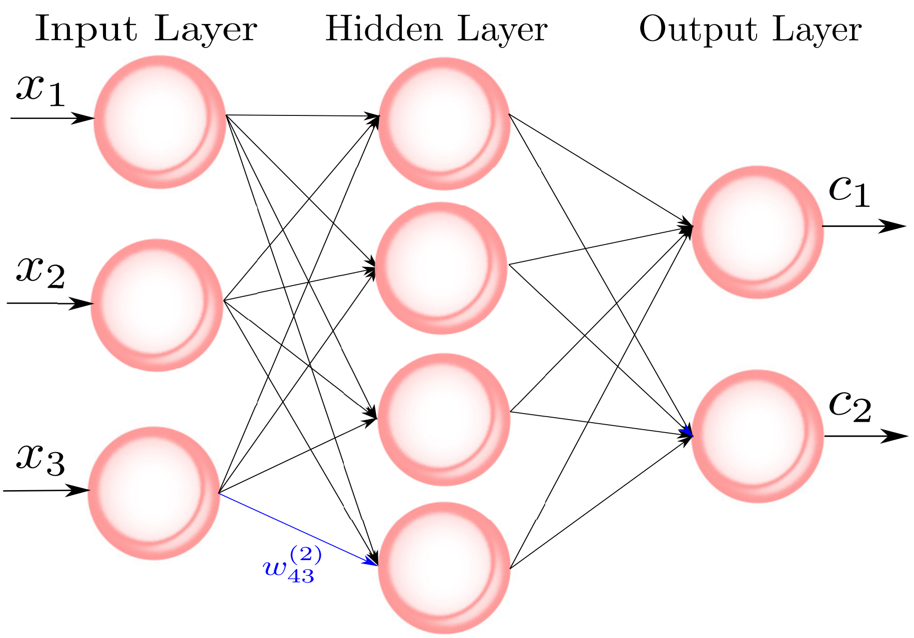


Figure A.3: Example of an ANN with 3 layers.

involving L layers can be defined as:

$$\begin{aligned} a^{(1)} &= \mathbf{x} \in \mathbb{R}^{n_1}; \\ a^{(l)} &= c_l(W^{(l)}a^{(l-1)} + b^{(l)}) \in \mathbb{R}^{n_l} \text{ for } l = 2,3,\dots,L, \end{aligned}$$

which opens up as

$$C(\mathbf{x}) = c_L(W^{(L)}(c_{L-1}(W^{(L-1)}(\dots c_2(W^2\mathbf{x}+b^2)+b^3)+b^4)\dots+b^{L-1})+b^L) \in \mathbb{R},$$

where the composite function C describes the entire network. Architecture of these networks are clearly vast, as there are many degrees of freedom. For example, c_l can take many forms, there is an infinite amount of combinations of neurons allocated to each layer and the number of outputs can be different (to mention a few). The general idea in convolutional neural networks are largely the same, and when originally developed were motivated by large input spaces. If one has a *fully connected* network, meaning each neuron in the previous layer passes information to every one in the next, then the order of magnitude of weights is large when the input space grows. This is problematic when trying to train a model as the amount of parameters which need to be tweaked is significantly large. A standard large input space, which we are interested in, are images. CNNs allow us to reduce dimensionality of these inputs whilst also trying to preserve relationships between pixels in the images themselves.

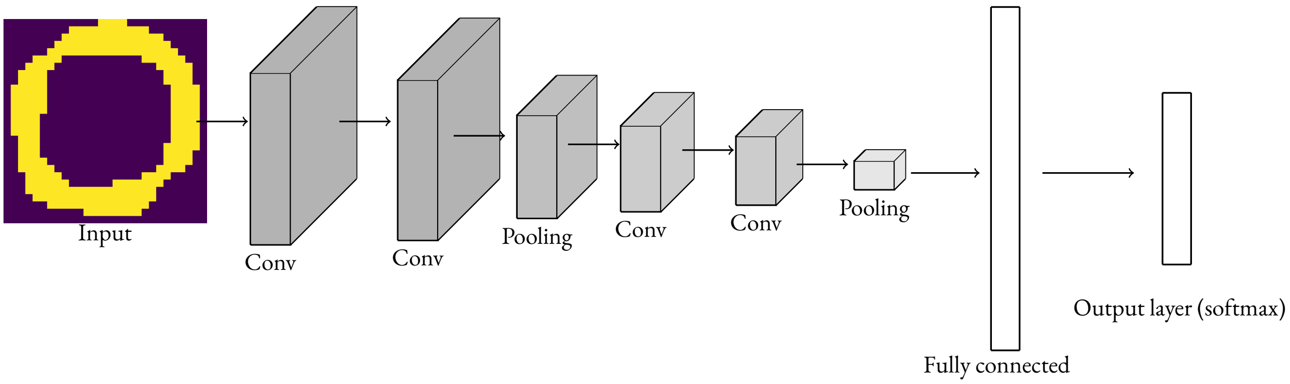


Figure A.4: CNN architecture used for EMNIST dataset. The labels 'Conv' represent a convolution layer.

They allow us to build classification functions for high resolution images which were otherwise not as well suited with standard ANNs. CNNs can be thought of as a sparse ANN.

CNNs themselves have been shown to have outstanding performances in a multitude of applications. Some of these include: image generation, semantic segmentation, image classification, facial recognition, object detection, amongst others. See [55, 73, 134] for more detail on some of those applications. Figure A.4 shows the CNN architecture which we use in our classification problem, it is a standard 'off the shelf' algorithm. The convolution layer is a stack of two-dimensional images which are created from applying *filters* over the input images. These are either trained or in-built into the function. For example, we could insist on two filters which when applied to the input images produce images which detect horizontal and vertical edges. Typically though, these filters form part of the optimization scheme when training the model. Although convolution layers increase the number of dimensions in the following layer, they reduce the number of pixels per stacked image. Applying further convolution layers and subsequently pooling layers, whose goal is to extract the most important information from local regions, reduces dimensionality of the problem such that we arrive in the end with a low dimensional fully-connected layer. A softmax function is then applied on the outputs of this fully-connected layer, producing probabilities for each classification label. In the training procedure, the weights associated with these connections between each layer's input and output are then tweaked such that the classification label associated with inputs in the training set has the highest probability in the output of the softmax function in the last layer. For the EMNIST dataset, the output in the network shown in figure A.4 will be a vector of probabilities whose entries are a probability between 0 and 1. A simple maximum function is then applied to the vector, taking the index of the vector with the largest result as the correct classification.

Here, we assume that the trained function is sufficiently accurate. Accuracy of classification functions for problems such as digit or number plate recognition are already accurate and can rival humans [22]. Whilst improving accuracy of classification functions has utility and should be done when possible, we do not actively seek marginal accuracy gains here since it is not the main idea which we are presenting. A pseudo-code of our 'on the fly' classification is shown in algorithm 7. Note that algorithm 7 is written in general form, as the training set can be any form of classification and the methods $\mathcal{P}$ can be either currently existing ones, as shown in chapter 2, or other reconstruction schemes which haven't been discussed.

Algorithm 7 Reconstruction classifier

```
procedure OFFLINE TRAINING
    Choose training set $X = \{(x_1,c_1),(x_2,c_2),\dots,(x_n,c_n)\}$
    Initialize total number of sweeps $S$ and select method $\mathcal{P}$, where $\mathcal{P}_s(\cdot)$ computes reconstruction using method $\mathcal{P}$ up to sweep $s$
    Train classification function $C$ by minimizing a form of (A.3) such that $C(x_i) \approx c_i$ for $i = 1,2,\dots,n$
procedure ON THE FLY CLASSIFICATION
    for $s = 0 : S-1$ do (loop over sweeps)
        $\sigma_s = \mathcal{P}_s(D)$
        $\text{class}_s = \text{mode}(\sigma_s)$
```

A.3

Classification experiments

In this section, we demonstrate algorithm 7 using two shape-based reconstruction schemes. We let the training set $X = X_{\text{EMNIST}}$, which consists of over $800,000$ images of handwritten lower/upper-case letters and digits. The corresponding classification task is designed such that the classification function assigns the input images to a number between 0 and 61 (each represents a lower/upper-case letter or single digit). In security screening applications, if we wanted to look for a certain object or shape, we could design X and a corresponding classification task such that it spots dangerous equipment or sharp objects inside the contents of a box or container.

