BETTER IMAGING FOR LANDMINE DETECTION: AN EXPLORATION OF 3D FULL-WAVE INVERSION FOR GROUND-PENETRATING RADAR

Watson, Francis Maurice

2016

ISSN 1749-9097

BETTER IMAGING FOR LANDMINE DETECTION

AN EXPLORATION OF 3D FULL-WAVE INVERSION FOR GROUND-PENETRATING RADAR

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

2016

Francis Maurice Watson
School of Mathematics

Contents

Word count 78282

List of Tables

List of Figures

List of Algorithms

The University of Manchester

Francis Maurice Watson

Doctor of Philosophy

Better Imaging for Landmine Detection: An exploration of 3D full-wave inversion for ground-penetrating radar

January 1, 2016

Humanitarian clearance of minefields is most often carried out by hand, conventionally using a a metal detector and a probe. Detection is a very slow process, as every piece of detected metal must treated as if it were a landmine and carefully probed and excavated, while many of them are not. The process can be safely sped up by use of Ground-Penetrating Radar (GPR) to image the subsurface, to verify metal detection results and safely ignore any objects which could not possibly be a landmine.

In this thesis, we explore the possibility of using Full Wave Inversion (FWI) to improve GPR imaging for landmine detection. Posing the imaging task as FWI means solving the large-scale, non-linear and ill-posed optimisation problem of determining the physical parameters of the subsurface (such as electrical permittivity) which would best reproduce the data. This thesis begins by giving an overview of all the mathematical and implementational aspects of FWI, so as to provide an informative text for both mathematicians (perhaps already familiar with other inverse problems) wanting to contribute to the mine detection problem, as well as a wider engineering audience (perhaps already working on GPR or mine detection) interested in the mathematical study of inverse problems and FWI.

We present the first numerical 3D FWI results for GPR, and consider only surface measurements from small-scale arrays as these are suitable for our application. The FWI problem requires an accurate forward model to simulate GPR data, for which we use a hybrid finite-element boundary-integral solver utilising first order curl-conforming Nédélec (edge) elements. We present a novel 'line search' type algorithm which prioritises inversion of some target parameters in a region of interest (ROI), with the update outside of the area defined implicitly as a function of the target parameters. This is particularly applicable to the mine detection problem, in which we wish to know more about some detected metallic objects, but are not interested in the surrounding medium. We may need to resolve the surrounding area though, in order to account for the target being obscured and multiple scattering in a highly cluttered subsurface.

We focus particularly on spatial sensitivity of the inverse problem, using both a singular value decomposition to analyse the Jacobian matrix, as well as an asymptotic expansion involving polarization tensors describing the perturbation of electric field due to small objects. The latter allows us to extend the current theory of sensitivity in for acoustic FWI, based on the Born approximation, to better understand how polarization plays a role in the 3D electromagnetic inverse problem. Based on this asymptotic approximation, we derive a novel approximation to the diagonals of the Hessian matrix which can be used to pre-condition the GPR FWI problem.

Declaration

No portion of the work referred to in the 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 Statement

Acknowledgements

I am sincerely grateful to my supervisor, Prof. Bill Lionheart, for the many illuminating and broad discussions, and the support, guidance and encouragement to develop as a mathematician and follow my research ideas. I am also most thankful to my examiners, Dr. Paul Ledger and Dr. Oliver Dorn, for their useful, interesting and insightful comments which have helped to improve this thesis.

I would like to thank Find A Better Way for their support of this research, as well as to Sir Bobby Charlton (founder) for the motivation to help solve the landmine problem. The FABW sponsored SEMIS group, in particular Prof. Tony Peyton and Dr. Frank Podd, have also been immensely helpful.

I am indebted to my colleagues in the School of Mathematics and Inverse Problems Group for providing a stimulating atmosphere to work. In particular, to Michael Crabb, Russell Miller and Sophia Coban for the useful comments, discussions and insights throughout my studies. Such computational research would also not have been possible without the support of the University's IT Support team, not least Dr. Michael Croucher.

Many thanks to Prof. David Daniels, Prof. Motoyuki Sato, Utsi Electronics and the Mines Advisory Group for sharing their knowledge and experiences of landmine detection and GPR shared by these groups and individuals has helped to provide focus to this research. An additional thank-you to Vincent Utsi for the use of GPR equipment, and Dr. Alessio Balleri at Cranfield University for the use of their test bed.

A special thank you to my family and friends for their moral support. Especially to my mum, dad and sister for always encouraging me along, and my good friends Nat, Jameson and Mike for taking the time to proof read this thesis and letting me talk maths at them.

Dedicated to the loving memory of
Ralph John Francis Watson

Table 1: List of Abbreviations

ABC	Absorbing Boundary Condition
ALIS	Advanced Landmine Imaging System
BFGS	Broden Fletcher Goldfarb and Shanno optimization algorithm
CFIE	Combined Field Integral Equation
EFIE	Electric Field Integral Equation
EIT	Electrical Impedance Tomography
FE-BI	Finite Element Boundary Integral
FEM	Finite Element Method
FWI	Full-Wave Inversion
GMRES	Generalised Minimal Residual algorithm
GPR	Ground-Penetrating Radar
IED	Improvised Explosive Device
IFFT	Inverse Fast Fourier Transform
ILU	Incomplete LU decomposition
ILU-C	Crout variant of ILU
LU	Lower Upper decomposition $A = LU$
MD	Metal Detector
MFIE	Magnetic Field Integral Equation
MINRES	Minimal Residual algorithm
MLFMA	Multi-Level Fast Multipole Algorithm
NENH	$\hat{\mathbf{n}}\times$Electric $\hat{\mathbf{n}}\times$Magnetic (formulation of CFIE)
NETH	$\hat{\mathbf{n}}\times$ Electric Tangential Magnetic (formulation of CFIE)
PDE	Partial Differential Equation
PEC	Perfect Electric Conductor
PMC	Perfect Magnetic Conductor
PML	Perfectly Matched Layer
ROI	Region of Interest
RWG	Rao Wilton Glisson basis functions
SAR	Synthetic Aperture Radar
SCPML	Stretched Co-ordinate PML
SHF	Super High Frequency (radio band)
SR1	Symetric Rank-One optimization algorithm
SVD	Singular Value Decomposition
SVE	Singular Value Expansion
TENH	Tangential Electric $\hat{\mathbf{n}}\times$Magnetic (formulation of CFIE)
TETH	Tangential Electric Tangential Magnetic (formulation of CFIE)
TENENH	Tangential Electric $\hat{\mathbf{n}}\times$Electric $\hat{\mathbf{n}}\times$Magnetic (formulation of CFIE)
TNT	Trinitrotoluene (explosive)
UHF	Ultra High Frequency (radio band)
UPML	Uniaxial PML
UXO	Unexploded Ordinance
VHF	Very High Frequency (radio band)

Chapter 1

Introduction

1.1 The Landmine Problem

1.1.1 Legacy of Landmines

According to the Landmine and Cluster Munition Monitor there is currently no credible estimate for the number of landmines currently planted in the ground, though some sources put the estimate at 10 million [171]. A better indication of the impact of mines on people is the amount of contaminated area, and what sort of land this is. According to The Monitor's latest report, Afghanistan, Bosnia and Herzegovina, Cambodia and Turkey all have more than 100 km^2 contaminated land [174], and a further 18 countries are estimated to have contamination between 10 and 100 km^2. They state that in 2013 there were 3,308 reported casualties due to landmines, an average of 9 per day, the vast majority of which (79%) were civilians. Of civilian casualties, 46% were children. It is further estimated by the United Nations (www.un.org) that for every 5000 mines cleared, one worker will be killed and two injured by accidental explosions.

While 162 states have signed the Mine Ban Treaty, there are still areas in the world where new mines are being placed. It has been confirmed that between September 2013 and October 2014, there was new use of antipersonnel mines by government forces in Syria and Myanmar (states not party to the Mine Ban Treaty), as was there in the internationally unrecognised breakaway area of Nagorno-Karabakh. There have been unconfirmed allegations of landmine use in the conflict between the Ukranian government forces and Russian-backed separatists [174].

1.1.2 Landmines, UXOs and IEDs

The Ottawa Treaty, often referred to as the Mine Ban Treaty but formally the Convention on the Prohibition of the Use, Stockpiling, Production and Transfer of Anti-Personnel Mines and on their Destruction, defines a landmine to be "a munition designed to be placed under, on or near the ground surface area and to be exploded by the presence, proximity or contact of a person or a vehicle" [173]. Such devices generally fall under two categories: anti-personnel and anti-vehicle (or anti-tank) landmines. Anti-personnel mine is defined by the Mine Ban Treaty as "a mine designed to be exploded by the presence, proximity or contact of a person and that will incapacitate, injure or kill one or more persons." State Parties of the Mine Ban Treaty (currently 162 countries [174]) are obliged not to use, develop, produce, acquire, stockpile or transfer anti-personnel landmines, but anti-vehicle mines are not covered by the ban. While they will generally require much greater pressure to activate, they still pose a threat to civilians. Devices set off by remote detonation are not covered by the treaty.

There are four main component parts to a landmine: an outer casing of plastic, wood, rubber or glass; a fuse or other firing mechanism; a detonator; and high explosives. Anti-personnel mines are often roughly cylindrical in shape, ranging from around 5 – 10 cm in diameter, but can also be cuboid or shaped like a butterfly. They may contain large amounts of metal, designed to fire shrapnel out over large distances, or may have a minimal amount of metal and so be difficult to detect using a metal detector.

Homemade copies, or other homemade devices designed to detonate with the proximity or contact of a person or vehicle, are called Improvised Explosive Devices (IEDs). Unexploded Ordnance (UXOs) are explosive weapons such as bombs, rockets, missiles, mortars and grenades which did not explode when used. Both IEDs and UXOs pose a similar threat to civilians as anti-personnel mines: all are indiscriminate; they prevent land from being used for farming, houses, schools and roads; and they kill or seriously injure people trying to rebuild their lives for decades after a conflict.

1.1.3 Humanitarian mine clearance

Most landmines are activated by pressure, and so their safe detection and removal poses a significant risk. The terrain and local infrastructure may also make some detection or clearance methods more difficult. Anything which requires large amounts of power requires good enough local infrastructure to supply generators with fuel, and rough terrain can make vehicle mounted clearance methods impossible. Minefields, of course, exist where this is the case, and so we must have at our disposal a range of detection methods small and light enough to be transported by hand or pack animal. It is possible for hand-held devices to have some remote component, such as computers or power generation, provided they are also small enough for transportation.

Humanitarian demining also has specific concerns which may not be relevant for military demining. For humanitarian purposes, an area must be completely cleared of all landmines in order for that area to be once again used, e.g. for agriculture. Military demining may not have the same concern though, as it may only be necessary to create a safe path through a known minefield. If this is the objective one can use a mine flail to clear a path, which effectively crushes, sets off or moves mines in the path out of the way. For humanitarian purposes, this only counts as preparing the ground prior to clearance, as it cannot be guaranteed that all the mines have been destroyed, and explosives will still be present.

1.1.4 Detection and classification tools

Given an area known to contain low metal content landmines, the mine clearance process involves finding an object which may be a landmine, determining if the object is or is not a landmine, followed by the safe removal of the object. In order to meet the United Nations and International Mine Action Standards are met, contaminated areas will generally be cleared manually, which is an expensive, slow and dangerous process. The most common detection method used by mine clearing personnel is a metal detector (MD), and every piece of detected metal is carefully probed to determine if a mine is present. Metal detectors must be sensitive enough to detect minimum metal mines (which may have as little as 1 gram of metal present), and so yield approximately 1000 false positives for every mine detected [172]. Metal free mines cannot be detected in this way, but these are rare. The need to determine what every piece of detected metal is, and the large number of false positives, contributes significantly to the amount of time it takes to clear minefields.

The other tools commonly at the disposal of mine clearance personnel are well-trained dogs (or other trained animals such as rats), which are able to smell the explosive chemicals such as TNT in landmines, and ground penetrating radar (GPR). Other ongoing developments for mine-detection include explosives detection with Nuclear Quadrupole Resonance, infra-red methods, acoustics or ultrasound, and electrical impedance tomography [106].

There are currently two field-deployed hand-held combined GPR and metal detector landmine detection systems commonly available: MINEHOUND [49, pp 636] and HSTAMIDS (Handheld STAnd-off MIne Detection System) [49, pp 665-667]. The ALIS (Advanced Landmine Imaging System) [58,145,146] has also undergone successful field trials, but is not yet in production. MINEHOUND presents 'raw' GPR data to the user audibly as a tone which varies in amplitude due to reflection amplitude, and pitch due to depth of target. HSTAMIDS similarly presents the data audibly, but this is after data has been processed (the processing algorithms used are not known to have been published), whereas ALIS presents the user with a SAR image of the subsurface. As well as these already fielded systems, much consideration has been given to appropriate imaging and signal processing methods for GPR landmine detection. For example, using migration or synthetic-aperture imaging [58,145,146,156,162,164], or pattern recognition and signal filtering methods [20,101,144,147,165,178,179,198]

The dual sensor ALIS has been able to achieve over 50% reduction in false alarm rate at test lane trials in Croatia [60, pp 39], and varying amounts up to 36% in Cambodia [60, pp37]. During MINEHOUND tials in Cambodia between August 2010 and December 2013, 845 landmines were correctly identified in an area of 573,109 m^2, while crucially 92% of metal objects were correctly rejected as clutter [43]. 99% of metal targets were able to be rejected in an anti-tank minefield in Afghanistan, since these targets are easier to discriminate. Trials of the HSTAMIDS system in Thailand also had a good clutter rejection of 77% compared with metal detection alone [50].

It is clear that one detection modality and equipment may never be sufficient for all locations landmines are found, and for all types of landmine. For example, metal detectors will never work on metal free landmines, GPR works poorly in highly conductive soils, and dogs and other animals cannot work in highly contaminated areas and, being animals and not machines, may make mistakes or simply not wish to work. Some detection methods may also work well together as a primary detection and secondary classification modality. For example, the case of a metal detector to find a target, and GPR to determine the shape and material of the target.

1.1.5 Demining procedure for combined metal-detection and GPR

When using the MINEHOUND, the demining procedure developed by the Halo Trust for humanitarian clearance is summarised as follows [42]:

1. Mark out a lane from the side of the minefield.
2. Walking along the safe (previously cleared) side of the lane, with MINEHOUND in metal detection mode mark every piece of metal with a red tag.
3. Walking along the safe side of the lane, with MINEHOUND in GPR mode explore each marked target for GPR signal.
4. Replace the red tag over metal targets which had no GPR signal with a blue tag.
5. Carefully excavate all red targets, on the assumption they are landmines.
6. Flail all remaining metal only targets, destroying them so they are not re-detected
7. Move the demined line forwards, marking out a new lane, and return to 2.