Although the EMNIST dataset is designed for a 2D classification task, since it contains 2D images, and the reconstruction task finds 3D arrays represent the reconstructed conductivity profile, it is sufficient for the proof-of-concept style work considered here. In fact, there

Class	Digit/Character	Class	Character	Class	Character	Class	Character
0	0	16	G	32	W	48	m
1	1	17	H	33	X	49	n
2	2	18	I	34	Y	50	o
3	3	19	J	35	Z	51	p
4	4	20	K	36	a	52	q
5	5	21	L	37	b	53	r
6	6	22	M	38	c	54	s
7	7	23	N	39	d	55	t
8	8	24	O	40	e	56	u
9	9	25	P	41	f	57	v
10	A	26	Q	42	g	58	w
11	B	27	R	43	h	59	x
12	C	28	S	44	i	60	y
13	D	29	T	45	j	61	z
14	E	30	U	46	k		
15	F	31	V	47	l		

Table A.1: EMNIST classification labels.

are many things that can be done in this area of work, and typical ML techniques have so far not been applied in these types of situations.

We model a cube-shaped container by a domain $\Omega^h = [3] \times [3] \times [3]\text{m}^3$. The domain Ω^h is divided into $[35] \times [35] \times [20]$ uniform rectangular grid cells, each of them having dimensions roughly equivalent to $[0.085] \times [0.085] \times [0.15]\text{m}^3$. We choose $n_s = n_r = 16$ and the shielded walls to be parallel to the xy plane, and the same but shifted dimensions for the other sides of the container. The sources and receivers follow the distribution shown in figure 1.1. We follow the same setup for sources, receivers and thickness of electromagnetic shielding as the numerical experiments shown in chapter 4 (see figure 1.1). The true phantoms used here resemble 3D capital letters, and are chosen such that they could be ambiguous with other letters or digits. For example, a true phantom resembling a 'P' may actually classify as a 'D' from the reconstructed image. Figure A.5 shows a handful of example phantoms which resemble capital letters. As mentioned, the classification problem is 2D. We place 3D capital letters in the box/container such that it is invariant in the z direction, ready for the 2D classification. In reality, objects present inside boxes or containers may not be parallel to the axis, but it is sufficient in this proof-of-concept work.

Table A.1 shows class labels and their corresponding characters or digits in the classification task. When showing numerical results, we will refer to these class labels when C classifies images. Since the classification problem is 2D, we classify each image in the xy plane, creating a vector of classifications. Therefore, we let

$$\mathbf{c}_s = [C(\sigma_{s,1}), C(\sigma_{s,2}), \ldots, C(\sigma_{s,N_z})]^T$$

and

$$\text{class}_s = \text{mode}(\mathbf{c}_s),$$

where $\sigma_{s,m}$ is a 2D image at sweep s, at the mth cell in the z direction. The vector $\mathbf{c}_s$ contains outputs from classified 2D images and the mode$(\cdot)$ operator outputs the most frequent class label as sweep s's classification from the vector $\mathbf{c}_s$.

Here, we consider the shape-based schemes as the reconstruction methods. They segment the pixel-based schemes during the inversion scheme, and therefore lend themselves well to classification problems such as this one. We consider both single and color level set regimes (defined in algorithms 5 and 6 respectively) for letters ‘L’, ‘O’ and ‘T’.

It is possible to consider pixel-based methods for these problems by using image segmentation, though we do not consider this.

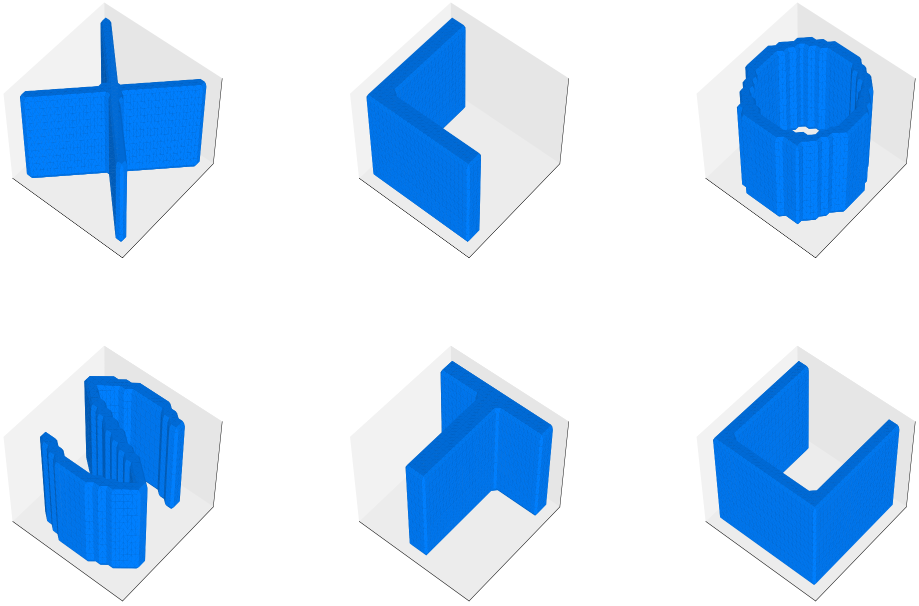


Figure A.5: Some examples of 3D capital letter true phantoms which can be used to demonstrate the classification technique.
Top Row: ‘I’, ‘L’, ‘O’.
Bottom Row: ‘S’, ‘T’, ‘U’.

Figures A.6 - A.11 show reconstructions of true phantoms resembling ‘L’, ‘O’ and ‘T’ using single and color level set methods and their real time classification during the inversion. We show both methods reconstructing each letter at a time, i.e. figures A.6 and A.7 show single and color level set reconstructions for a letter 'L' respectively. Figure A.6 shows a single level set inversion of a true phantom resembling a 3D capital 'L'. In the top part of the figure, snapshots of the reconstructed shape are shown at different sweep numbers. In the bottom part of the figure, information from the classification algorithm is presented for the full duration of the inversion. In this figure, the top sub plot shows the mode classification at each sweep with the colour representing how many of the 2D slices take the mode classifier. The middle sub plot shows the pseudo-cost at each sweep number and the bottom sub plot shows the top three classifications over the algorithm's lifespan. Note that all three sub plots have something in common; each settle down after a certain amount of iterations to a particular result. The classification algorithm settles at the mode classification being equal to 21, which from observation of Table A.1, is the correct classification. Moreover, the pseudo-cost begins to converge toward a constant value and class label 21 begins to dominate in the top three classifier plot. The algorithm seems to converge in both the classification and pseudo-cost. In the early phases of the inversion, the classification scheme is fighting between class labels 5, 16 and 21, which from observation of Table 4.2, refer to '6', 'G' and 'L' respectively. By observing the design of these letters, we can see why ambiguities could arise in the classification. From observation of the inversion at sweep $s = 5$ we can see why this could classify as '6' or 'G' because of its banana-like appearance. These early stages can be seen as a 'burn-in' phase, since the shape associated with the level set function is likely to change dramatically in subsequent iterations due to the cost minimization task. After roughly 50 iterations, the pseudo-cost begins to reduce marginally and the class label 21 dominates in the classification task.

Figure A.7 shows a reconstruction of the same true phantom using the color level set scheme. The color level set scheme in this case is penalized in its *a priori* information as we guess a conductivity value twice what is present for one of the shapes. Although we have this error, with holes in the shape appearing, the algorithm overall correctly classifies the shape as class label 21. Though in comparison with figure A.6, the classification is more erratic. This is possibly due to holes appearing in the shape, affecting certain slices.

Figure A.8 shows a single level set reconstruction of a 3D 'O'. By observation of the reconstructed shape snapshots, the algorithm performs well in recovery of the hole and exterior of the cylinder. Whilst we observe good recovery and convergence of the pseudo-cost, the classification task is unsure between class labels 13, 24 and 26 which from Table 4.2 represent

'D', 'O' and 'Q' respectively. Just by observation of the appearance of these letters, ambiguities between all these can exist, therefore any artefacts which appear in the reconstructed shape can throw the classification task from the correct class label 24 to neighbouring classifiers 13 and 26. It is also evident from the snapshots why these class labels appear in the classification task, as there is a slight bulge of the shape in some of the sweep snapshots. Figure A.9 shows a color level set reconstruction for the same true phantom. Unlike in the 'L' reconstruction, the inversion suffers heavily. Holes and gaps between the two shape domains appear and it is no surprise then that the classification algorithm generates no distinct class label as the pseudo-cost lowers, with the algorithm fighting between class labels 12, 16, 19 and 21 which from Table 4.2 resemble 'C', 'G', 'J' and 'L' respectively. From the reconstructed shape snapshots, we can see why the classification task favours these classifiers. In particular, the reconstructed shape resembles a cylinder with holes which results in some 2D cross sections resembling incomplete circles.

Figure A.10 shows a single level set reconstruction of a 3D 'T'. From observation of the snapshots, the algorithm performs well at locating the true shape and retains its main features. This is to some degree also reflected in the classification scheme, though the algorithm is unsure between class labels 15, 29 and 45 which from Table 4.2 resemble 'F', 'T' and 'j' respectively. Just like the other experiments, it is evident from the class labels themselves why the classification process struggles. Figure A.11 shows a color level set reconstruction for the same true phantom. Despite incorrect *a priori* information, the reconstructed shape appears to resemble the true phantom. As typical to all color level set inversions in these experiments, holes appear in the shapes. Results from the classification task tell us that the reconstructed shape is not the correct class label 29 ('T') but either class label 15, 18, 44 or 45 ('F','I','i' and 'j' respectively), which seem reasonable. In comparison to the other two color level set classifications, this appears more uncertain than the others. This could be due to the fact that 'T' has a lot of similarity with other letters.

The classification experiments described here show that it is achievable to correctly identify shapes within enclosed containers or boxes using near-field electromagnetic imaging. In this section, we have demonstrated algorithm 7 for a subset of methods $\mathcal{P}$, which are strictly shape-based and current state-of-the-art for our application. We predict that using pixel-based reconstructions coupled with image segmentation in the classification scheme could result in quicker convergence to class label as pixel-based schemes update conductivity globally, therefore distinctive features of the true phantom are already present in the reconstructed conductivity profile after just one sweep.

Rich datasets for training 3D classification algorithms are difficult to obtain. In general,

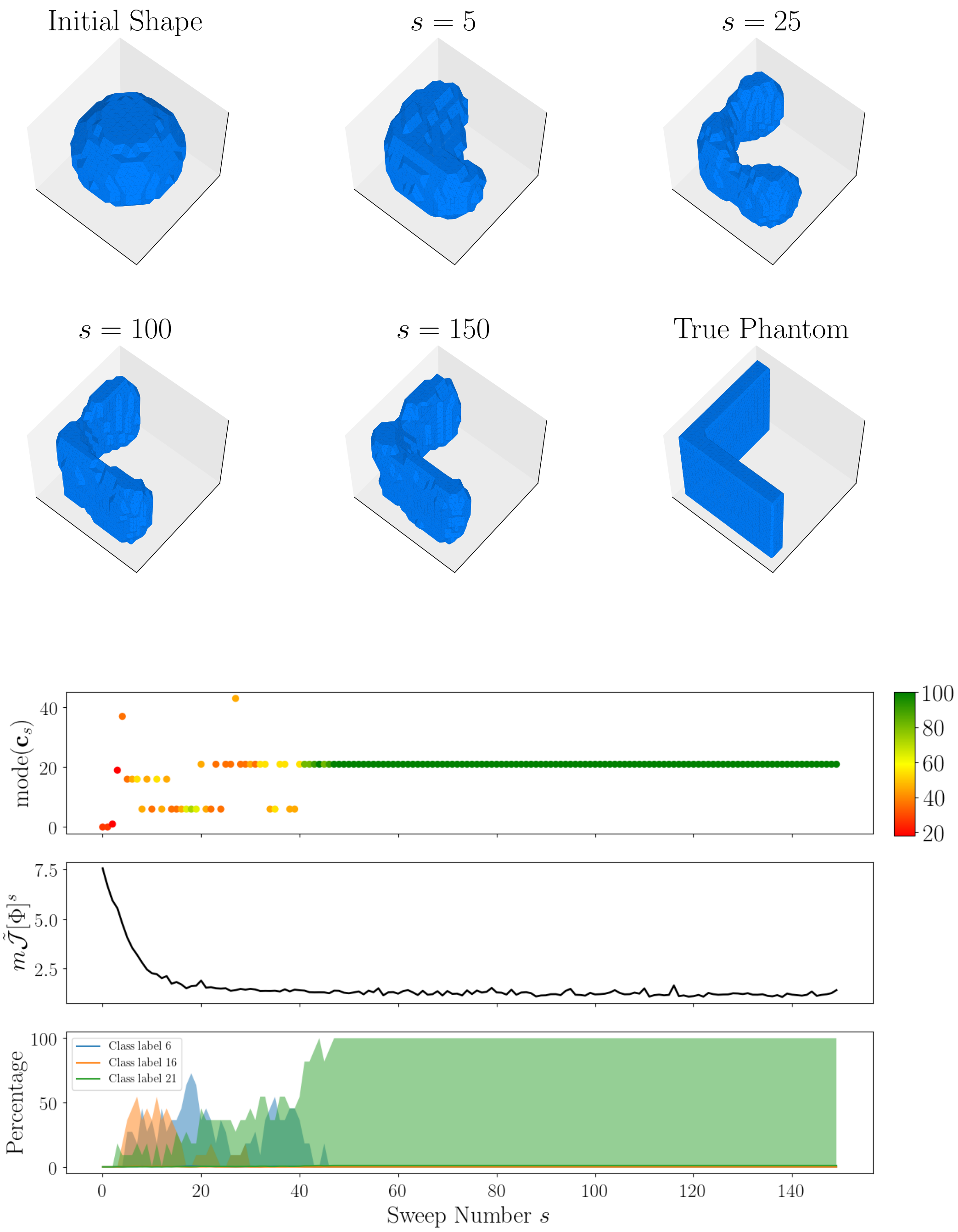


Figure A.6: Top: Surface plots of 3D shape evolution using LK-Single level set reconstruction. Shown are the initial shape, snapshots at sweeps $s = 5, 25, 100, 150$ and a true phantom which resembles capital ‘L’.
Bottom: Results of applying classification function C to individual 2D slices. First plot shows the mode classifier over the sweep number with a percentage attached to that mode. Second plot shows the corresponding pseudo-cost and the third shows the top three classes which are output from the classification function C.