Note that in this procedure, deminers are working from one side of a field across, a small distance in at a time, as opposed to individual lanes taking deminers a large distance into the field. The excavation (and destroying of mines) portion of the procedure is done to international standards, and is the same as if a metal detector alone had been used: the GPR simply allows some metal targets to be ignored.

It is preferred that any new equipment could easily be incorporated into a similar procedure as above, so that deminers can continue to work in a way with which they are familiar with (removing some need for retraining), and that has been shown to work. More advanced GPR imaging could easily be incorporated into this procedure, simply by adding an additional stage in which those metal targets which *did* have a GPR signal are also imaged, enabling more to be rejected as clutter. In this thesis we are considering the use of Full-Wave Inversion (FWI), which is a computationally expensive method. The computational expense means it may be some time before FWI can be carried out in real-time on field portable computers. This should not deter us, as improvements in computer power will be naturally driven by other applications, and landmines will continue to be an issue for a long time to come.

It is also possible that the computation could be carried out in the naturally large amount of time between the MD and GPR scan. GPR data could be collected simultaneously to MD data to be processed (only if a return was detected) while the deminer continues up the lane (transmitted to a nearby computer if necessary). A system such as a barcode on the red tags could identify the processed images with detected targets, which could be aligned with metal detection data to determine an origin. For a simple safety check, the GPR data taken when the deminer explores the target in GPR mode (the second pass) could be compared with that taken in the background in MD mode (the first pass), validating the pre-processed image and ensuring the data does correspond to the given ‘GPR code’. Deminers should of course be given both the raw and processed data, to let them make the decision of mine/not a mine. The practicality of such a method could only be determined by field testing, and it may of course be preferable to wait a little longer for more powerful portable computers.

1.1.6 Some specific challenges for GPR landmine detection

Unlike in metal detection, the electromagnetic field emitted by a GPR is greatly affected by soil characteristics and clutter in the ground. Firstly, any landmine detection system (and imaging method) will need to cope with the wide range of soil types in which landmines are found: not only a wide range of permittivities, which primarily will effect the wavelength in the ground (and therefore image resolution) as well as the ground surface reflection, but also a wide range of conductivities, which will affect signal attenuation. Both will affect the contrast of detected targets to the background.

GPR is also greatly affected by inhomogeneities in the soil, whether they are clutter objects, for example pieces of waste tin can, or local changes in the soil itself such as patches of clay, layers or stones. These inhomogeneities cause additional reflections and change the ray-path of the electromagnetic wave, making it harder to image targets in the subsurface.

Some specific challenges to GPR landmine detection occur because of clutter in the ground. For example, landmines may often be placed under trees because people will shelter there from the midday sun. In some places, these trees have been grown in tin pots, and the tree planted out still in the pot. As the tree grows the pot is broken and pieces of tin are spread out in the soil, creating a highly cluttered environment in which it can be very hard to detect and classify a landmine amongst the roots, pieces of tin and soil [65]. The soil itself can often confuse the results of a GPR for de-miners. For example, patches of clay soil, water or air in the ground can give a similar response to a landmine, leading de-miners to carefully excavate an area to find nothing present [42]. Landmines are found in both rural and (semi) urban environments, in which the type of clutter and ground will be vastly different. Examples such as these highlight why imaging methods for mine detection must be able to cope with an inhomogeneous subsurface, and may need to do more than determine the shape of an object to identify it.

Not only should imaging methods be able to cope with these difficulties, but they must also *improve* on the current status quo. That is, we must have a 100% detection rate, and to reduce false positives by more than 90%. A new GPR and imaging method is only an improvement then if, of the false positive detections of a metal detection system, at most 1 in 100 false positives are not rejected – we should aim for a 99% reduction. Of course, performance at the same level as systems currently available but either being more affordable would also be an improvement, but not the area this thesis will concentrate on.

1.2 Thesis overview

1.2.1 Aims and objectives

This thesis is concerned with the use of GPR imaging algorithms to classify detected objects, to allow de-miners to safely ignore targets which cannot be part of a landmine (reducing the rate of false positives), thus safely speeding the detection process. Such algorithms must be capable of operating in the highly cluttered and wide-ranging environment in which landmines are found, as well as able to obtain more information useful in reducing the rate of false positives. Specifically, we will explore the possibility of using Full-Wave Inversion methods (FWI), in which quantitative information is gained about detected targets and multiple scattering is solved for as part of the imaging algorithm. The aims of this thesis are then as follows:

1. Present the state of the art, theory and implementational aspects of Full-Wave Inversion for GPR in a manner suitable for the general scientific audience;

2. Develop shallow subsurface FWI algorithms for hand-held GPR data, which are capable of imaging a highly cluttered environment and give more information about detected (metalic) objects to help reduce the rate of false positives;

3. Better understand spatial sensitivity of the 3D electromagnetic imaging problem, specifically with regards to how the number, layout and polarization of GPR antennas in a hand-held device affects the ability to image the subsurface and to resolve or destinguish different targets.

While this thesis primarily presents numerical experiments into FWI, laying the groundwork for future development using the method for mine detection, we also give all the relevant background material and numerical recipes, so that readers have all the information available in one location for further study in FWI for mine detection. Our hope is that the thesis provides an informative text both for mathematicians (perhaps familiar with other inverse problems) wanting to contribute to the mine detection problem, as well as a wider engineering audience (perhaps already working in GPR and/or mine detection) interested in the mathematical study of inverse problems and FWI.

1.2.2 Layout of thesis

We begin in Chapter 2 with an introduction to inverse problems theory and ground-penetrating radar hardware and imaging, as well a review of the current state of full-wave inversion and an overview of how this optimisation problem can be solved. This background material is supplemented in Chapter 3 by a presentation of the fundamental equations governing electromagnetic wave scattering – Maxwell's equations.

Chapter 4 explores FWI in 2D, both with a numerical experiment into reconstructing fairly complex targets but also explaining in some depth all the aspects of the inversion procedure. We expect this chapter to be useful primarily to those unfamiliar with either FWI or non-linear inverse problems, and others may wish to read ahead to the more novel content of the thesis. We use a finite-difference approximation of the Helmholtz equation as the forward model (simulating GPR data), and apply appropriate Total Variation regularisation to the inverse problem. As we are well able to reconstruct targets in the shallow subsurface with data simulated for small hand-held arrays, in Chapter 5 we explore more formally what is required of an acquisition system for FWI (as well as improving mine detection) by way of an SVD analysis. Such analysis is often used in inverse problems, and while it has been applied to FWI to understand the conditioning of the problem, we go further by using the singular vectors to explore the null and image spaces.

A 2D approximation is not appropriate for landmine detection, as landmines are distinctly 3D objects and detection is hampered by out of plane (possibly multiple) scattering. Performing FWI in 3D requires an accurate 3D forward model, simulating the more complex scattering problem and incorporating polarization effects as well as source models and ground transmission/reflection. In Chapter 6 we present a combined finite-element boundary-integral (FE-BI) solver, in which the boundary integral portion describes the ground transmission/reflection of electromagnetic waves and allows any incident source field to be applied, giving a novel way to simulate GPR data. Simulations are compared with GPR field data, and a qualitative comparison gives promising results.

Chapter 7 is concerned with sensitivity of the electromagnetic inverse problem in 3D. We present an asymptotic approximation of the scattering of electromagnetic waves by a small object from the literature, using polarization tensors, and show that it can be written in a more familiar and useful form for GPR. This approximation is used to understand spatial sensitivity, and using it we derive a novel expression for the wavenumber coverage in the image domain, extending literature results which apply to scalar valued waves. This gives a novel understanding as to how cross-polar measurements can improve resolution, by adding an additional class of test functions for the inverse problem. We highlight this idea of sensitivity using the polarization tensor approximation via some numerical experiments into the performance of given acquisition arrays in distinguishing mine-like targets.

While FWI for GPR is well studied in 2D, as are iterative and non-linear 3D inverse electromagnetic and wave scattering problems, we present in Chapter 8 the first numerical experiments into 3D FWI for near-surface GPR data recorded on small hand-held arrays. We find we are easily able to reconstruct isolated objects, and then give a novel algorithm to improve reconstruction of a target in a known location surrounded by a highly inhomogeneous medium. While we are restricted by computational cost as to the size of problem we are able to solve, this chapter provides the basis of for future development of 3D FWI for landmine detection. We discuss how the method can give certainty that detected targets are not landmines, and what would need to be done for this to be realised.

Finally, in Chapter 9 we derive a novel approximation to terms in the Hessian matrix which can be used to precondition the l-BFGS optimisation scheme. This extends theory used for preconditioning FWI of scalar-valued waves, but also approximates the change in components due to their contrast to the surrounding medium. We use the ability to estimate the change in Hessian components in this approximation in a novel trust-region model, enabling us to efficiently determine an appropriate first step length in the optimisation procedure. This is particularly useful as quasi-Newton algorithms such as l-BFGS often require many cost function evaluations in the first iteration to find an appropriate step length, due to a poor initial Hessian approximation. We compare the pre-conditioned to l-BFGS to l-BFGS initialised with the identity matrix in some numerical experiments, to help to understand its effectiveness. While the Hessian is derived with an expression appropriate for FWI of electromagnetic waves, the same principle could be used for any modality which has an asymptotic expansion involving polarization tensors.

1.2.3 Outputs arising from this thesis

The following papers are in preparation as a result of this research:

- A. Adler, WRB. Lionheart, and F. Watson. Shape sensitivity of EIT and GPR. Working title

- Francis Watson. A full-wave inversion strategy for surface GPR measurements to determine 3D targets in an area of interest. Working title

- Francis Watson. A novel preconditioner for the GPR full-wave inversion problem in 3D

The following paper was presented at GPR 2014:

- F Watson and WRB Lionheart. SVD analysis of GPR full-wave inversion. In *Ground Penetrating Radar (GPR), 2014 5th International Conference on*, pages 484–490, 2014

It has also resulted in an open access suite of Matlab functions to solve GPR FWI problems [194].

Contributed talks were given at the BAMC 2014 and 2015; GPR 2014; the Medical Imaging Workshop at the University of Strathclyde, Glasgow, May 2014; and the Manchester Student SIAM Chapter Conference 2015. An invited seminar was given at Hull University in May 2015, and poster presentations given at the Parliamentary SET for Britain Exhibition 2014 and the BAMC 2013 (the latter being awarded the Best Student Poster prize).

Chapter 2

Full-Wave Inversion of GPR Data

2.1 Introduction

This chapter sets out the basic theory needed throughout the thesis. We begin by introducing to the reader to the concept and basic theory of inverse problems. An inverse problem is, given some measurements, determine what caused them – clearly GPR imaging is an inverse problem. We then discuss the basics of GPR theory, covering the current (commercial) industry standard of both equipment and data processing and imaging. In Section 2.4 we discuss the GPR imaging problem in the context of related electromagnetic inverse scattering problems, before reviewing the non-linear imaging method this thesis concentrates on, namely Full Wave Inversion.

Readers with a background in mathematical inverse problems may wish to refer to only Section 2.3, covering the basics of GPR theory to better understand the problem at hand. Posing the imaging problem as a (nonlinear) regularised least-squares problem will likely be familiar to members of the inverse problems community, but the current literature is reviewed in Section 2.5 if they are interested in the current state of GPR FWI as compared to other nonlinear inverse problems.

Equally, those with a background in GPR will likely find Section 2.3 quite basic, but be more interested in the mathematical perspective of imaging in general, Section 2.2, as well as the overview of the state of FWI. As well as reviewing the literature on GPR FWI, Section 2.5 also covers the basics of how posing the imaging problem as an optimisation one can be solved, giving features such as the gradient and Hessian of the data misfit functional's physical meaning.

2.2 Inverse and ill-posed problems

2.2.1 Forward versus inverse problems

We follow Mueller and Siltanen [121] to overview the basic concepts and features of a non-linear inverse problem, which is defined as the opposite of a direct, or forward, problem. That is, a direct problem is one in which we find an effect from a known cause, and an inverse problem is one in which we try to determine the cause of an observed effect. Given a map

$$\mathcal{F} : X \to Y \qquad \mathcal{F} : m \mapsto d,$$

the forward problem is to calculate the observable quantity $d = \mathcal{F}(m)$, and the inverse problem is to determine some model parameter(s) or function(s) m which caused the observed quantity d. Without loss of generality, we may consider $X \subset \mathbb{R}^p$ and $Y \subset \mathbb{R}^q$. For example, in X-ray tomography, the forward problem is to simulate some X-ray data which would be observed from a known object. The inverse problem is to determine what object gave rise to the recorded data. Landmine detection is of course an inverse problem, by whatever modality we use to detect them.

In the mathematical study of inverse problems, one is generally concerned with ill-posed problems. A well-posed problem $f : X \to Y$, as defined by Hadamard [67, pp. 4952], is one which has

1. Existence: there is at least one solution to $f(x)$, $x \in \mathcal{D}(f) \subset X$, the domain of f;
2. Uniqueness: given $m \in \mathcal{D}(f)$, there is at most one solution to $f(x)$;
3. Stability: the solution $y = f(x)$ varies continuously with the input x.

An ill-posed problem is one which fails one or more of these conditions. Necessarily, the direct problem $\mathcal{F}$ is well posed: $\mathcal{F}$ is a well-defined, single valued continuous function. However, $\mathcal{F}^{-1}$ may not exist, and if it does it may fail at least one of Hadamard's conditions. For example, the deblurring of a noisy, out of focus or deteriorated photograph to restore the original is an ill-posed problem: a solution exists, but we can likely find multiple possible originals as information has been lost in the blurring. However, restoring a photograph from its negative is well posed: the map between photograph and negative is bijective. The former is interesting to us mathematically, whereas the latter is a trivial process.

For practical problems, one can only have indirect measurements which contain noise,

$$d = \mathcal{F}(m) + \varepsilon. \tag{2.2.1}$$

We may have statistical information on the noise ε, such as its distribution or standard deviation, or we may think of it as a deterministic but unknown quantity. In either case, we assume it is bounded by $\|\varepsilon\| < \delta$, for some $\delta > 0$. The inverse problem is now:

> Given some measurements which contain noise, $d = \mathcal{F}(m) + \varepsilon$, find the parameters m which gave rise to the observed data.

The naive solution to the inverse problem is to apply $\mathcal{F}^{-1}$ to d, or if the operator does not exist we can define it in the least-norm sense

$$\mathcal{F}^{-1}(d) := \operatorname*{argmin}_{m \in X} \|\mathcal{F}(m) - d\|. \tag{2.2.2}$$

Since we are considering ill-posed inverse problems, one of Hadamard's conditions fails for $\mathcal{F}^{-1}$. If 1. is violated, $d = \mathcal{F}(m) + \varepsilon$ does not lie in the range of $\mathcal{F}$, $d \notin \mathcal{F}(\mathcal{D}(\mathcal{F}))$. Criteria 2. fails if $\mathcal{F}(m_1) = \mathcal{F}(m_2)$ for some $m_1 \neq m_2 \in X$, and 3. is violated if $\mathcal{F}$ does not allow a continuous inverse.

When one of Hadamard's conditions fail, the naive inversion $\mathcal{F}^{-1}$ may give a solution far from the true model parameters. It may be an non-physical solution which does not lie in $\mathcal{D}(\mathcal{F})$, and the contribution from noise ε may dominate the solution. Often, the contribution from noise will be highly oscillatory.

2.2.2 Regularised inversion

Since the naive solution to the inverse problem will fail, we introduce the restricted inverse problem,

> Let $d = \mathcal{F}(m) + \varepsilon$. Given d and $\delta > 0$ with $\|d - \mathcal{F}(m)\| \leq \delta$, extract any information about m.

This information could include the number and location of inclusions, the shape and size of an object, or a simple binary decision such as landmine or not a landmine.

To solve this problem, we introduce the regularised inverse operator $\mathcal{R}_\lambda : Y \to X$, which depends on the regularisation parameter $0 < \lambda < \infty$. For Hilbert spaces X and Y, and $\mathcal{F} : X \to Y$ an injective bounded linear operator, the family of maps $\mathcal{R}_\lambda$ is called a *regularisation strategy* if

$$\lim_{\lambda \to 0} \mathcal{R}_\lambda \mathcal{F}(m) = m \qquad \forall m \in X. \tag{2.2.3}$$

Given a noise level $\delta > 0$, so that $\|d - \mathcal{F}(m)\| \leq \delta$, a choice of regularisation parameter $\lambda(\delta)$ is called *admissible* if

$$\begin{aligned} &\lambda(\delta) \to 0 \text{ as } \delta \to 0 \text{ and} \\ &\sup_{d} \left\{ \|\mathcal{R}_{\lambda(\delta)} d - m\| \ : \ \|\mathcal{F}(m) - d\| \leq \delta \right\} \to 0 \text{ as } \delta \to 0 \ \forall m \in X. \end{aligned} \tag{2.2.4}$$

A regularisation strategy then is one which recovers the true parameters m if there is no noise and $\lambda = 0$, and if the regularisation parameter is admissible we have a solution which does not blow up with noise: the regularised inverse operator $\mathcal{R}_\lambda$ is robust to noise.

To solve an inverse problem, typically we must discretise the direct problem in order to compute solutions numerically. The model parameters m are approximated by some vector $\mathbf{m}$, where either $m_i \approx m(\mathbf{x}_i)$, the value of m at some discrete points, or more generally m_i are the coefficients of a set of functions $\phi_i(\mathbf{x})$ such that $m \approx \sum_i m_i \phi_i$. Equally, the data is simulated at a discrete set of points $\mathbf{d} = \mathcal{F}(\mathbf{m})$. We have abused the notation $\mathcal{F}(\mathbf{m})$ to imply the numerical solution to the direct problem. Throughout this thesis, we have used bold-face to denote vectors of arbitrary dimension (e.g. $\mathbf{m}$), and capital letters in standard maths-italics for matrices (e.g. L).

Note that our choice of discretisation is in itself a form of regularisation: we are restricting the solution to the inverse problem to lie in our discrete function space. The numerical solution of the direct problem is in general a non-trivial part of the inverse problem. If the inverse problem is sensitive to noise, it will also be sensitive to numerical errors in the solution to the discretised direct problem.

A classical choice of regularisation strategy, and a common choice for linear inverse problems, is Tikhonov regularisation T_λ,

$$T_\lambda(\mathbf{d}) := \operatorname*{argmin}_{\mathbf{m}} \left\{ \|\mathcal{F}(\mathbf{m}) - \mathbf{d}\|_2^2 + \lambda \|\mathbf{m}\|_2^2 \right\}. \tag{2.2.5}$$

This can be understood as balancing two requirements:

1. T_λ gives a small residual $\mathcal{F}(T_\lambda) - \mathbf{d}$;

2. T_λ should be small in the L^2-norm.

The regularisation parameter λ then tunes these requirements. There may typically be many choices of T_λ satisfying 1, so one of the roles of 2. is to ensure the solution is unique.

We may have prior knowledge of the solution to the inverse problem, such as it may lie close to some $\mathbf{m}^*$. In this case, we can generalise Tikhonov regularisation to

$$T_\lambda(\mathbf{d}) := \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \|\mathcal{F}(\mathbf{m}) - \mathbf{d}\|_2^2 + \lambda \|\mathbf{m} - \mathbf{m}^*\|_2^2 \right\}. \tag{2.2.6}$$

If the noise in the data is Gaussian white noise, the Tikhonov regularised solution is equivalent to the maximum *a posteriori* (MAP) estimate of $\mathbf{m}$ given data $\mathbf{d}$ and noise variance η in the Bayesian interpretation of the inverse problem, where in the Bayesian case λ is related to the variance of the noise (see for example [25, pp. 183-188]).

Tikhonov regularisation can be generalised further as

$$T_\lambda(\mathbf{d}) := \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \|\mathcal{F}(\mathbf{m}) - \mathbf{d}\|_P^2 + \lambda \|\mathbf{m} - \mathbf{m}^*\|_Q^2 \right\}, \tag{2.2.7}$$

where $\|\mathbf{x}\|_Q^2 \equiv \mathbf{x}^T Q \mathbf{x}$ is a weighted l_2 norm. In the Bayesian interpretation, the matrix P is the inverse covariance matrix of $\mathbf{d}$, while Q is the inverse covariance matrix of $\mathbf{m}$. For example, if we know the solution is smooth we may choose for a regularisation strategy

$$T_\lambda(\mathbf{d}) := \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \|\mathcal{F}(\mathbf{m}) - \mathbf{d}\|_2^2 + \lambda \|L\mathbf{m}\|_2^2 \right\}. \tag{2.2.8}$$

or

$$T_\lambda(\mathbf{d}) := \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \|\mathcal{F}(\mathbf{m}) - \mathbf{d}\|_2^2 + \lambda \|L(\mathbf{m} - \mathbf{m}^*)\|_2^2 \right\}, \tag{2.2.9}$$

where L is a discretised differential operator. In general, our regularisation operator will take the form

$$\mathcal{R}_\lambda(\mathbf{d}) := \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \|\mathcal{F}(\mathbf{m}) - \mathbf{d}\|_2^2 + \lambda R(\mathbf{m}) \right\}, \tag{2.2.10}$$

where R is a function penalising deviation from prior known information about the solution to the inverse problem. Often R is an l_p norm, and for $0 \leq p \leq 1$ such a regularisation strategy promotes a sparse solution to the inverse problem.

2.3 GPR theory

2.3.1 GPR data and equipment

Ground penetrating radar equipment consists of one or more transmitting antennae and one or more receiving antennae either on or above the ground surface or buried in boreholes; a processor; and a display. The transmitting antenna radiates an electromagnetic wave in the microwave band, and reflections of the wave caused by objects in the subsurface or changes in dielectric constants are recorded at the receiving antenna, as shown in Figure 2.1. The source signal may be amplitude, frequency or phase modulated; or a noise signal. Most GPR systems use an impulse signal, with the reflections being recorded by a sampling receiver in the time domain. Stepped frequency and frequency modulated systems, recording in the frequency domain, are becoming more popular as their dynamic range can be greater than time domain radar, though there is little commercially available [41, pp 13, 36].

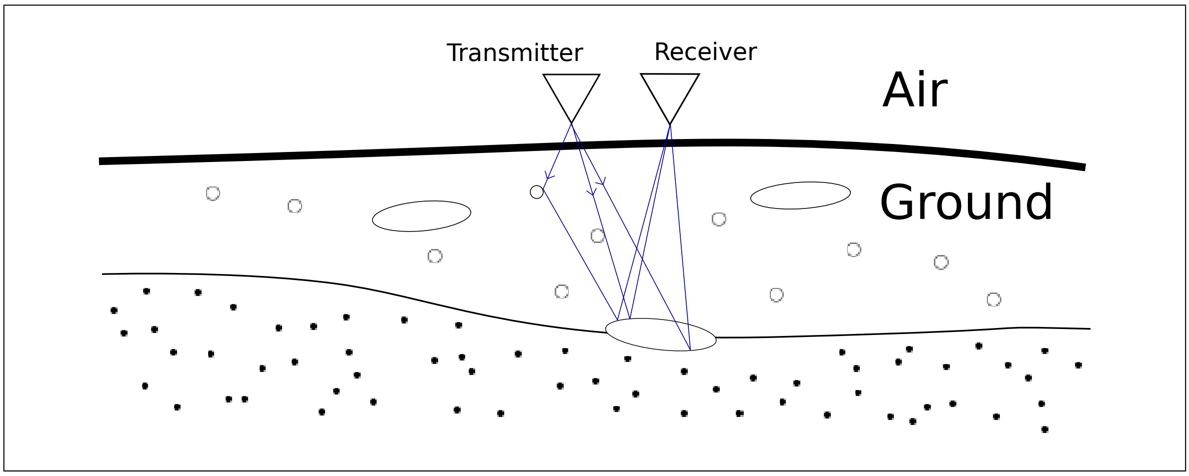


Figure 2.1: Schematic of GPR system and susbsurface, showing some of the ray paths between the transmittter and receiver.

The majority of commercially available GPR systems are either monostatic (using either a pair of co-located antenna or a single antenna which switches between transmitting and receiving) or are bistatic (using a single transmit and single receive antenna) pulsed systems.

2.3.2 GPR data processing and imaging

Traditional GPR data processing is largely focussed on signal processing, rather than inverse problems and imaging. We make the (rather informal) distinction that signal processing is largely to reduce clutter in the data, presenting it in a way which can more readily be interpreted by the end user; whereas inverse problems and imaging are the approach by which we find some set of parameters describing the subsurface which, when inserted in to the governing equations of electrodynamics, would best reproduce the data.

Let us denote the recording of the full set of (time domain) GPR data as the 3D matrix D, where element D_{ijk} is the GPR recording at receiving antenna position x_i, y_j and time t_k (assuming a single receiving antenna). Subsets of D are commonly referred to as A, B and C-scans, where an A-scan is a single recorded waveform, a B-scan is an ensemble of waveforms in one surface coordinate direction, and the C-scan is an ensemble of waveforms over both surface coordinate directions for a fixed time (or depth). That is, the A, B and C-scans are defined as

$$f_A(t_k) = D_{ijk} \qquad \text{for } i, j \text{ constant}, \;\; k \in 1:n \tag{2.3.1a}$$

$$f_B(x_i, t_k) = D_{ijk} \qquad \text{for } i \in 1:p, \;\; j \text{ constant } k \in 1:n, \tag{2.3.1b}$$

$$f_C(x_i, y_j) = D_{ijk} \qquad \text{for } i \in 1:p, \;\; j \in 1:q, \;\; k \text{ constant} \tag{2.3.1c}$$

In general, time t is interchanged with depth z by an approximation of the speed of wave propagation (or, more precisely, the distance from the antenna).

The received time waveform in an A-scan can be described as the convolution of functions, plus noise ε, as [45]

$$f_A(t) = f_s(t) \star f_{a_1}(t) \star f_c(t) \star f_g(t) \star f_t(t) \star f_g(t) \star f_{a_2}(t) + \varepsilon, \tag{2.3.2}$$

where f_s is the signal applied to the antenna, f_{a_n} are the antenna impulse responses, f_c is the antenna cross coupling response, f_g is the ground impulse response, and f_t is the response of subsurface targets. In this setting, A-scan processing can primarily be considered as determining the signal f_t due to the targets underground, and finding the parts of f_t relating to targets of interest. This largely consists of [41, pp 147-174]:

- zero-offset removal (ensuring the mean value of the A-scan is close to zero);

- noise reduction (such as averaging co-located samples);
- clutter reduction (such as averaging a group of A-scans over the area of interest);
- varying gain to account for amplitude loss with distance
- high-pass filtering to improve signal-to–clutter ratio, and low-pass to improve signal-to-noise ratio;
- wavelet optimisation resulting in the deconvolution of the wanted signal (e.g. application of the Weiner filter);
- determining the resonant frequency of characteristic targets;
- spectral analysis methods which avoid limitations of the fast Fourier transform when a frequency domain signal has been recorded, such as multiple signal classification (MUSIC).

We see that A-scan processing essentially involves pre-processing the GPR data.

As above, a B-scan can be considered an ensemble collection of (pre-processed) A-scans. This will exhibit hyperbolae-like structures caused by the reflection of waves from targets, due to the distance of the target to the antenna as it is moved being hyperbolic. B-scan processing is largely to determine target surfaces, either using migration methods or synthetic-aperture radar. Alternatively, B-scan processing can be considered an imaging processing problem, in which one carries out 2D convolution operations to highlight features in the data such as edges. Many of the processes applicable to B-scans are also suitable for C-scan processing.

2.4 Electromagnetic inverse problems and imaging

Ground penetrating radar is an inverse problem in which we wish to gain knowledge of the structure of, or objects in, the subsurface, from measurements of reflected electromagnetic waves. The electromagnetic waves emitted by the GPR system are primarily in the Ultra-High Frequency band (UHF, 300 MHz – 3GHz), although there are some systems which operate in the upper VHF (Very-High Frequency, 30 – 300

MHz) or lower SHF (Super-High Frequency, 3 – 30 GHz) region. This puts GPR in the middle of a range of electromagnetic inverse problems.

At the bottom end of the frequency scale is Electrical Impedance Tomography (EIT, see for example [27,175]), which operates from zero frequency up to the low kHz band. Both the objects being imaged with EIT and the domain are far smaller than the wavelength, and so one can think of the modality as determining the resistivity (reciprocal of conductivity) to the application of a direct current (i.e. an infinite wavelength), which is a highly ill-posed non-linear problem. At the other end of the spectrum lies X-ray tomography [52], where X-rays have a frequency above 300 PHz and a wavelength less than 1 nm. In X-ray tomography, one measures the absorption of X-ray beams as they pass through a body. The beams are generally considered not to scatter at all, and so the inverse problem is linear and only mildly ill-posed. Some small amount of scattering can occur, which if included in the forward model makes the inverse problem mildly nonlinear.