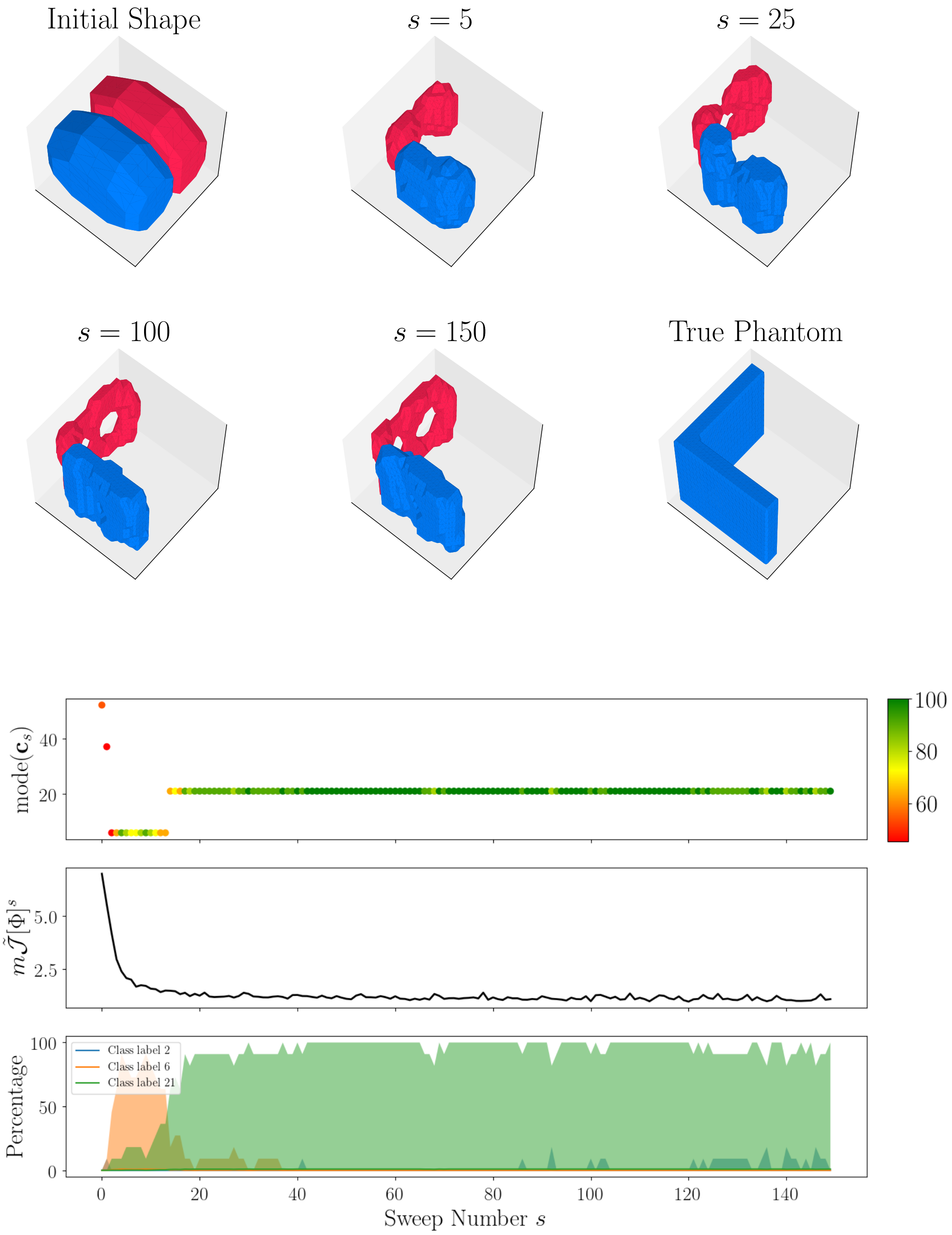


Figure A.7: Top: Surface plots of 3D shape evolution using LK-Color level set reconstruction. Shown are the initial shape, snapshots at sweep numbers $s = 5, 25, 100, 150$ and a true phantom which resembles a capital 'L'.
Bottom: Results of applying classification function C to individual 2D slices. First plot shows the mode classifier over the sweep number with a percentage attached to that mode. Second plot shows the corresponding pseudo-cost and the third shows the top three classes which are output from the classification function C.

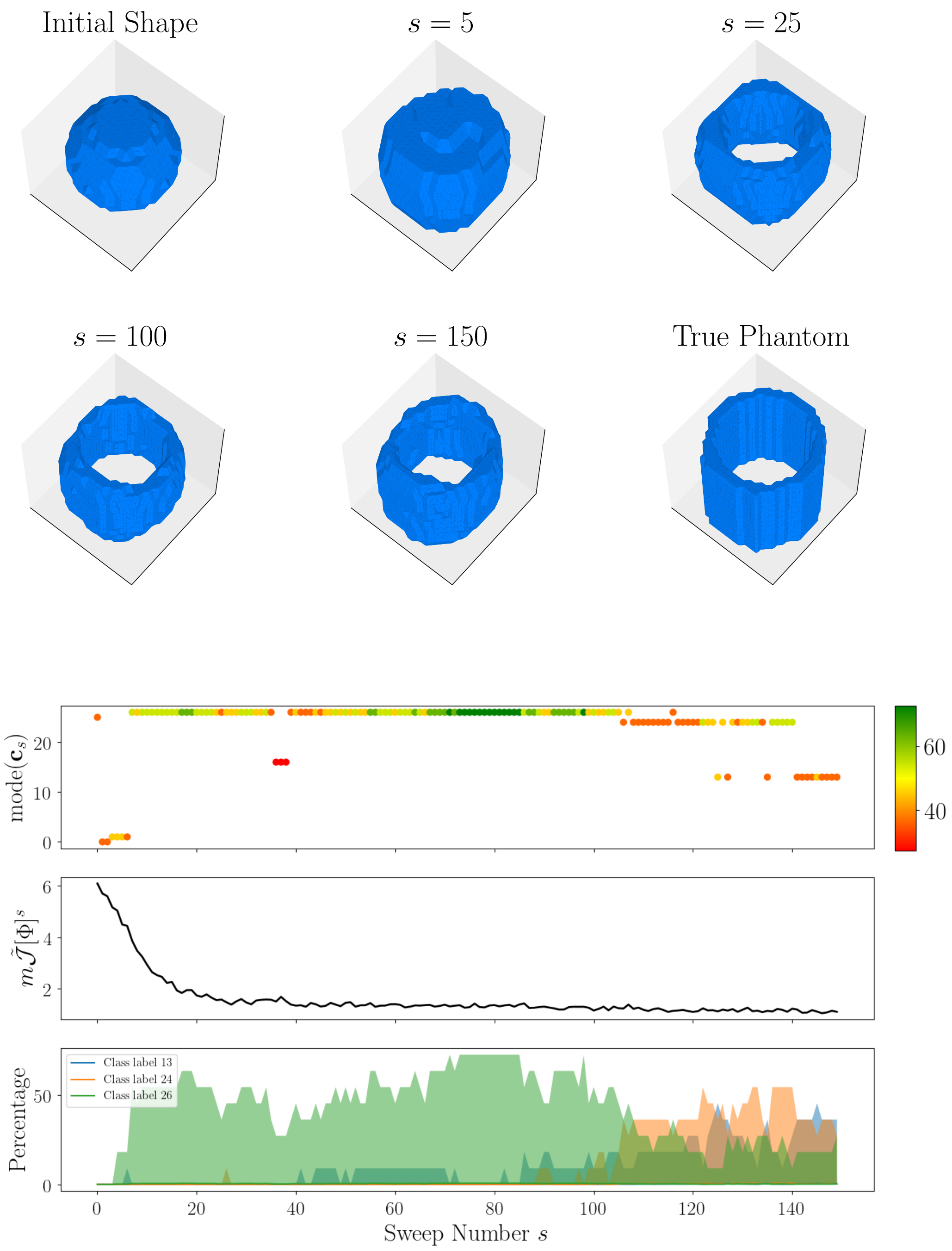


Figure A.8: Top: Surface plots of 3D shape evolution using LK-Single level set reconstruction. Shown are the initial shape, snapshots at sweep numbers $s = 5, 25, 100, 150$ and a true phantom which resembles a capital 'O'.
Bottom: Results of applying classification function C to individual 2D slices. First plot shows the mode classifier over the sweep number with a percentage attached to that mode. Second plot shows the corresponding pseudo-cost and the third shows the top three classes which are output from the classification function C.