Between these lower and upper limits of the electromagnetic frequency scale are the modalities most similar to GPR: radio frequency imaging such as Radar; microwave tomography (GHz band); optical tomography [17] and Tomographic Diffractive Microscopy (e.g. [61,89]) (infra-red to visible light spectrum); and many other modalities and variants throughout the spectrum. In these examples, as with GPR, the objects being imaged are generally on the order of or larger than the wavelength, and so the dominant feature of the forward problem is scattering. The ill-posedness of the inverse electromagnetic scattering problem is between that of EIT and X-ray tomography, though in the low-frequency limit (i.e. for objects sufficiently small compared to the wavelength in the near field), it may be similar to to EIT.

We can use the Helmholtz equation to highlight the non-linearity in the inverse scattering problem. Consider first a domain Ω with homogeneous wave speed c_0, in which the wave field u_0 satisfies

$$\left(\nabla^2 + \frac{\omega^2}{c_0^2}\right) u_0(\omega, \mathbf{x}) = s(\omega, \mathbf{x}), \mathbf{x} \in \Omega. \tag{2.4.1}$$

For a domain with wave speed perturbed by some δc, the total field $u + \delta u$ satisfies

$$\left(\nabla^2 + \frac{\omega^2}{(c_0 + \delta c(\mathbf{x}))^2}\right) (u_0 + \delta u)(\omega, \mathbf{x}) = s(\omega, \mathbf{x}), \qquad \mathbf{x} \in \Omega \tag{2.4.2}$$

Subtracting (2.4.1) from (2.4.2), we find the perturbation in wave field δu owing to perturbation in wave speed satisfies

$$\left(\nabla^2 + \frac{\omega^2}{c_0^2}\right)\delta u = -\omega^2 V(\mathbf{x})(u_0 + \delta u), \tag{2.4.3}$$

where the reflectivity V is given by

$$V(\mathbf{x}) = \frac{1}{c_0^2} - \frac{1}{(c_0 + \delta c)^2}. \tag{2.4.4}$$

In integral form, the solution of equation (2.4.3) is given by

$$\delta u = \int G_0(\mathbf{x}, \mathbf{z}) V(\mathbf{z}) \omega^2 \big(u_0 + \delta u\big)(\omega, \mathbf{z})\ \mathrm{d}\mathbf{z}, \tag{2.4.5}$$

where G_0 is the freespace Green's function for the Helmholtz equation (detailed in Chapter 3). Equation (2.4.5) is referred to as the Lipmann-Schwinger equation [36, pp. 53].

In equation (2.4.3), the scattered wave δu appears both on the left hand side and as a source term, and in equation (2.4.5) it appears both within and outside of the integral. The wave scattered at $\mathbf{z}$ by the perturbation $\delta c(\mathbf{x})$ is re-scattered at $\mathbf{x}$ by the perturbation $\delta c(\mathbf{z})$. This is the source of the non-linearity of (δu) with δc, and therefore of the inverse problem. In Chapter 3, we see that the Helmholtz equation is a good approximation to the electromagnetic scattering problem under certain assumptions, but the non-linearity of the scattering problem holds in the full electromagnetic scattering case.

The Lipmann-Schwinger equation can be expanded as a Neumann series

$$\begin{aligned}\delta u = &\int G_0(\mathbf{x}, \mathbf{z}) V(\mathbf{z}) \omega^2 u_0(\omega, \mathbf{z})\ \mathrm{d}\mathbf{z} \\ &+ \int G_0(\mathbf{x}, \mathbf{z}') V(\mathbf{z}') \omega^2 \int G_0(\mathbf{z}', \mathbf{z}) V(\mathbf{z}) \omega^2 u_0(\omega, \mathbf{z})\ \mathrm{d}\mathbf{z}\ \mathrm{d}\mathbf{z}' \\ &+ \int G_0(\mathbf{x}, \mathbf{z}'') V(\mathbf{z}'') \omega^2 \int G_0(\mathbf{z}'', \mathbf{z}') V(\mathbf{z}') \omega^2 \dots \\ &\quad \int G_0(\mathbf{z}', \mathbf{z}) V(\mathbf{z}) \omega^2 u_0(\omega, \mathbf{z})\ \mathrm{d}\mathbf{z}\ \mathrm{d}\mathbf{z}'\ \mathrm{d}\mathbf{z}'' + \dots,\end{aligned} \tag{2.4.6}$$

which can be shown to converge provided the scattered fields are sufficiently small [36, pp. 54]. The series (2.4.6) makes the multiple scattering mentioned above clear. The linear approximation of (2.4.5) in δc is given by truncating the Neuman series to first order,

$$\delta u = \int G_0(\mathbf{x}, \mathbf{z}) V(\mathbf{z}) \omega^2 u_0(\omega, \mathbf{z})\ \mathrm{d}\mathbf{z} + \mathcal{O}(\delta c^2) := \delta u_{\mathrm{B}} + \mathcal{O}(\delta c^2), \tag{2.4.7}$$

where δu_{B} is called the Born approximation. We see that (2.4.7) is a single scattering approximation, with the error term $\mathcal{O}(\delta c^2)$ incorporating all multiple scattering.

Denote by $\mathcal{F}_{\mathrm{B}}$ the map $\mathcal{F}_{\mathrm{B}} : \delta c \mapsto \delta u_{\mathrm{B}}$ the linear operator (or Born operator). In the linear approximation, closed-form solutions to the inverse electromagnetic scattering problem of finding can be made to determine some (approximation to) $\mathcal{F}_{\mathrm{B}}^{-1}$. For example, Somersalo et al obtain an approximate solution in the form of an inverse Fourier transform of the data [160]. They make the assumption that the wave fields are in the form of distorted plane waves, for which they give well bounded estimates of the second order error term in the perturbation of permittivity, permeability and conductivity. Closed-form solutions can also be posed as an inverse generalized Radon transform, for example [26,47,48,117]. The use of an inverse Radon transform to solve the inverse scattering problem not only relates the inverse scattering problem to X-ray CT [52], but a much more general class of geometrical inverse problems, for example as discussed by Palamodov [127–129]. For time domain data, by Miller et al [117], we have

$$\tilde{V}(\mathbf{x}) = \frac{1}{\pi^2} \int \frac{|\cos^3 \alpha(\mathbf{r}, \mathbf{x}, \mathbf{s})|}{c_0^3 A(\mathbf{r}, \mathbf{x}_0, \mathbf{x})} \delta u(\mathbf{r}, \mathbf{s}, t = \tau_0) \, \mathrm{d}^2 \boldsymbol{\xi}(\mathbf{r}, \mathbf{x}, \mathbf{s}). \tag{2.4.8}$$

Here, $\tilde{V}$ is an estimation of V, $\mathbf{r}$ and $\mathbf{s}$ are the locations of source and receiver, α is half the angle between incident and scattered rays at $\mathbf{x}$, A is the amplitude of the relevant Green's function (i.e. 2D or 3D), τ is half the the travel time from $\mathbf{s}$ to $\mathbf{x}$ then to $\mathbf{r}$, and $\boldsymbol{\xi}$ is a unit vector in the direction of travel-time gradient $\nabla_{\mathbf{x}} \tau(\mathbf{r}, \mathbf{x}_0, \mathbf{s})$, and α is half the angle between $\boldsymbol{\xi}$ and the ray path from source to observation point.

Many similar expressions to (2.4.8) have been used for the inverse scattering problem in seismology, including the well known Kirchhoff migration formula

$$P = \frac{1}{2\pi} \int \left[\frac{\cos \alpha}{c_0 r} \frac{\partial}{\partial t} u_0(\mathbf{s}, \mathbf{r}, r/c_0) \right] \mathrm{d}\mathbf{s}, \tag{2.4.9}$$

as applied to the GPR landmine detection problem by Feng and Sato [58]. In (2.4.9) P is a back-projection of the data, which we can understand as data residual terms being 'projected' to where reflections could have originated if we reversed time in the wave equation. This results in an 'image' of the subsurface, in that it will have greatest amplitude at the locations in which reflection occurred due to constructive interference.

Such methods will break down when either there is a poor estimation of the wavespeed or multiple scattering is prevalent, and give rise to coherent artefacts in the image. With a poor estimation of wavespeed, the perturbations in wave speed found by expressions such as (2.4.8) and (2.4.9) will be wrongly placed in the subsurface, and data due to multiple scattering will be resolved as if it were due to additional scattering surfaces – seen as extra layers beneath those physically present. They also give only qualitative rather than quantitative information, in that they locate discontinuities of wavespeed in the subsurface (which give rise to reflections), and therefore show the shape of the objects, rather than determine what the wavespeed (or other physical parameters) is as a function of space.

In the landmine detection problem we are often faced with a highly inhomogeneous medium, with locally varying ground conditions and many clutter objects. We would also like to gain quantitative information about detected objects, as an additional measure to decide if a target could be a landmine, and so would like to solve the full non-linear electromagnetic inverse problem. Generally, there are no closed form solutions to fully nonlinear electromagnetic inverse problems (one notable exception is the D-bar method for EIT, see e.g. [87]), and so we must attempt to solve the regularised inverse problem (2.2.10) numerically with some appropriate optimisation scheme. All of the aforementioned electromagnetic inverse problems can (or in some cases must) be treated this way. For GPR reflection data, posing the inverse problem as the non-linear regularised least-squares problem such as (2.2.10) is referred to as Full-Waveform Inversion, which we review in the subsequent section.

2.5 Improving GPR imaging with full-wave inversion

2.5.1 Introduction

Full-waveform inversion (FWI) is an imaging approach in which we find the quantitative subsurface parameters (such as the dielectric permittivity) which would best fit the recorded GPR data. The solution of the FWI problem requires many numerical solutions of Maxwell's equations for comparison with the recorded radar data, and so has a very high computational cost. Indeed, solving this non-linear optimisation problem requires at least one solution of Maxwell's equations per iteration, per frequency and per source location to calculate the current cost of the objective function, with more to perform a line search and calculate a gradient. Because of this computational cost, much of the research into FWI has been restricted to 2D inversions.

Despite the computational cost, there are several benefits in taking a FWI approach to GPR imaging, the most obvious of which being gaining information about material properties. Unlike the more common direct imaging approaches, which are able to determine qualitative information about detected objects, by finding the set of subsurface parameters which would best reproduce the data we gain this as quantitative information – as well as the qualitative information seen by the boundaries between material types. We may even be able to determine how well resolved this quantitative information is [111].

Further, by taking a full-waveform approach, our inversion naturally includes non-linear effects in the data such as multiple scattering and diffraction. Contrarily, in a direct linear inversion, any observed non-linear effects will not match any single scattering and background wavespeed assumptions made, and so will cause artifacts in the image. The data is often filtered before direct inversions to attempt to remove these non-linear parts of the data and so reduce the resulting artefacts, but this has the effect of reducing the information content of the data.

2.5.2 An overview of full-wave inversion

Given a recorded GPR data set $\mathbf{d}_{\text{obs}}$, which will contain some unknown noise, and a model of simulating data $\mathcal{F} : \mathbf{m} \mapsto \mathbf{d}_{\text{sim}}$ for a given set of parameters $\mathbf{m}$ describing the electromagnetic properties of the subsurface (in some discretisation), the full-waveform problem is to find the $\mathbf{m}$ which solves

$$\mathbf{m}_{\text{inv}} = \operatorname*{argmin}_{\mathbf{m}\in \mathrm{M}} \mathcal{J}(\mathbf{m}) + \mathcal{R}(\mathbf{m}) = \operatorname*{argmin}_{\mathbf{m}\in \mathrm{M}} \frac{1}{2}\|\mathcal{F}[\mathbf{m}] - \mathbf{d}_{\text{obs}}\|_p^2 + \mathcal{R}(\mathbf{m}), \tag{2.5.1}$$

where $\mathcal{R}$ is a regularisation term to prevent over-solving and incorporate a-priori information, and M some permissible set of parameters (for example, enforcing non-negativity). We have left $p \in \mathbb{N}^+$ unspecified here, but in general the l_2 norm is used for ease of differentiability, making (2.5.1) a least-squares problem. Other choices of norm, such as the l_1 norm, can make the full-wave problem more robust to noise [29].

While the forward operator $\mathcal{F}$ represents a linear PDE, the solution to this PDE varies non-linearly with the parameters $\mathbf{m}$ due to features such as multiple scattering, refraction and diffraction. Therefore, full-wave inversion is to pose the task of imaging the subsurface as a non-linear optimisation problem. FWI is also a relatively large scale problem, since simulating the data will require solving this PDE with sources at many locations and at many frequencies. Further, any iterative solution of (2.5.1) involves at least two data simulations per iteration (one to simulate $\mathbf{d}_{\text{sim}}$ and another to calculate the gradient).

The final difficulty in FWI stems from the fact that changes in $\mathbf{m}$ generally produce far smaller changes in $\mathbf{d}_{\text{sim}} = \mathcal{F}[\mathbf{m}]$, or that many different parameter models produce very similar data. Thus, small differences in data $\mathbf{d}_{\text{obs}} - \mathbf{d}_{\text{sim}}$ can be mapped to much larger perturbations in $\mathbf{m}_{\text{inv}}$ than they were caused by – a feature characteristic of inverse problems – hence the need to incorporate the regularisation term.

FWI was first posed for seismic imaging by Tarantola in 1984 [167], though many other inverse and imaging problems have previously been posed as a nonlinear least-squares problem. Since then FWI has had much development in the seismic industry, and we refer the reader to the review paper by Virieu and Operto [181] and references therein, to save listing all contributions here. Some of the first work on FWI for GPR data was by Ernst et al. [56] and Kuroda et al. [88], both considering 2D inversion of cross-borehole data, and both using a gradient-based method in the time domain. Kuroda et al. inverted only permittivity ϵ, whereas Ernst et al. implemented a scheme which switched between inverting permittivity and conductivity and have successfully applied their algorithm to field data [55]. Meles et al. [110, 113] and Lavoué et al. [90] both proposed algorithms which simultaneously update permittivity and conductivity, whereas Busch et al. [31] have used an algorithm which switches between simultaneous updates of permittivity, phase and source wavelet, and simultaneous updates of conductivity and wavelet amplitude.

A large majority of FWI work for GPR has been carried out in 2D due to the large computational cost of solving 3D electromagnetic problems. Notable work on moving towards 3D FWI include that of Busch et al. [31], who use a 3D forward model to update a layered model of the subsurface, and that of Klotzsche et al. [85], who use a 2D forward model in several slices to update a 3D subsurface model (i.e. they take a tomographic approach). Seismic FWI has now been carried out in 3D using an acoustic model of the subsurface, for example by Ben-Hadj-Ali et al [23] in the frequency domain; as well as in the time domain by Vigh and Starr [180], Warner et al [191], and Houbiers et al [75]. Seismic 3D FWI has so far been limited to low frequencies (< 7 Hz) [181], and 3D elastic FWI is still poses a computational challenge but has been carried out for example by Butzer et al [32].