Figure A.9: Top: Surfaces plots of 3D shape evolution using LK-Color level set reconstruction. Shown are the initial shape, snapshots at sweep numbers $s = 5, 25, 100, 150$ and a true phantom which resembles a capital 'O'.
Bottom: Results of applying classification function C to individual 2D slices. First plot shows the mode classifier over the sweep number with a percentage attached to that mode. Second plot shows the corresponding pseudo-cost and the third shows the top three classes which are output from the classification function C.

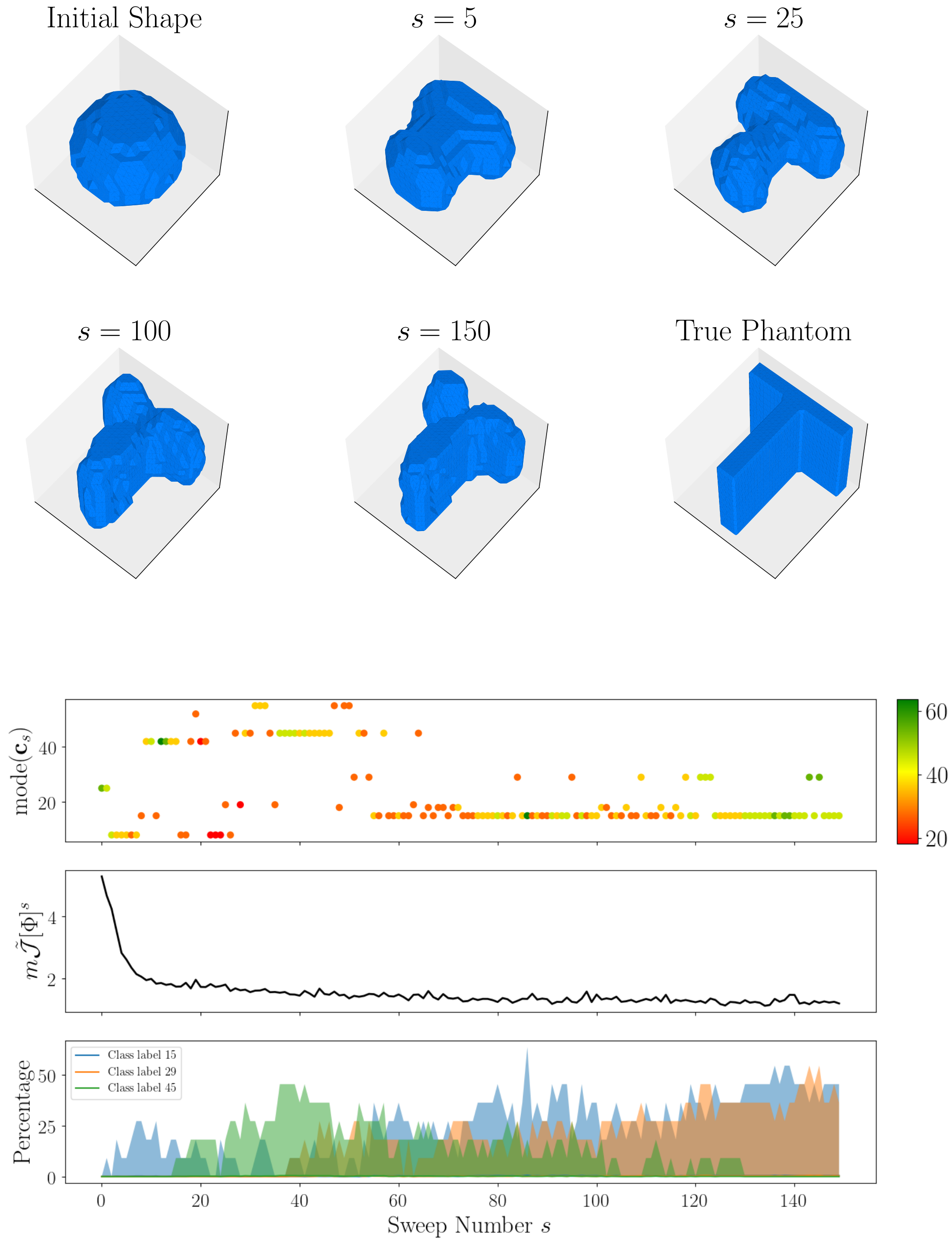


Figure A.10: Top: Surface plots of 3D shape evolution using LK-Single level set reconstruction. Shown are the initial shape, snapshots at sweep numbers $s = 5, 25, 100, 150$ and a true phantom which resembles a capital 'T'.
Bottom: Results of applying classification function C to individual 2D slices. First plot shows the mode classifier over the sweep number with a percentage attached to that mode. Second plot shows the corresponding pseudo-cost and the third shows the top three classes which are output from the classification function C.

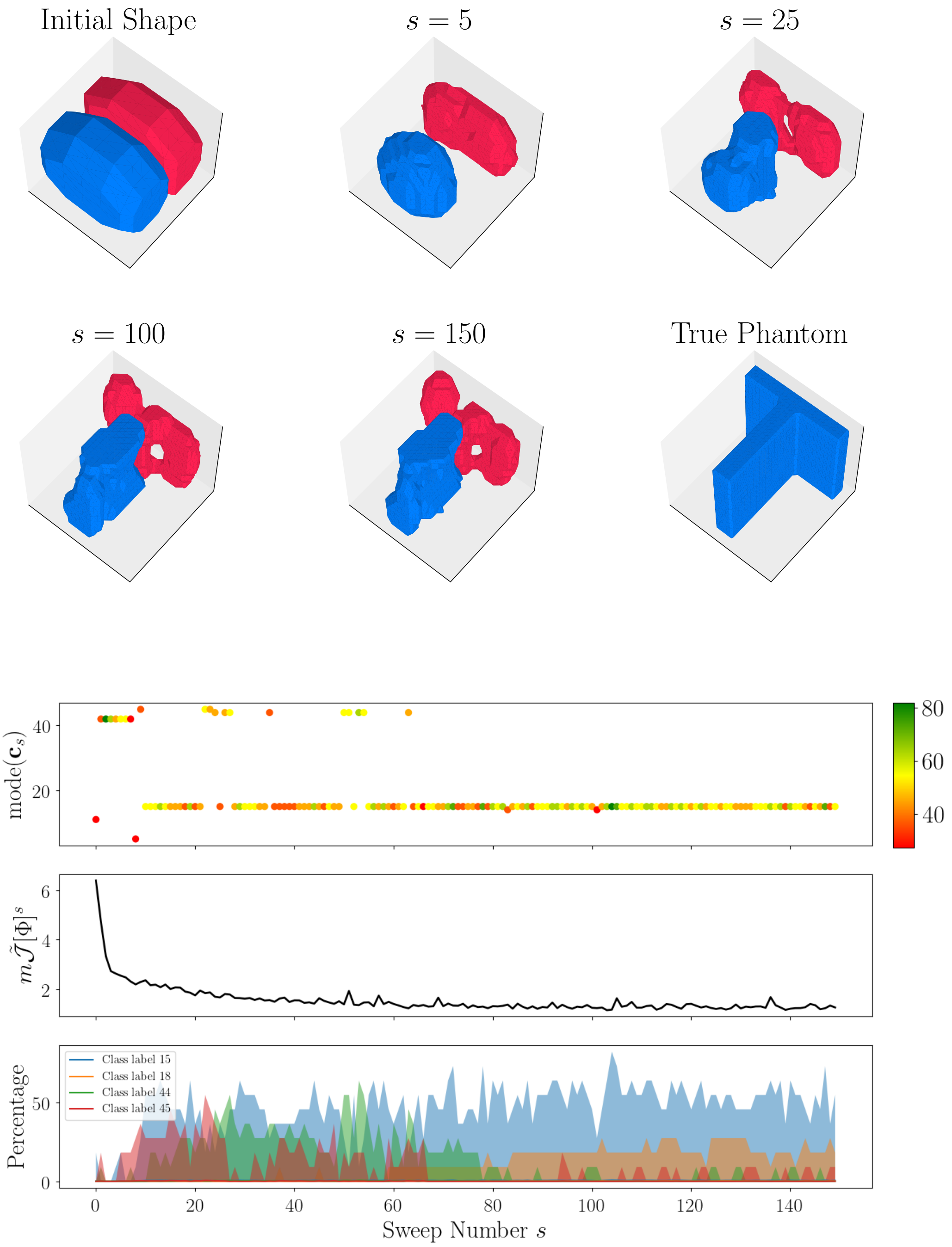


Figure A.11: Top: Surface plots of 3D shape evolution using LK-Color level set reconstruction. Shown are the initial shape, snapshots at sweep numbers $s = 5, 25, 100, 150$ and a true phantom which resembles a capital 'T'.
Bottom: Results of applying classification function C to individual 2D slices. First plot shows the mode classifier over the sweep number with a percentage attached to that mode. Second plot shows the corresponding pseudo-cost and the third shows the top three classes which are output from the classification function C.