As discussed in Section 2.2, one needs to incorporate regularisation in order to stabilise the inverse problem. For GPR FWI, the treatment of the regularisation term has had little study. In general as little regularisation as possible is used, for example Ernst et al [56] remarks on the presence of artifacts due to not incorporating a regularisation term. Similarly, Meles et al [110] make no mention of regularisation, and Kuroda et al [88] mention it only as the subject of future work which may improve reconstruction. Meles et al [113] later do discuss regularisation when considering the non-linearity of the problem, but they do not include any in their scheme. They mistakenly state that it does not relieve the non-linearity of the inverse problem, but of course with a greater amount of (linear) regularisation the optimisation problem does become more linear (to the limit where one matches only the prior knowledge).

Lavoué [92] does give a proper account of regularisation, although their attention is restricted to use of a generalised Tikhonov reguarlisation with a discretised Laplace operator (as with (2.2.9)). They find their chosen regularisation is insufficient for designing a stable scheme for joint reconstruction of permittivity and conductivity, needing some other scaling between the two parameter types. The simulations they present though do not have smoothly varying conductivity or permittivity, and so the choice is inappropriate for their numerical experiments, and so their conclusions are limited.

There appears to be a general reluctance in the GPR FWI community to incorporate regularisation (or more than a minimal amount), as if modifying the problem from a data matching one alone will result in incorrect solutions. In many of the papers referenced here, the only form of a regularisation strategy is through choice of an initial inversion model, choice of parameterisation and scaling, and data weighting strategies. In part this is due to a lack of prior knowledge of the subsurface [113], although we would argue that there is likely better *a priori* knowledge available in many applications than might be considered. For example, if one expects smooth variations in the subsurface then regularisation with a discrete Laplacian is appropriate, but if one expects to reconstruct jump changes (distinct solid objects) then something more like Total Variation should be used (see Section 4.3.6). The same is not necessarily true in the case of seismic FWI, where there has been more study of appropriate regularisation methods. For example, Asnaashari et al [18] give an in-depth review of the effect of (generalised) Tikhonov regularisation to incorporate prior knowledge, including the effect of a weighting matrix, allowing dynamic variation of the regularisation parameter λ, and comparing the different roles of a starting model and regularisation term.

2.5.3 Optimisation for the full-wave problem

As mentioned above, the full-wave problem (2.5.1) will need to be solved with a suitable iterative method, in which we generate a sequence of iterates $\{\mathbf{m}_k\}_{k=0}^{k_{\max}}$ which converge towards the unique minimiser of $\mathcal{J}$, $\mathbf{m}_{\text{inv}}$, and we follow Nocedal and Wright [123] for an overview of appropriate methods. The updates will be determined by a local approximation $\mathcal{M}_k$ to $\mathcal{J}$ at $\mathbf{m}_k$. The model $\mathcal{M}$ is often based on the quadratic approximation of the objective function $\mathcal{J}$, i.e. the truncated Taylor series of $\mathcal{J}$

$$\mathcal{M}_k(\mathbf{p}) := \mathcal{J}(\mathbf{m}_k) + \mathbf{p}^T \nabla_{\mathbf{m}} \mathcal{J}(\mathbf{m}_k) + \frac{1}{2}\mathbf{p}^T \nabla^2_{\mathbf{m}} \mathcal{J}(\mathbf{m}_k)\mathbf{p}, \tag{2.5.2}$$

so that the model $\mathcal{M}_k$ approximates $\mathcal{J}$ at $\mathbf{m}_k$ by

$$\mathcal{J}(\mathbf{m}_k + \mathbf{p}) = \mathcal{M}_k(\mathbf{p}) + \mathcal{O}\big(\|\mathbf{p}\|_\infty^3\big) \tag{2.5.3}$$

as $\|\mathbf{p}\|_\infty \to 0$. See Appendix A.2 for the definition of the Landau Big-$\mathcal{O}$ notation. Here $\mathbf{p}$ is some (small) perturbation to $\mathbf{m}_k$ and $\mathcal{M}_k$ is the quadratic model. From this approximation we can derive descent directions $\mathbf{p}_k$, for which $\mathcal{J}(\mathbf{m}_k + \mathbf{p}_k) < \mathcal{J}(\mathbf{m}_k)$. If we assume that $H_k = \nabla^2_{\mathbf{m}} \mathcal{J}(\mathbf{m}_k)$ is positive definite, then the minimiser of $\mathcal{M}_k$ is given by setting the derivative of $\mathcal{M}_k$ to zero, resulting in the *Newton Direction*

$$\mathbf{p}_k = -H_k^{-1}\mathbf{g}_k, \tag{2.5.4}$$

where $\mathbf{g}_k = \nabla_{\mathbf{m}}\mathcal{J}(\mathbf{m}_k)$.

The main drawback of Newton type methods is the requirement to calculate (and store) the inverse Hessian matrix H_k^{-1}, which is prohibitive in FWI due to its size. More generally, we can define a descent direction,

$$\mathbf{p}_k = -B_k^{-1}\mathbf{g}_k. \tag{2.5.5}$$

for some choice of matrix B_k. Equation (2.5.5) results in the steepest descent method if B_k is the identity matrix, and a quasi-Newton method if $B_k \approx H_k$. Quasi-Newton methods make use of the fact that changes in the gradient provide information about the Hessian matrix. That is [123, pp23],

$$\mathbf{g}_{k+1} = \mathbf{g}_k + H_k(\mathbf{m}_{k+1} - \mathbf{m}_k) + o(\|\mathbf{m}_{k+1} - \mathbf{m}_k\|), \tag{2.5.6}$$

as $\|\mathbf{m}_{k+1} - \mathbf{m}_k\| \to 0$. See Appendix A.2 for the definition of the Landau little-o notation. The approximate Hessian can be chosen to mimic property (2.5.6) if it satisfies the secant equation,

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k, \tag{2.5.7}$$

where

$$\mathbf{s}_k = \mathbf{m}_{k+1} - \mathbf{m}_k \qquad \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k. \tag{2.5.8}$$

The most popular formula for B_k [123, pp139] is the BFGS formula (named after its inventors Broyden, Fletcher, Goldfarb and Shanno), which is

$$B_{k+1} = B_k - \frac{B_k\mathbf{s}_k\mathbf{s}_k^T B_k}{\mathbf{s}_k^T B_k \mathbf{s}_k} + \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k}. \tag{2.5.9}$$

Practical quasi-Newton algorithms will store and update B_k^{-1}, rather than calculating B_{k+1} and then solving (2.5.5) for $\mathbf{p}_k$, as the matrix-vector product is generally far cheaper to calculate than solving a linear system. The BFGS algorithm is can be written in this way as

$$B_{k+1} = (I - \rho_k\mathbf{s}_k\mathbf{y}_k^T)B_k(I - \rho_k\mathbf{y}_k\mathbf{s}_k^T) + \rho_k\mathbf{s}_k\mathbf{s}_k^T, \qquad \rho_k = \frac{1}{\mathbf{y}_k^T\mathbf{s}_k}, \tag{2.5.10}$$

where we have redefined B_k to be $B_k := B_k^{-1}$ and I is the identity matrix.

One final suitable class of methods are the (non-linear) conjugate gradient methods, which take for a descent direction

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k\mathbf{p}_{k-1}, \tag{2.5.11}$$

where β_k is a scalar ensuring $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$ are conjugate, which improves the rate of convergence over steepest descent methods. The directions $\mathbf{p}_{k+1}$ and $\mathbf{p}_k$ are conjugate if their inner product with respect to the linearised model $\mathcal{M}_k$ is zero,

$$\langle \mathbf{p}_{k+1}, \mathbf{p}_k \rangle_{\mathcal{M}_k} = 0, \tag{2.5.12}$$

Using the linearised model $\mathcal{M}_k$ of the non-linear optimisation problem, (2.5.12) means that

$$\mathbf{p}_k^T J^T J \mathbf{p}_{k+1} = 0. \tag{2.5.13}$$

Here, J is the Jacobian matrix $J_{ij} = \partial \mathcal{F}_i / \partial m_j$, which forms a linear map between data residuals and model perturbations. The Jacobian is not calculated to find conjugate directions (e.g. by the Fletcher-Reeves method [123, pp. 121]). CG is appealing for large scale optimisation problems because each iteration only requires evaluation of the objective function and its gradient. No matrix operations are required, and there is a very minimal storage cost. For the full-wave problem though, this benefit may not be realised if calculation of the cost and gradient far outweigh the cost of calculating a descent direction. Indeed, we will see that the Hessian matrix has a certain physical meaning in FWI, and so we may wish to incorporate it (or an approximation of it) into our optimisation method in order to improve the rate of convergence.

After an update direction $\mathbf{p}_k$ has been calculated, one must determine how much to update the model in that direction – i.e. the globalisation of the local optimisation problem (2.5.2). The Newton method has a natural steplength of 1, which is the exact minimiser of the quadratic model $\mathcal{M}_k$. However, gradient, CG and quasi-Newton methods may be badly scaled, and a steplength of 1 may either be an insufficient update or too large a step. Instead, a line search can then be carried out in the direction $\mathbf{p}_k$ to find

$$\alpha_k = \operatorname*{argmin}_{\alpha>0} \mathcal{J}(\mathbf{m}_k + \alpha \mathbf{p}_k), \tag{2.5.14}$$

after which we set $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{p}_k$. Line search algorithms will in general not solve (2.5.14) exactly, but attempt to find an α_k which satisfies the strong Wolfe conditions,

$$\mathcal{J}(\mathbf{m}_k + \alpha_k \mathbf{p}_k) \leq \mathcal{J}(\mathbf{m}_k) + c_1 \alpha_k \nabla_{\mathbf{m}} \mathcal{J}(\mathbf{m}_k)^T \mathbf{p}_k \tag{2.5.15a}$$

$$|\nabla_{\mathbf{m}} \mathcal{J}(\mathbf{m}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k| \leq c_2 |\nabla_{\mathbf{m}} \mathcal{J}(\mathbf{m}_k)^T \mathbf{p}_k|, \tag{2.5.15b}$$

where $0 < c_1 < c_2 < 1$. (2.5.15a) ensures the cost function has decreased sufficiently, and (2.5.15b) ensures α lies close to a local minimum. For quasi-Newton methods, if the approximate Hessian is close to the true Hessian then the steplength $\alpha = 1$ often satisfies the Wolfe conditions[pp. 44, 46] [123], and so should be tested first. In steepest descent and CG methods, often the curvature condition (2.5.15b) is ignored and one only attempts to satisfy the sufficient decrease (Armijo) condition (2.5.15a).

All of the above methods, and variants thereof, have been applied to solving the 2D FWI problem, with varying degrees of success. For example, in GPR several have used the steepest descent method [55, 56, 110, 113, 113], which has a minimal cost of calculating a descent direction but may converge slowly, while Kuroda et al [88] noted the convergence increases which can be gained by using a nonlinear CG method. Lavouè et al [90, 91] have applied the l-BFGS-b (limited memory BFGS bounded) method to simultaneous reconstruction of permittivity and conductivity, and found the algorithm efficiently dealt with non-linearities during the optimisation procedure. For seismic FWI, we highlight the paper by Pratt et al. [137], which compares Gauss-Newton and full Newton methods, showing how incorporating the (approximate) Hessian can accelerate convergence of the optimisation scheme though at large computational costs. Further, Métivier et al. [114–116] demonstrated how the Truncated Newton method, which efficiently calculates only a small number of Hessian vector products directly without calculating (and storing) the full Hessian matrix, can yield an effective algorithm (performing favourably when comapared to l-BFGS), particularly where multiple scattering dominates the data.

2.5.4 Efficient gradient calculation

Consider the data misfit term $\mathcal{J}(\mathbf{m}) = \frac{1}{2}\|\mathcal{F}[\mathbf{m}]-\mathbf{d}\|^2 = \frac{1}{2}\delta\mathbf{d}^T\delta\mathbf{d}^*$, where $\delta\mathbf{d} = \mathcal{F}[\mathbf{m}-\mathbf{d}$ are the data residuals, ^{T} denotes the transpose and * denotes complex conjugation. The gradient of the data misfit term is given by

$$\nabla_{\mathbf{m}}\mathcal{J} = \frac{\partial \mathcal{J}}{\partial \mathbf{m}} = \Re\{J^T\delta\mathbf{d}^*\}, \tag{2.5.16}$$

where J is the Jacobian matrix

$$J_{ij} = \frac{\partial \mathcal{F}_i}{\partial m_j}, \qquad i = 1, \dots, n, \quad j = 1, \dots, p, \tag{2.5.17}$$

n is the number of data points (number of receivers × number of frequencies × number of source locations) and p the number of subsurface parameters. J is a large dense matrix, and so computation and storage of this matrix is at least undesirable if not infeasible. Following Pratt et al. [137], we show the gradient vector can be calculated directly, without forming J, via the adjoint state method.

Without loss of generality, consider the single frequency, single source experiment where the forward operator $\mathcal{F}$ is to solve for $\mathbf{u}$ the linear system $A(\mathbf{m})\mathbf{u} = \mathbf{s}$ and evaluate at the first n mesh/grid points which correspond to the first n parameters m_i (grid points can be re-ordered arbitrarily, and for multiple sources and frequencies we simply sum each corresponding gradient vector). Then (2.5.16) can be written as

$$\begin{bmatrix} \frac{\partial \mathcal{J}}{\partial m_1} \\ \frac{\partial \mathcal{J}}{\partial m_2} \\ \vdots \\ \frac{\partial \mathcal{J}}{\partial m_p} \end{bmatrix} = \Re \left\{ \begin{bmatrix} \frac{\partial u_1}{\partial m_1} & \frac{\partial u_2}{\partial m_1} & \cdots & \frac{\partial u_p}{\partial m_1} \\ \frac{\partial u_1}{\partial m_2} & \frac{\partial u_2}{\partial m_2} & \cdots & \frac{\partial u_p}{\partial m_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_1}{\partial m_p} & \frac{\partial u_2}{\partial m_p} & \cdots & \frac{\partial u_p}{\partial m_p} \end{bmatrix} \begin{bmatrix} \delta d_1^* \\ \vdots \\ \delta d_n^* \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\}, \tag{2.5.18}$$

where we have extended the vector $\delta\mathbf{d}$ to contain 0s where a measurement is not taken at a grid point. Taking partial derivatives of the forward system with respect to parameter m_i, we find

$$A\frac{\partial \mathbf{u}}{\partial m_i} + \frac{\partial A}{\partial m_i}\mathbf{u} = 0, \tag{2.5.19}$$

which we can rewrite as

$$\frac{\partial \mathbf{u}}{\partial m_i} = A^{-1}\left(-\frac{\partial A}{\partial m_i}\mathbf{u}\right) = A^{-1}\mathbf{f}^{(i)}, \tag{2.5.20}$$

where $\mathbf{f}^{(i)}$ is referred to as the i^{th} virtual source term. Thus, the Jacobian can be written as

$$J = A^{-1}\left[\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(p)}\right] = A^{-1}F. \tag{2.5.21}$$

Substituting (2.5.21) into (2.5.16), we find

$$\nabla_{\mathbf{m}}\mathcal{J} = -\Re\left\{(A^{-1}F)^T\delta\mathbf{d}^*\right\} = -\Re\left\{F^T(A^T)^{-1}\delta\mathbf{d}^*\right\}. \tag{2.5.22}$$

So, calculation of the gradient via the adjoint-state method requires one more forward solve (per frequency and source location), whereas calculation of the Jacobian would take p forward solves via (2.5.21).