generating 3D datasets which result in a robust classification algorithm after training is a difficult problem in its own right. In fact, in the ML community it is common to find papers on how datasets were created [24]. Nevertheless, the pseudo-3D scheme demonstrates the idea of classifying electromagnetic reconstructions of boxes and containers despite not using 3D datasets for training. The difficulty of generating 3D datasets increases when the classification task is of a sensitive nature, such as in security screening, since the resulting classification task must have integrity and be robust. In these applications, the classification scheme could be trained on images which resemble certain objects which would be flagged up during a security check, whether that be scanning a cargo container or luggage in an airport. However, the classification algorithm is only as good what it sees in the training set, therefore this would have to be vast and diverse to try and cover all possibilities, if it was to be put into practice.

It is encouraging that in some situations the classification scheme performs well, since after all electromagnetic inversion schemes should improve over time. Furthermore, all reconstructions shown are generated from access to limited data (approximately 1% of the total number of unknowns), therefore reconstructions from current state-of-the-art methods could improve with access to more data, which may result in improvements to the classification.

Bibliography

[1] A. Abubakar and P. van den Berg. Iterative forward and inverse algorithms based on domain integral equations for three-dimensional electric and magnetic objects. *J. Comput. Phys.*, 195(1):236–262, 2004.

[2] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[3] E. Al Hosani and M. Soleimani. Multiphase permittivity imaging using absolute value electrical capacitance tomography data and a level set algorithm. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2070):20150332, 2016.

[4] H. Ammari, J. Chen, Z. Chen, J. Garnier, and D. Volkov. Target detection and characterization from electromagnetic induction data. *Journal de mathématiques pures et appliquées*, 101(1):54–75, 2014.

[5] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28(-):1–174, 2019.

[6] D. Aruliah. *Fast solvers for time-harmonic Maxwell's equations in 3D*. PhD thesis, University of British Columbia, 2001.

[7] D. Aruliah, U. Ascher, and D. Oldenburg. A method for the forward modelling of

3-D electromagnetic quasi-static problems. *Mathematical Models and Methods in Applied Sciences*, 11(1):1–21, 2001.

[8] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[9] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[10] A. Bondeson, T. Rylander, and P. Ingelström. *Computational Electromagnetics*. Springer, 2005.

[11] T. Bonesky, K. Bredies, D. Lorenz, and P. Maass. A generalized conditional gradient method for nonlinear operator equations with sparsity constraints. *Inverse Problems*, 23(5):2041–2058, 2007.

[12] A. Bossavit. Solving Maxwell equations in a closed cavity, and the question of 'spurious modes'. *IEEE Transactions on magnetics*, 26(2):702–705, 1990.

[13] L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[14] J. R. Bowler and S. J. Norton. Eddy current inversion for layered conductors. *Research in Nondestructive Evaluation*, 4(4):205–219, 1992.

[15] B. Brown, M. Marletta, and J. Reyes. Uniqueness for an inverse problem in electromagnetism with partial data. *Journal of Differential Equations*, 260(8):6525–6547, 2016.

[16] M. Burger. A level set method for inverse problems. *Inverse Problems*, 17(5):1327–1355, 2001.

[17] M. Burger and S. Osher. A survey on level set methods for inverse problems and optimal design. *European J. Appl. Math.*, 16(2):263–301, 2005.

[18] A. P. Calderón. On an inverse boundary value problem. *Comput. Appl. Math*, pages 2–3.

[19] M. Campbell, A. J. Hoane Jr, and F.-h. Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[20] P. Caro, P. Ola, and M. Salo. Inverse boundary value problem for Maxwell equations with local data. *Communications in Partial Differential Equations*, 34(11):1425–1464, 2009.

[21] N. J. Champagne II, J. G. Berryman, and H. M. Buettner. Fdfd: A 3D finite-difference frequency-domain code for electromagnetic induction tomography. *Journal of Computational Physics*, 170(2):830–848, 2001.

[22] L. Chen, S. Wang, W. Fan, J. Sun, and S. Naoi. Beyond human recognition: A CNN-based framework for handwritten character recognition. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 695–699. IEEE, 2015.

[23] X. Chen. *Computational Methods for Electromagnetic Inverse Scattering*. IEEE Press. John Wiley & Sons Singapore, 2018.

[24] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[25] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*, volume 93 of *Applied Mathematical Sciences*. Springer Science and Business Media, 3 edition, 2013.

[26] S. J. Cooke, R. Shtokhamer, A. A. Mondelli, and B. Levush. A finite integration method for conformal, structured-grid, electromagnetic simulation. *Journal of computational physics*, 215(1):321–347, 2006.

[27] P. Cunningham and S. J. Delany. k-Nearest neighbour classifiers. *Multiple Classifier Systems*, 34(8):1–17, 2007.

[28] B. Darrer. *Electromagnetic induction imaging through metallic shields*. PhD thesis, University College London, 2017.

[29] B. Darrer, J. Watson, P. Bartlett, and F. Renzoni. Magnetic Imaging: A New Tool for uk National Nuclear Security. *Scientific reports*, 5(-):7944, 2015.

[30] B. Darrer, J. Watson, P. Bartlett, and F. Renzoni. Toward an Automated Setup for Magnetic Induction Tomography. *IEEE Transactions on Magnetics*, 51(1):1–4, 2015.

[31] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[32] Y. Deng and X. Liu. Electromagnetic imaging methods for nondestructive evaluation applications. *Sensors*, 11(12):11774–11808, 2011.

[33] A. Ditkowski, K. Dridi, and J. S. Hesthaven. Convergent Cartesian grid methods for Maxwell's equations in complex geometries. *Journal of Computational Physics*, 170(1):39–80, 2001.

[34] C. Dodd and W. Deeds. Analytical solutions to eddy-current probe-coil problems. *Journal of applied physics*, 39(6):2829–2838, 1968.

[35] O. Dorn and U. Ascher. Shape reconstruction in 3D electromagnetic induction tomography using a level set technique. *Proc. 23rd International Review of Progress in Applied Computational Electromagnetics ACES*, pages 1–6, 2007.

[36] O. Dorn, H. Bertete-Aguirre, J. Berryman, and G. Papanicolaou. A nonlinear inversion method for 3D electromagnetic imaging using adjoint fields. *Inverse problems*, 15(6):1523, 1999.

[37] O. Dorn, H. Bertete-Aguirre, J. Berryman, and G. Papanicolaou. Sensitivity analysis of a nonlinear inversion method for 3d electromagnetic imaging in anisotropic media. *Inverse problems*, 18(2):285, 2002.

[38] O. Dorn and A. Hiles. A level set method for magnetic induction tomography of 3D boxes and containers. In D. Lesselier and C. Reboud, editors, *Electromagnetic Non-Destructive Evaluation (XXI)*, pages 33–40. IOS Press, 2018.