Consider the i^{th} component of the gradient vector for a single source single frequency data set,

$$(\nabla_{\mathbf{m}}\mathcal{J})_i = \Re\left\{\mathbf{u}^T\frac{\partial A^T}{\partial m_i}\left(A^T\right)^{-1}\delta\mathbf{d}^*\right\} = \Re\left\{\mathbf{f}^{(i)^T}\mathbf{v}\right\}. \tag{2.5.23}$$

The system matrix A will in general be symmetric, except for complex parts of the matrix corresponding to absorbing boundary conditions derived from the Sommerfeld radiation condition. The term $\mathbf{v} = (A^T)^{-1}\delta\mathbf{d}^*$ is like a back-projected data residual – i.e. reversing time. The term $\partial A/\partial m_i$ consists only of highly local nonzeros, and so we see the part of the backprojected data residual local to m_i is scaled and multiplied by the forward predicted wave field - i.e. a convolution in the time domain. This is demonstrated in Figure 2.2, which shows the forward and adjoint fields for a single source location at 1 GHz, their correlation and contribution to the gradient, the sum of contributions from all source locations at 1 GHz and the full gradient term (summed over all frequency components). We see that as more (unique) measurements are added, the components constructively interfere where a perturbation in wavespeed causing reflection is present, and destructively interfere where one is not.

2.5.5 The effect of the Hessian in full-wave inversion

Numerical evidence [137] suggests the Hessian matrix $H = \nabla^2_{\mathbf{m}}\mathcal{J}$ plays an important role in both the stability and accelerating convergence rates in solving the full-wave problem (2.5.1), though it is more costly both to compute and store than the Jacobian. Let us write the Hessian as the sum of two parts,

$$H(\mathbf{m}) = B(\mathbf{m}) + C(\mathbf{m}), \tag{2.5.24}$$

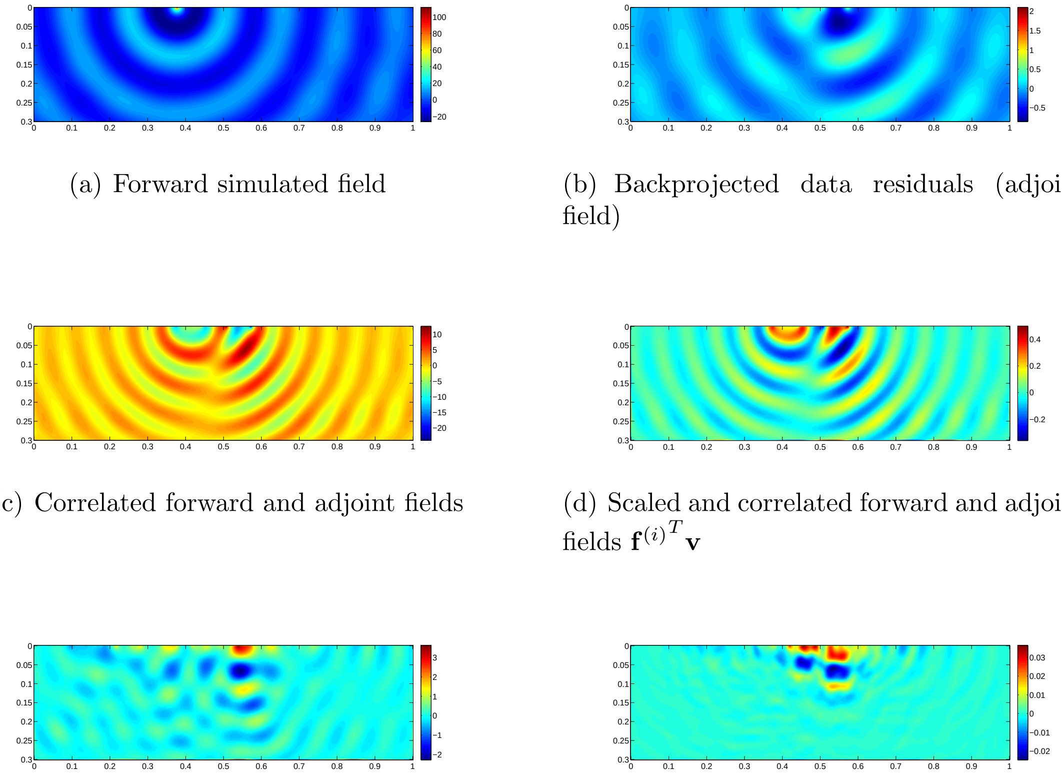
where the matrices B and C are given by

$$B(\mathbf{m}) = \Re\left\{J^T J^*\right\} \tag{2.5.25a}$$

$$C(\mathbf{m}) = \Re\left\{\left[\left(\frac{\partial J^T}{\partial m_1}\right)\delta d^*, \left(\frac{\partial J^T}{\partial m_2}\right)\delta d^*, \dots, \left(\frac{\partial J^T}{\partial m_p}\right)\delta d^*\right]\right\}. \tag{2.5.25b}$$

B is referred to as the Gauss-Newton approximation of the Hessian.

Again, let us consider a single source single frequency experiment where the first n elements of solution vector $\mathbf{u}$ correspond to receiver locations. Each element of B is



(a) Forward simulated field

(b) Backprojected data residuals (adjoint field)

(c) Correlated forward and adjoint fields

(d) Scaled and correlated forward and adjoint fields $\mathbf{f}^{(i)^T}\mathbf{v}$

(e) Sum of gradient contributions over all source locations at 1 GHz

(f) Full gradient (summed over frequencies)

Figure 2.2: Formulation of the gradient via the adjoint state method from a single source point at 1GHz. (a) shows the forward simulated wave, (b) the back-projected data residuals, (c) the correlated forward and adjoint fields, (d) the gradient contribution from this measurement, (e) the sum of gradient contributions over all source locations at 1 GHz and (f) the full gradient (summed over 100 frequencies between 1 and 3 GHz). The true domain is shown in Figure 4.2 in Chapter 4, and the gradient is calculated from a homogeneous background.

then given by

$$B_{ij} = \Re \left\{ \sum_{r=1}^{n} \frac{\partial u_r}{\partial m_i} \frac{\partial u_r^*}{\partial m_j} \right\}, \tag{2.5.26}$$

which is the zero-lag correlation of the first derivatives of the wave field at receiver locations with respect to parameters m_i and m_j. The elements B_{ij} then are at a maximum for the autocorrelation $i = j$, and decrease with distance between i and j. B would be a diagonal matrix in the high frequency approximation, but in general it is banded. Further, we know that the amplitudes of the derivatives decrease with distance from the source/receiver, and so gradient based optimisation methods are less sensitive to perturbations in $\mathbf{m}$ away from the source/receiver. B^{-1} therefore acts as a 'refocussing filter' in the descent direction, compensating for the different illumination of parameters by the acquisition array [115, 137].

For the same single source, single frequency experiment, each element of C is given by

$$C_{ij} = \Re \left\{ \sum_{r=1}^{n} \frac{\partial^2 u_r}{\partial m_i \partial m_j} \delta d_r^* \right\}, \tag{2.5.27}$$

the zero-lag correlation of the data residuals and the second order derivatives of the wave field at the receivers with respect to m_i and m_j. These second order derivatives represent a recorded signal which has been scattered twice, at m_i and m_j [137]. The affect of C^{-1} on the descent direction is therefore to compensate for first order multiple reflections.

Chapter 3

Electromagnetic Theory

3.1 Introduction

The dynamics of the electromagnetic waves emitted from a GPR antenna, transmitted into the ground, scattered in the subsurface, and received at another antenna, are governed by Maxwell's equations – named after James Clerk Maxwell who published an early form of the equations between 1861 and 1862 [108]. In this chapter we describe these fundamental equations, for completeness of the thesis and to reference for a reader unfamiliar with electromagnetic motion.

First, we give Maxwell's equations in both differential time and differential time-harmonic form, and constitutive relations for material properties. We then derive vector (curl-curl) wave equations for the electric field in 3D, and show that these reduce to the Helmholtz equation in 2D, as well as giving the variational form for the vector wave equation. Boundary conditions and free-space Green's functions are also covered. Finally, we describe the surface equivalence principle which describes the field radiated from a domain as integrals of the tangential electric and magnetic fields on the boundary of the domain. We follow Jin [80], Monk [119] and Volakis et al [186] for an overview of Maxwell's equations.

3.2 Maxwell's equations of electrodynamics

3.2.1 Maxwell's equations in differential time-domain form

In a closed domain Ω, with boundary $\partial\Omega$, where all the field quantities and their derivatives are continuous, and where the material properties behave linearly with respect to the applied electromagnetic fields, Maxwell's equations in differential (point) form are given by [80, 119, 186]

$$\nabla \times \mathbf{E}(\mathbf{r},t) = -\frac{\partial \mathbf{B}(\mathbf{r},t)}{\partial t} - \mathbf{M}(\mathbf{r},t) \qquad \text{(Faraday's Law)} \tag{3.2.1a}$$

$$\nabla \times \mathbf{H}(\mathbf{r},t) = \frac{\partial \mathbf{D}(\mathbf{r},t)}{\partial t} + \mathbf{J} \qquad \text{(Maxwell-Ampère law)} \tag{3.2.1b}$$

$$\nabla \cdot \mathbf{D}(\mathbf{r},t) = \rho(\mathbf{r},t) \qquad \text{(Gauss' law)} \tag{3.2.1c}$$

$$\nabla \cdot \mathbf{B}(\mathbf{r},t) = \rho_{\mathrm{m}}(\mathbf{r},t) \qquad \text{(Gauss' law - magnetic)} \tag{3.2.1d}$$

$$\nabla \cdot \mathbf{J}(\mathbf{r},t) = -\frac{\partial \rho(\mathbf{r},t)}{\partial t} \qquad \text{(continuity)} \tag{3.2.1e}$$

$$\nabla \cdot \mathbf{M}(\mathbf{r},t) = -\frac{\partial \rho_{\mathrm{m}}(\mathbf{r},t)}{\partial t} \qquad \text{(continuity - magnetic),} \tag{3.2.1f}$$

where the vector fields are given by

$$\begin{aligned}
\mathbf{E} &= \text{electric field in volts/meter} \quad (\text{V/m}) \\
\mathbf{H} &= \text{magnetic field in amperes/meter} \quad (\text{A/m}) \\
\mathbf{J} &= \text{electric current density in amperes/meter}^2 \quad (\text{A/m}^2) \\
\mathbf{M} &= \text{magnetic current density in volts/meter}^2 \quad (\text{V/m}^2) \\
\mathbf{D} &= \text{electric flux density in coulombs/meter}^2 \quad (\text{C/m}^2) \\
\mathbf{B} &= \text{magnetic flux density in webers/meter}^2 \quad (\text{Wb/m}^2)
\end{aligned}$$

and the scalar quantities are

$$\begin{aligned}
\rho &= \text{electric charge density in coulombs/meter}^3 \quad (\text{C/m}^3) \\
\rho_m &= \text{magnetic charge density in webers/meter}^3 \quad (\text{Wb/m}^3).
\end{aligned}$$

The Maxwell-Ampère law tells us that both flowing electric current and an electric field changing in time will give rise to a magnetic field encircling them. Similarly, Faraday's law states that a magnetic field changing in time will give rise to a circulating electric field. Faraday's law also includes a magnetic current density **M** which can give rise to a ciculating **E**-field - note that this is a fictitious quantity (as is ρ_{m}) introduced for convenience in solving Maxwell's equations or representing solutions. Often Maxwell's equations are written with these terms neglected.

Gauss' Law states that an electric charge acts as a source or a sink for electric fields - it tells us that **D** field lines will diverge away from positive charges, diverge towards negative, and will start and stop on charges. It also tells us that the divergence of **D** is equal to the net amount of charge in that region. Gauss' Law for Magnetism says the equivalent for magnetic flux density. Since ρ_{m} is a fictitious quantity, setting $\rho_{\mathrm{m}} = 0$ we see that the divergence of the magnetic flux are zero, and so the fields will tend to form closed loops.

Finally, the continuity equation ensures that if current is flowing into or out of a region, this is met by an increase or decrease in charge, respectively. Because of the continuity equation, only 3 of the Maxwell equations are independent (discounting the magnetic continuity equation for the fictitious quantities).

3.2.2 Constitutive relations and material properties

As we noted in Section 3.2.1, two of Maxwell's equations are dependent or *auxiliary*. Removing two dependent equations we are left with an indefinite system (more unknowns than equations), and so we must therefore have some constitutive relations between the field quantities. Constitutive relations describe the macroscopic properties of the medium, and for a simple inhomogeneous and isotropic medium these are

$$\mathbf{D} = \epsilon(\mathbf{r})\mathbf{E} \tag{3.2.2a}$$

$$\mathbf{B} = \mu(\mathbf{r})\mathbf{H}, \tag{3.2.2b}$$

where the scalars $0 < \epsilon < \infty$ and $0 < \mu < \infty$ are (electric) permittivity measured in fads/meter (F/m) and (magnetic) permeability measured in henrys/meter (H/m), respectively. More generally, for an anisotropic material, ϵ and μ are 3×3 symmetric tensor fields

$$\epsilon(\mathbf{r}) = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix}, \qquad \mu(\mathbf{r}) = \begin{bmatrix} \mu_{xx} & \mu_{xy} & \mu_{xz} \\ \mu_{yx} & \mu_{yy} & \mu_{yz} \\ \mu_{zx} & \mu_{zy} & \mu_{zz} \end{bmatrix} \qquad \forall \mathbf{r} \in \Omega, \tag{3.2.3}$$

which are in general positive definite. ϵ and μ may also be considered functions of frequency ω, and for non-linear materials these are also functions of the electromagnetic fields (such as ferroelectric and ferromagnetic materials). However, for the purposes of this thesis, we shall assume throughout that the variation of all electromagnetic parameters with ω is negligible over the range of frequencies of interest, and we will not consider non-linear materials.