[39] O. Dorn and D. Lesselier. Level set methods for inverse scattering. *Inverse Problems*, 22:R67–R131, 2006.

[40] O. Dorn and D. Lesselier. Level set techniques for structural inversion in medical imaging. In *Deformable Models*, pages 61–90. Springer, 2007.

[41] O. Dorn and D. Lesselier. Level set methods for inverse scattering—some recent developments. *Inverse Problems*, 25(12):125001, 2009.

[42] O. Dorn, E. Miller, and C. Rappaport. A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets. *Inverse Problems*, 16(5):492–506, 2000.

[43] O. Dorn and K. Prieto. From data to images: A shape based approach for fluorescence tomography. In *Science: Image In Action*, pages 255–266. World Scientific, 2012.

[44] O. Dorn and R. Villegas. History matching of petroleum reservoirs using a level set technique. *Inverse Problems*, 24(3):035015, 2008.

[45] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Birkhauser, 2013.

[46] M. Fusco. FDTD algorithm in curvilinear coordinates (EM scattering). *IEEE Transactions on Antennas and Propagation*, 38(1):76–89, 1990.

[47] V. Girault and P.-A. Raviart. *Finite element approximation of the Navier-Stokes equations*, volume 749. 1979.

[48] P. González-Rodríguez, M. Kindelan, M. Moscoso, and O. Dorn. History matching problem in reservoir engineering using the propagation-backpropagation method. *Inverse Problems*, 21(2):565–590, 2005.

[49] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[50] E. Haber. Quasi-newton methods for large scale electromagnetic inverse problems. *Inverse Problems*, 21(1):305–323, 2005.

[51] E. Haber and U. Ascher. Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients. *SIAM Journal on Scientific Computing*, 22(6):1943–1961, 2001.

[52] E. Haber, U. Ascher, D. Aruliah, and D. Oldenburg. Fast simulation of 3D electromagnetic problems using potentials. *Journal of Computational Physics*, 163(1):150–171, 2000.

[53] J. Häggblad and O. Runborg. Accuracy of staircase approximations in finite-difference methods for wave propagation. *Numerische Mathematik*, 128(4):741–771, 2014.

[54] B. Harrach and M. Ullrich. Monotonicity-based shape reconstruction in electrical impedance tomography. *SIAM Journal on Mathematical Analysis*, 45(6):3382–3403, 2013.

[55] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[56] F. Hettlich. The domain derivative of time-harmonic electromagnetic waves at interfaces. *Mathematical Methods in the Applied Sciences*, 35(14):1681–1689, 2012.

[57] C. F. Higham and D. J. Higham. Deep learning: An introduction for applied mathematicians. *SIAM Review*, 61(4):860–891, 2019.

[58] A. J. Hiles and O. Dorn. Sparsity and level set regularization for near-field electromagnetic imaging in 3D. *Inverse Problems*, 36(2):025012, 2020.

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

[60] B. Hofmann, A. Leitão, and J. P. Zubelli. *New Trends in Parameter Identification for Mathematical Models*. Springer, 2018.

[61] R. Holland. Finite-difference solution of Maxwell's equations in generalized nonorthogonal coordinates. *IEEE Transactions on Nuclear Science*, 30(6):4589–4591, 1983.

[62] S. Holman and P. Richardson. Emission tomography with a multi-bang assumption on attenuation. *arXiv preprint arXiv:2001.04190*, 2020.

[63] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[64] U. S. Inan and R. A. Marshall. *Numerical Electromagnetics: the FDTD method*. Cambridge University Press, 2011.

[65] N. Irishina, D. Álvarez, O. Dorn, and M. Moscoso. Structural level set inversion for microwave breast screening. *Inverse Problems*, 26(3):035015, 2010.

[66] K. Ito, K. Kunisch, and Z. Li. Level-set function approach to an inverse interface problem. *Inverse Problems*, 17(5):1225–1242, 2001.

[67] B. Jin, T. Khan, and P. Maass. A reconstruction algorithm for electrical impedance tomography based on sparsity regularization. *International Journal for Numerical Methods in Engineering*, 89(3):337–353, 2012.

[68] B. Jin and P. Maass. Sparsity regularization for parameter identification problems. *Inverse Problems*, 28(12):123001, 70pp, 2012.

[69] T. Kluth. 3D electrical impedance tomography with sparsity constraints: Algorithm and implementation in application to the complete electrode model. Master's thesis, University of Bremen, 2011.

[70] S. B. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, 2013.

[71] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes. RMDL: Random Multimodel Deep Learning for Classification. In *Proceedings of the 2nd International Conference on Information System and Data Mining*, pages 19–28, 2018.

[72] E. E. Kriezis and T. P. Theodoulidis. *Eddy Current Canonical Problems (with Applications to Nondestructive Evaluation).* Tech Science Press, 2006.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[74] P. D. Ledger and W. R. B. Lionheart. Characterizing the shape and material properties of hidden targets from magnetic induction data. *The IMA Journal of Applied Mathematics*, 80(6):1776–1798, 2015.

[75] A. Leitão and O. Scherzer. On the relation between constraint regularization, level sets, and shape optimization. *Inverse Problems*, 19(1):L1–L11, 2003.

[76] J. Lemley, S. Abdul-Wahid, D. Banik, and R. Andonie. Comparison of recent machine learning techniques for gender recognition from facial images. In *MAICS*, pages 97–102, 2016.

[77] R. Ling. A finite-difference frequency-domain (FD-FD) approach to electromagnetic scattering problems. *Journal of electromagnetic waves and applications*, 3(2):107–128, 1989.

[78] A. Litman, D. Lesselier, and D. Santosa. Reconstruction of a two-dimensional binary obstacle by controlled evolution of a level-set. *Inverse Problems*, 14(3):485–706, 1998.

[79] D. Liu, A. K. Khambampati, S. Kim, and K. Y. Kim. Multi-phase flow monitoring with electrical impedance tomography using level set based method. *Nuclear Engineering and Design*, 289:108–116, 2015.

[80] L. Ma and M. Soleimani. Magnetic induction tomography methods and applications: a review. *Measurement Science and Technology*, 28(7):072001, 2017.

[81] I. Mayergoyz. Boundary integral equations of minimum order for the calculation of three-dimensional eddy current problems. *IEEE Transactions on Magnetics*, 18(2):536–539, 1982.

[82] P. McGillivray, D. Oldenburg, R. Ellis, and T. Habashy. Calculation of sensitivities for the frequency-domain electromagnetic problem. *Geophys.J.Int.*, 116(1):1–4, 1994.

[83] T. Misaki and H. Tsuboi. Computation of 3-dimensional eddy current problems by using boundary element method. *IEEE Transactions on magnetics*, 21(6):2227–2230, 1985.

[84] A. H. Mohammadian, V. Shankar, and W. F. Hall. Computation of electromagnetic scattering and radiation using a time-domain finite-volume discretization procedure. *Computer Physics Communications*, 68(1-3):175–196, 1991.

[85] P. Monk and E. Suli. Error estimates for Yee's method on non-uniform grids. *IEEE Transactions on Magnetics*, 30(5):3200–3203, 1994.

[86] U. J. Muehlematter, M. Mannil, A. S. Becker, K. N. Vokinger, T. Finkenstaedt, G. Osterhoff, M. A. Fischer, and R. Guggenberger. Vertebral body insufficiency fractures: detection of vertebrae at risk on standard CT images using texture analysis and machine learning. *European radiology*, 29(5):2207–2217, 2019.

[87] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.

[88] G. Mur. Edge elements, their advantages and their disadvantages. *IEEE transactions on magnetics*, 30(5):3552–3557, 1994.

[89] F. Natterer, H. Sielschott, O. Dorn, T. Dierkes, and V. Palamodov. Fréchet derivatives for some bilinear inverse problems. *SIAM journal on applied mathematics*, 62(6):2092–2113, 2002.

[90] F. Natterer and F. Wuebbeling. *Mathematical Methods in Image Reconstruction*. Mathematical Modeling and Computation Series. SIAM, 2001.

[91] J. Neuberger. *Sobolev Gradients and Differential Equations*. Springer-Verlag, 1997.

[92] A. Nicolas. 3D eddy current solution by BIE techniques. *IEEE Transactions on Magnetics*, 24(1):130–133, 1988.

[93] X.-X. Niu and C. Y. Suen. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.

[94] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science and Business Media, 2 edition, 2006.

[95] S. J. Norton and J. R. Bowler. Theory of eddy current inversion. *Journal of Applied Physics*, 73(2):501–512, 1993.

[96] P. Ola, L. Päivärinta, and E. Somersalo. An inverse boundary value problem in electrodynamics. *Duke Mathematical Journal*, 70(3):617–653, 1993.

[97] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, 2003.

[98] O. Podebrad, M. Clemens, and T. Weiland. New flexible subgridding scheme for the finite integration technique. *IEEE transactions on magnetics*, 39(3):1662–1665, 2003.

[99] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Numerical recipes. 3:294–301, 1989.

[100] K. Prieto and O. Dorn. Sparsity and level set regularization for diffuse optical tomography using a transport model in 2D. *Inverse Problems*, 33(1):014001, 2016.

[101] B. M. Project, A. Erdélyi, and H. Bateman. *Tables of integral transforms: based in part on notes left by Harry Bateman and compiled by the staff of the Bateman manuscript project*. McGraw-Hill, 1954.

[102] Z. Rahimi. *The finite integration technique (FIT) and the application in lithography simulations*. PhD thesis, 2011.

[103] W. Rucker and K. Richter. A BEM code for 3-D eddy current calculations. *IEEE Transactions on Magnetics*, 26(2):462–465, 1990.

[104] S. Sainson. *Electromagnetic Seabed Logging: A new tool for geoscientists*. Springer International Publishing Switzerland, 2017.

[105] F. Santosa. A level-set approach for inverse problems involving obstacles. *ESAIM: Control, Optimisation and Calculus of Variations*, 1(-):17–33, 1996.

[106] H. G. Schantz, C. Weil, and A. H. Unden. Characterization of error in a near-field electromagnetic ranging (NFER) real-time location system (RTLS). In *2011 IEEE Radio and Wireless Symposium*, pages 379–382. IEEE, 2011.

[107] J. A. Sethian. *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*, volume 1999. Cambridge University Press Cambridge, 1996.

[108] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova. Machine learning for predictive modelling based on small data in biomedical engineering. *IFAC-PapersOnLine*, 48(20):469–474, 2015.

[109] K. Shao, K. Zhou, and J. Lavers. Boundary element analysis method for 3-D eddy current problems using the second order vector potential. *IEEE transactions on magnetics*, 27(5):4089–4092, 1991.

[110] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[111] K. Sharma, A. Kaur, and S. Gujral. Brain tumor detection based on machine learning algorithms. *International Journal of Computer Applications*, 103(1), 2014.

[112] W. Shin. *3D finite-difference frequency-domain method for plasmonics and nanophotonics*. PhD thesis, Stanford University, 2013.

[113] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[114] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[115] A. Sinha, J. Lee, S. Li, and G. Barbastathis. Lensless computational imaging through deep learning. *Optica*, 4(9):1117–1125, 2017.

[116] M. Soleimani. *Image and shape reconstruction methods in magnetic induction and electrical impedance tomography*. PhD thesis, The University of Manchester, 2005.

[117] M. Soleimani and W. R. B. Lionheart. Absolute conductivity reconstruction in magnetic induction tomography using a nonlinear method. *IEEE Transactions on medical imaging*, 25(12):1521–1530, 2006.

[118] M. Soleimani, W. R. B. Lionheart, and O. Dorn. Level set reconstruction of conductivity and permittivity from boundary electrical measurements using experimental data. *Inverse Problems in Science and Engineering*, 14(2):193–210, 2006.

[119] E. Somersalo, D. Isaacson, and M. Cheney. A linearized inverse boundary value problem for Maxwell's equations. *Journal of Computational and Applied Mathematics*, 42(1):123–136, 1992.

[120] E. Somersalo, D. Isaacson, and M. Cheney. A linearized inverse boundary value problem for Maxwell's equations. *Journal of computational and applied mathematics*, 42(1):123–136, 1992.

[121] A. Taflove. Application of the finite-difference time-domain method to sinusoidal steady-state electromagnetic-penetration problems. *IEEE Transactions on electromagnetic compatibility*, EMC-22(3):191–202, 1980.

[122] A. Tamburrino, F. Calvano, S. Ventre, and G. Rubinacci. Non-iterative imaging method for experimental data inversion in eddy current tomography. *NDT & E International*, 47:26–34, 2012.

[123] A. Tamburrino, G. Rubinacci, M. Soleimani, and W. R. B. Lionheart. A noniterative inversion method for electrical resistance, capacitance and inductance tomography for two phase materials. In *Proc. 3rd World Congress on Industrial Process Tomography, Canada*, pages 233–238, 2003.

[124] A. Tamburrino, Z. Sua, S. Ventre, L. Udpa, and S. S. Udpa. Monotonicity based imang method in time domain eddy current testing. *Electromagnetic Nondestructive Evaluation (XIX)*, 41(1), 2016.

[125] A. N. Tikhonov and V. I. Arsenin. *Solutions of ill-posed problems*. Winston, Washington, DC, 1977.

[126] N. Tomita, Y. Y. Cheung, and S. Hassanpour. Deep neural networks for automatic detection of osteoporotic vertebral fractures on CT scans. *Computers in biology and medicine*, 98(-):8–15, 2018.

[127] R. Villegas, O. Dorn, M. Kindelan, and M. Moscoso. Imaging low sensitivity regions in petroleum reservoirs using topological perturbations and level sets. *Journal of Inverse and Ill-posed Problems jiip*, 15(2):199–223, 2007.

[128] M. Wang. *Industrial tomography: systems and applications*. Elsevier, 2015.

[129] P. Wang, Z. Liu, X. Zhang, L. Xu, J. He, and Y. Wan. Wideband Signal Based Near-Field Electromagnetic Ranging for Indoor Localization. In *Advances in Intelligent Systems Research (AISR), 2018 International Conference on Advanced Control, Automation and Artificial Intelligence (ACAAI 2018)*, volume 155, pages 243–247, 2018.

[130] S. H. Ward and G. W. Hohmann. Electromagnetic theory for geophysical applications. In *Electromagnetic Methods in Applied Geophysics: Volume 1, Theory*, pages 130–311. Society of Exploration Geophysicists, 1988.

[131] G. Watson. A treatise on the theory of bessel functions. *Univ. Press, Cambridge*, 1944.

[132] T. Weiland. A discretization model for the solution of Maxwell's equations for six-component fields. *Archiv Elektronik und Uebertragungstechnik*, 31(-):116–120, 1977.

[133] J. Wood, R. Ward, C. Lloyd, P. Tatum, C. Shenton-Taylor, S. Taylor, G. Bagley, M. Joseph, and J. Watson. Effect of shielding conductivity on magnetic induction tomographic security imagery. *IEEE Transactions on Magnetics*, 53(4):1–6, 2017.

[134] X. Wu, R. He, Z. Sun, and T. Tan. A Light CNN for Deep Face Representation with Noisy Labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[135] K. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on antennas and propagation*, 14(3):302–307, 1966.

[136] K. S. Yee and J. S. Chen. The finite-difference time-domain (FDTD) and the finite-volume time-domain (FVTD) methods in solving Maxwell's equations. *IEEE Transactions on Antennas and Propagation*, 45(3):354–363, 1997.

[137] M. Zhdanov. *Inverse theory and applications in geophysics*. Cambridge Monographs on Applied and Computational Mathematics. Elsevier, 2015.

Index