Further, the current densities an be written as the sum of impressed (excitation) and induced (conduction) currents as

$$\mathbf{J} = \mathbf{J}_{\mathrm{i}} + \mathbf{J}_{\mathrm{c}} = \mathbf{J}_{\mathrm{i}} + \sigma(\mathbf{r})\mathbf{E}, \tag{3.2.4a}$$

$$\mathbf{M} = \mathbf{M}_{\mathrm{i}} + \mathbf{M}_{\mathrm{c}} = \mathbf{M}_{\mathrm{i}} + \sigma_{\mathrm{m}}(\mathbf{r})\mathbf{H}, \tag{3.2.4b}$$

where (3.2.4a) is Ohm's law. Here, σ is the electric current conductivity in siemens/m (S/m)), and σ_{m} is the (fictitious) magnetic current conductivity in ohms/m (Ω/m). Similarly, σ and σ_r are scalars for an isotropic material, and 3×3 symmetric tensors for an anisotropic material.

3.2.3 Time-harmonic Maxwell's equations

If we assume a time dependence of $\exp(-\mathrm{i}\omega t)$, $\mathrm{i} = \sqrt{-1}$, for the electric and magnetic fields and flux densities, where $\omega = 2\pi f$ is the angular frequency, then Maxwell's equations (3.2.1) reduce to

$$\nabla \times \mathbf{E}(\mathbf{r},\omega) = -\mathrm{i}\omega\mathbf{B}(\mathbf{r},\omega) - \mathbf{M}(\mathbf{r},\omega), \tag{3.2.5a}$$

$$\nabla \times \mathbf{H}(\mathbf{r},\omega) = \mathrm{i}\omega\mathbf{D}(\mathbf{r},\omega) + \mathbf{J}(\mathbf{r},\omega), \tag{3.2.5b}$$

$$\nabla \cdot \mathbf{D}(\mathbf{r},\omega) = \rho(\mathbf{r},\omega), \tag{3.2.5c}$$

$$\nabla \cdot \mathbf{B}(\mathbf{r},\omega) = \rho_{\mathrm{m}}(\mathbf{r},\omega), \tag{3.2.5d}$$

$$\nabla \cdot \mathbf{J}(\mathbf{r},\omega) = -\mathrm{i}\omega\rho(\mathbf{r},\omega), \tag{3.2.5e}$$

$$\nabla \cdot \mathbf{M}(\mathbf{r},\omega) = -\mathrm{i}\omega\rho_{\mathrm{m}}(\mathbf{r},\omega), \tag{3.2.5f}$$

in which we have abused notation somewhat by using the same bold-face vectors to denote the complex amplitudes of the aforementioned time-dependent real-valued fields. This notation is continued hereafter, and we also drop the function dependencies $(\mathbf{r},\omega)$. Making use of the constitutive relations given in Section 3.2.2, equations (3.2.5) reduce to

$$\nabla \times \mathbf{E} = -\mathbf{M} - \mathrm{i}\omega\mu\mathbf{H} = -\mathbf{M}_{\mathrm{i}} - (\sigma_{\mathrm{m}} + \mathrm{i}\omega\mu)\,\mathbf{H} \tag{3.2.6a}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \mathrm{i}\omega\epsilon\mathbf{E} = \mathbf{J}_{\mathrm{i}} + (\sigma + \mathrm{i}\omega\epsilon)\,\mathbf{E} \tag{3.2.6b}$$

$$\nabla \cdot (\epsilon\mathbf{E}) = \rho = -\left(\nabla \cdot \mathbf{J}_{\mathrm{i}}\right)/\mathrm{i}\omega \tag{3.2.6c}$$

$$\nabla \cdot (\mu\mathbf{H}) = \rho_{\mathrm{m}} = -\left(\nabla \cdot \mathbf{M}_{\mathrm{i}}\right)/\mathrm{i}\omega. \tag{3.2.6d}$$

Common equivalent expressions involve using the complex permittivity $\dot{\epsilon} = \epsilon - \mathrm{i}\sigma/\omega$ and permeability $\dot{\mu} = \mu - \mathrm{i}\sigma_{\mathrm{m}}/\omega$.

3.3 Boundary conditions

The solution to Maxwell's equations in some domain Ω becomes unique upon application of boundary conditions on $\partial\Omega$. Consider a (possibly charged) interface $\partial\Omega$ between two domains Ω_1 and Ω_2, as shown in Figure 3.1. Natural boundary conditions at the interface can be derived through Maxwell's equations in integral form [186], and are given by

$$\hat{\mathbf{n}} \times (\mathbf{E}_1 - \mathbf{E}_2) = -\mathbf{M}_{\mathrm{is}}, \tag{3.3.1a}$$

$$\hat{\mathbf{n}} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_{\mathrm{is}}, \tag{3.3.1b}$$

$$\hat{\mathbf{n}} \cdot (\epsilon_1\mathbf{E}_1 - \epsilon_2\mathbf{E}_2) = \rho_{\mathrm{s}}, \tag{3.3.1c}$$

$$\hat{\mathbf{n}} \cdot (\mu_1\mathbf{H}_1 - \mu_2\mathbf{H}_2) = \rho_{\mathrm{ms}}, \tag{3.3.1d}$$

where $\mathbf{E}_i$ and $\mathbf{H}_i$ are the electric and magnetic fields in medium i, $\mathbf{J}_{\mathrm{is}}$ and $\mathbf{M}_{\mathrm{is}}$ are the impressed electric and magnetic current densities at the interface, ϵ_i and μ_i are the permittivity and permeability in medium i, ρ_{s} and ρ_{ms} are the electric surface charge density and the (fictious) magnetic charge density at the interface, and $\hat{\mathbf{n}}$ the outward unit normal to $\partial\Omega$. Ignoring the fictitous magnetic currents and charges, we have

$$\hat{\mathbf{n}} \times (\mathbf{E}_1 - \mathbf{E}_2) = \mathbf{0}, \tag{3.3.2a}$$

$$\hat{\mathbf{n}} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_{\mathrm{is}}, \tag{3.3.2b}$$

$$\hat{\mathbf{n}} \cdot (\epsilon_1\mathbf{E}_1 - \epsilon_2\mathbf{E}_2) = \rho_{\mathrm{s}}, \tag{3.3.2c}$$

$$\hat{\mathbf{n}} \cdot (\mu_1\mathbf{H}_1 - \mu_2\mathbf{H}_2) = 0. \tag{3.3.2d}$$

Equations (3.3.2) are known as the field continuity, or transmission, conditions. The first condition states that the tangential component of the electric field is continuous across the surfaces. Similarly, the second states that the magnetic field is discontinuous by an amount equal to the impressed electric current $\mathbf{J}_{\text{is}}$, which we note will be zero unless a source is actually placed at the surface. Similarly, the second pair state that the normal component of the magnetic field differs by a factor of $\mu_1 {\mu_2}^{-1}$, and that the normal components of $\epsilon\mathbf{E}$ and are discontinuous by an amount equal to ρ_{s}.

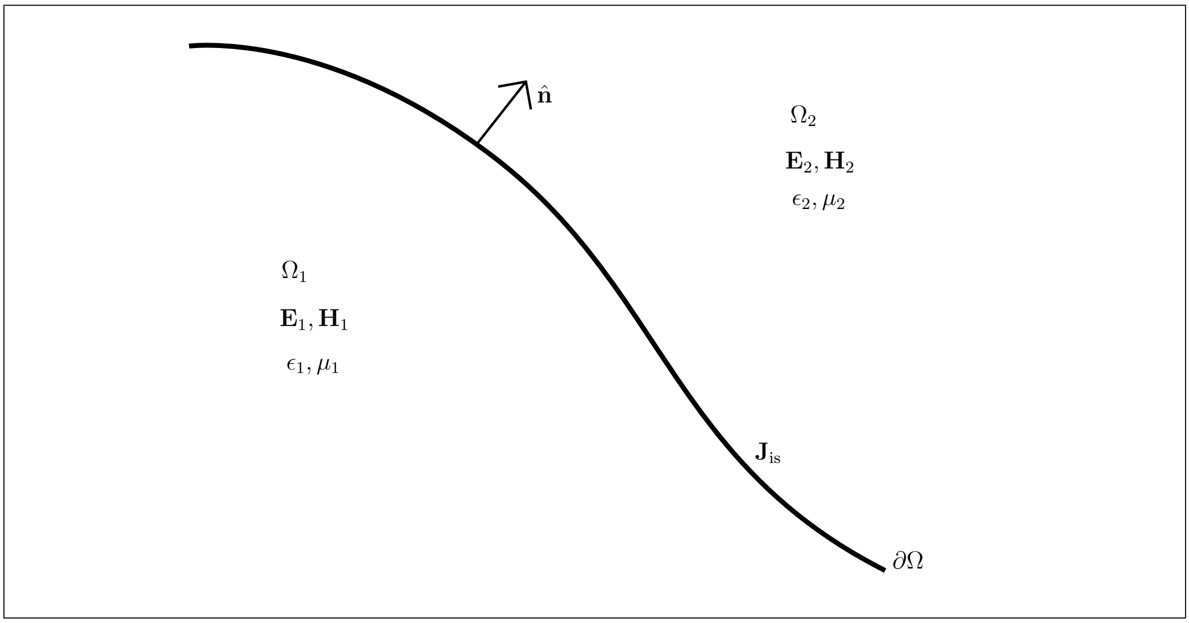


Figure 3.1: Boundary between two media.

If medium 2 is a perfect electric conductor, the continuity conditions reduce to

$$\hat{\mathbf{n}} \times (\mathbf{E}_1) = \mathbf{0}, \tag{3.3.3a}$$

$$\hat{\mathbf{n}} \times (\mathbf{H}_1) = \mathbf{J}_{\text{is}}, \tag{3.3.3b}$$

$$\hat{\mathbf{n}} \cdot (\epsilon_1 \mathbf{E}_1) = \rho_{\text{s}}, \tag{3.3.3c}$$

$$\hat{\mathbf{n}} \cdot (\mu_1 \mathbf{H}_1) = 0, \tag{3.3.3d}$$

since medium 2 cannot sustain an internal field.

In some settings, it can either be difficult or costly to impose the field continuity conditions (3.3.2). In these cases, one may make use of approximate impedance–type boundary conditions such as the Leontovich Boundary Conditions [66, 98, 151, 186],

$$\begin{aligned} \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{E}) =& -\eta Z_0 \hat{\mathbf{n}} \times \mathbf{H}, \\ \hat{\mathbf{n}} \times \mathbf{E} =& \frac{Z_0}{\eta} \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{H}), \end{aligned} \tag{3.3.4}$$

where η is the normalised characteristic impedance and $Z_0 = \sqrt{\mu_0 \epsilon_0^{-1}}$ the free-space impedance in Ohms (Ω), with $\epsilon_0 \approx 8.854 \times 10^{-12}$ F/m is the free-space permittivity, $\mu_0 = 4\pi \times 10^{-7}$ H/m . The characteristic impedance is often taken to be $\eta = \sqrt{\mu_r \epsilon_r^{-1}}$ when one side of the medium is free-space, where ϵ_r and μ_r are the relative permittivity and permeability of the material defined by $\epsilon = \epsilon_0 \epsilon_r$ and $\mu = \mu_0 \mu_r$. This follows by demanding the reflected field due to (3.3.4) is the same as that given by the natural boundary conditions (3.3.2), and is exact for an infinite planar interface only but an approximation for curved boundaries relying on $|\mathrm{Im}\left\{\sqrt{\epsilon_r \mu_r}\right\}| \gg 1$. Other choices are appropriate for approximating for example a coated conductor or a material which varies slowly in the transverse plane [186, pp 17-19].

For an unbounded or open domain, we must specify a condition on $\mathbf{E}$ and $\mathbf{H}$ as $r \to \infty$ in order to obtain a unique solution, where $r = \sqrt{x^2 + y^2 + z^2}$. For causal solutions – i.e. for waves generated by the $\mathbf{J}$ and ρ within the domain and not by waves coming in from infinity - $\mathbf{E}$ and $\mathbf{H}$ are required to satisfy the radiation condition

$$\lim_{r \to \infty} r \left[\nabla \times \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix} + \mathrm{i} k_0 \hat{\mathbf{r}} \times \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix} \right] = \mathbf{0}, \tag{3.3.5}$$

where $\hat{\mathbf{r}} = (x\hat{\mathbf{e}}_x + y\hat{\mathbf{e}}_y + z\hat{\mathbf{e}}_z)/r$ and $k_0 = 2\pi\lambda_0^{-1} = \omega\sqrt{\epsilon_0 \mu_0}$ is the free-space wavenumber (1/m). (3.3.5) is usually referred to as the Sommerfeld radiation condition for general three-dimensional fields. For 2D fields in the $y = 0$ plane, the condition becomes

$$\lim_{r \to \infty} \sqrt{r} \left[\frac{\partial}{\partial r} \begin{pmatrix} E_y \\ H_y \end{pmatrix} + \mathrm{i} k_0 \begin{pmatrix} E_y \\ H_y \end{pmatrix} \right] = 0, \tag{3.3.6}$$

$r = \sqrt{x^2 + z^2}$, which we note provides conditions on E_x, E_z, H_x and H_z components via Maxwell's equations (3.2.6).

3.4 Wave equations

3.4.1 Vector wave equations

Recall that Faraday's equation states that a changing magnetic field gives rise to a circling electric field, and that Ampère's Law states that a change in electric field gives rise to circling magnetic field. Clearly, the coupling of the two results in wave motion: the magnetic field leads to a change in electric, which leads to a change in magnetic, and so forth. This is evident if we combine equations (3.2.6a) and (3.2.6b) to form second-order vector wave equations – the form of Maxwell's equations which the finite element method makes use of.

Taking the curl of (3.2.6a),

$$\nabla \times \left(\mu^{-1}\nabla \times \mathbf{E}\right) + \mathrm{i}\omega\nabla \times \mathbf{H} = -\nabla \times \mu^{-1}\mathbf{M}. \tag{3.4.1}$$

and substituting for the curl of **H** via Ampère's law (3.2.6b),

$$\nabla \times \left(\mu^{-1}\nabla \times \mathbf{E}\right) + \omega^2\epsilon\mathbf{E} = -\mathrm{i}\omega\mathbf{J} - \nabla \times \mu^{-1}\mathbf{M}. \tag{3.4.2}$$

Substituting $\epsilon = \epsilon_0\epsilon_r$ and $\mu = \mu_0\mu_r$, we have

$$\nabla \times \left(\mu_r^{-1}\nabla \times \mathbf{E}\right) - k_0^2\epsilon_r\mathbf{E} = -\mathrm{i}k_0 Z_0\mathbf{J} - \nabla \times \mu_r^{-1}\mathbf{M}, \tag{3.4.3}$$

which is an inhomogeneous vector wave equation for **E**. In terms of $\mathbf{J}_\mathrm{i}$, (3.4.3) can be written as

$$\nabla \times \left(\mu_r^{-1}\nabla \times \mathbf{E}\right) + \left(\mathrm{i}k_0 Z_0\sigma - k_0^2\epsilon_r\right)\mathbf{E} = -\mathrm{i}k_0 Z_0\mathbf{J}_\mathrm{i} - \nabla \times \mu_r^{-1}\mathbf{M}. \tag{3.4.4}$$

Here, λ_0 the free-space wavelength (m), and $Z_0 = \sqrt{\mu_0\epsilon_0^{-1}}$ the free-space impedance in ohms (Ω). Similarly, solving instead for the magnetic field leads to the dual of (3.4.4),

$$\nabla \times \left(\epsilon_r^{-1}\nabla \times \mathbf{H}\right) + \left(\mathrm{i}k_0 Y_0\sigma - k_0^2\mu_r\right)\mathbf{H} = -\mathrm{i}k_0 Y_0\mathbf{M} - \nabla \times \epsilon_r^{-1}\mathbf{J}_\mathrm{i}, \tag{3.4.5}$$

where $Y_0 = Z_0^{-1}$ is free-space admittance (S).

3.4.2 Scalar wave equations

If we assume that both the medium of interest (described by μ_r, ϵ_r and σ) and the electric and magnetic fields are invariant in one coordinate direction, we can reduce the 3D vector wave equation to a 2D scalar one. Considering a medium and field invariant in the $\hat{\mathbf{y}}$ direction, the y component of equation (3.4.4) reduces to

$$\left[\frac{\partial}{\partial x}\left(\mu_r^{-1}\frac{\partial}{\partial x}\right) + \frac{\partial}{\partial z}\left(\mu_r^{-1}\frac{\partial}{\partial z}\right) + \left(k_0^2\epsilon_r - \mathrm{i}k_0 Z_0\sigma\right)\right]E_y = -\mathrm{i}k_0 Z_0 J_y + \frac{\partial}{\partial x}\mu_r^{-1}M_z - \frac{\partial}{\partial z}\mu_r^{-1}M_x. \tag{3.4.6}$$

We can write this in the form

$$\left[\nabla\left(\mu_r^{-1}\nabla\right) + \left(k_0^2\epsilon_r - \mathrm{i}k_0 Z_0\sigma\right)\right]E_y = S, \tag{3.4.7}$$

and for a homogeneous μ_r

$$\left[\nabla^2 + k^2\right] E_y = S \tag{3.4.8}$$

(absorbing μ_r into S), where ∇ is the 2D Laplace operator in the x, z plane, and the complex wavenumber $k^2 = \omega^2/c^2$ with complex wave speed (assuming scalar valued permittivity and permeability). Note that (3.4.7) and (3.4.8) are in the familiar form of the Helmholtz equation, as also describes acoustic waves in a compressible medium.

For a homogeneous and charge free domain Ω, (3.4.4) reduces to

$$\nabla \times \nabla \times \mathbf{E} - k^2\mathbf{E} = 0. \tag{3.4.9}$$

Using the identity $\nabla \times \nabla \times \mathbf{f} \equiv \nabla(\nabla \cdot \mathbf{f}) - \nabla^2 \mathbf{f}$, and noting that by Gauss' law $\nabla \cdot \mathbf{E} = 0$, we again arrive at the Helmholtz equation

$$\nabla^2 u + k^2 u = 0 \tag{3.4.10}$$

for each $u = E_x, E_y, E_z$.

3.5 Variational formulation

Consider the following model scattering problem, in which the $\mathbf{E}$ field satisfies governing equations and boundary conditions

$$\nabla \times \left(\mu_r^{-1} \nabla \times \mathbf{E}\right) - k_0^2 \epsilon_r \mathbf{E} = -\mathrm{i} k_0 Z_0 \mathbf{J} \tag{3.5.1a}$$

$$\nabla \cdot (\epsilon_r \mathbf{E}) = 0, \tag{3.5.1b}$$

$$\hat{\mathbf{n}} \times \mathbf{E} = \mathbf{f} \qquad \text{on } \Gamma_D, \tag{3.5.1c}$$

$$\hat{\mathbf{n}} \times (\mu_0 \mu_r)^{-1} \nabla \times \mathbf{E} = \mathbf{g} \qquad \text{on } \Gamma_N, \tag{3.5.1d}$$

where $\Gamma_D \cup \Gamma_N = \partial\Omega$ is the (closed) boundary of the Lipschitz domain $\Omega \subset \mathbb{R}^3$, which may be formed of multiple materials so that μ_r and ϵ_r are functions of space. (3.5.1) is the strong form of the scattering problem. To derive the variational formulation, following [136], we multiply (3.5.1a) by an arbitrary test function $\mathbf{F} \in H^1_{\text{curl}}(\Omega)$, where $H^1_{\text{curl}}(\Omega)$ is defined by (A.1.11), and integrate over the domain Ω, giving

$$\int_\Omega \left\{\mathbf{F} \cdot \nabla \times \left(\mu_r^{-1} \nabla \times \mathbf{E}\right) - \mathbf{F} \cdot k_0^2 \epsilon_r \mathbf{E}\right\} \mathrm{d}V = -\mathrm{i} k_0 Z_0 \int_\Omega \mathbf{F} \cdot \mathbf{J} \, \mathrm{d}V. \tag{3.5.2}$$

Using the identity

$$\mathbf{a}\cdot(\nabla\times\mathbf{b}) = (\nabla\times\mathbf{a})\cdot\mathbf{b} - \nabla\cdot(\mathbf{a}\times\mathbf{b}),$$

we arrive at

$$\begin{aligned}\int_\Omega (\nabla\times\mathbf{F})\cdot\big(\mu_r^{-1}\nabla\times\mathbf{E}\big)\,\mathrm{d}V - \int_\Omega \nabla\cdot\big(\mathbf{F}\times\big(\mu_r^{-1}\nabla\times\mathbf{E}\big)\big)\,\mathrm{d}V \\ - \int_\Omega \mathbf{F}\cdot k_0^2\epsilon_r\mathbf{E}\,\mathrm{d}V = -\mathrm{i}k_0Z_0\int_\Omega \mathbf{F}\cdot\mathbf{J}\,\mathrm{d}V.\end{aligned} \tag{3.5.3}$$

Invoking the divergence theorem,

$$\int_\Omega \nabla\cdot\mathbf{A}\,\mathrm{d}V = \oint_{\partial\Omega}\mathbf{A}\cdot\hat{\mathbf{n}}\,\mathrm{d}S,$$

and the identity $(\mathbf{A}\times\mathbf{B})\cdot\hat{\mathbf{n}} = -\mathbf{A}\cdot(\hat{\mathbf{n}}\times\mathbf{B})$ yields

$$\begin{aligned}\int_\Omega \big((\nabla\times\mathbf{F})\cdot(\mu_r^{-1}\nabla\times\mathbf{E}) - k_0^2\epsilon_r\mathbf{F}\cdot\mathbf{E}\big)\,\mathrm{d}V \\ + \oint_{\partial\Omega}\mathbf{F}\cdot\big(\hat{\mathbf{n}}\times\mu_r^{-1}\nabla\times\mathbf{E}\big)\,\mathrm{d}S = -\mathrm{i}k_0Z_0\int_\Omega\mathbf{F}\cdot\mathbf{J}\,\mathrm{d}V,\end{aligned} \tag{3.5.4}$$

where $\hat{\mathbf{n}}$ is the outward unit normal to the surface $\partial\Omega$. On substituting for the curl of the magnetic field from equation (3.2.6a), the weak problem is stated as: find $\mathbf{E}\in X$ such that

$$\begin{aligned}\int_\Omega \Big((\nabla\times\mathbf{F})\cdot(\mu_r^{-1}\nabla\times\hat{\mathbf{E}}) - k_0^2\epsilon_r\mathbf{F}\cdot\hat{\mathbf{E}}\Big)\,\mathrm{d}V =& ik_0Z_0\oint_{\partial\Omega}\mathbf{F}\cdot(\hat{\mathbf{n}}\times\hat{\mathbf{H}})\,\mathrm{d}S \\ &- ik_0Z_0\int_\Omega\mathbf{F}\cdot\mathbf{J}\,\mathrm{d}V\end{aligned} \tag{3.5.5}$$

for all $\mathbf{F}\in Y$. The sets X and Y are

$$X = \big\{\mathbf{u}\in H^1_{\text{curl}}(\Omega) : \hat{\mathbf{n}}\times\mathbf{u} = \mathbf{f} \text{ on } \Gamma_D,\ \nabla\cdot(\epsilon_r\mathbf{u}) = 0\big\} \tag{3.5.6a}$$

$$Y = \big\{\mathbf{u}\in H^1_{\text{curl}}(\Omega) : \hat{\mathbf{n}}\times\mathbf{u} = \mathbf{0} \text{ on } \Gamma_D,\ \nabla\cdot(\epsilon_r\mathbf{u}) = 0\big\}, \tag{3.5.6b}$$

enforcing the divergence condition and the Dirichlet boundary condition, while the Neuman boundary condition is directly substituted into the right-hand side of (3.5.5). The divergence condition can be enforced as a Lagrange multiplier, but it can be shown that for $\omega \gg 0$ the multiplier is equal to 0 [96]. If a solution solves the weak problem (3.5.5) with (3.5.6), it is also a solution to the strong problem (3.5.1).

3.6 Green's functions

3.6.1 Scalar Green's functions

In three dimensions, the free-space Green's function for the scalar Helmholtz equation $G_0(\mathbf{r},\mathbf{r}')$ satisfies

$$\left(\nabla^2 + k^2\right) G_0(\mathbf{r},\mathbf{r}') = -\delta(\mathbf{r}-\mathbf{r}'), \tag{3.6.1}$$

and decays as $R = |\mathbf{r}-\mathbf{r}'|$ tends to infinity, such that it satisfies the Sommerfeld radiation condition

$$\lim_{R\to\infty} R\left(\frac{\partial G_0}{\partial R} - \mathrm{i}kG_0\right) = 0. \tag{3.6.2}$$

Following [166], we make the change of variables $\mathbf{r}_1 = \mathbf{r}-\mathbf{r}'$, so that $G_0(\mathbf{r}_1, 0)$ satisfies

$$\frac{1}{r_1^2}\frac{d}{dr_1}\left[r_1^2\frac{dG_0(\mathbf{r}_1,0)}{r_1}\right] + k^2G_0(\mathbf{r}_1,0) = -\delta(\mathbf{r}_1), \tag{3.6.3}$$

where $r_1 = |\mathbf{r}_1|$. For $r_1 \neq 0$, (3.6.3) can be written as

$$\frac{d^2[r_1G_0(\mathbf{r}_1,0)]}{dr_1^2} + k^2r_1G_0(\mathbf{r}_1,0) = 0, \tag{3.6.4}$$

which (with the radiation condition) has the solution

$$r_1G_0(\mathbf{r}_1,0) = A\mathrm{e}^{-ikr_1}. \tag{3.6.5}$$

To determine the coefficient A, we substitute (3.6.5) into equation (3.6.3) and integrate over the volume of a sphere of radius ε centred at $\mathbf{r}_1 = \mathbf{0}$. Since for any volume $\mathcal{V}$ enclosing $\mathbf{r}_1$ we have

$$\iiint_{\mathcal{V}} \delta(\mathbf{r}_1)\,\mathrm{d}V = 1, \tag{3.6.6}$$

taking the limit $\epsilon \to 0$ we find that $4\pi A = 1$, and so

$$G_0(\mathbf{r}_1,0) = \frac{\mathrm{e}^{-\mathrm{i}kr_1}}{4\pi r_1}, \qquad \text{or} \qquad G_0(\mathbf{r},\mathbf{r}') = \frac{\mathrm{e}^{-\mathrm{i}k|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}. \tag{3.6.7}$$

In the far field, we have the limit

$$\lim_{r\to\infty} G_0(\mathbf{r},\mathbf{r}') = \frac{e^{-\mathrm{i}kr}}{4\pi r}e^{\mathrm{i}k\mathbf{r}\cdot\hat{\mathbf{r}}'}, \tag{3.6.8}$$

where $r = |\mathbf{r}|$ and $\hat{\mathbf{r}}' = \mathbf{r}'/|\mathbf{r}'|$.

Following the same procedure, in two dimensions we have

$$G_{0,\mathrm{2D}}(\mathbf{r},\mathbf{r}') = \frac{H_0^{(2)}(k|\mathbf{r}-\mathbf{r}'|)}{4\mathrm{i}}, \tag{3.6.9}$$

where $H_0^{(2)}$ is the zeroth-order Hankel function of the second kind.

The gradient of the 3D scalar Green's function is given by

$$\nabla G_0(\mathbf{r},\mathbf{r}') = -\nabla' G_0(\mathbf{r},\mathbf{r}') = -\left(\mathrm{i}k_0 + \frac{1}{R}\right) G_0(\mathbf{r},\mathbf{r}')\hat{\mathbf{R}}, \tag{3.6.10}$$

where $\hat{\mathbf{R}} = (\mathbf{r}-\mathbf{r}')/|\mathbf{r}-\mathbf{r}'|$, and ∇ and ∇' are the gradient operators with respect to $\mathbf{r}$ and $\mathbf{r}'$, respectively. Further, the Hessian of the scalar Green's function is given by

$$\begin{aligned}\nabla\nabla G_0(\mathbf{r},\mathbf{r}') = \nabla'\nabla' G_0(\mathbf{r},\mathbf{r}') =& \hat{\mathbf{R}}\hat{\mathbf{R}}\left[\frac{1}{R^2} + \left(\mathrm{i}k_0 + \frac{1}{R}\right)^2\right] G_0(\mathbf{r},\mathbf{r}') \\ &- (\overline{\overline{\mathrm{I}}} - \hat{\mathbf{R}}\hat{\mathbf{R}})\left(\mathrm{i}k_0 + \frac{1}{R}\right)\frac{G_0(\mathbf{r},\mathbf{r}')}{R},\end{aligned} \tag{3.6.11}$$

where $\overline{\overline{\mathrm{I}}}$ is the identity dyadic (i.e. the 3 × 3 identity matrix).

3.6.2 Dyadic Green's functions

Consider the electric and magnetic fields produced by an electric current source $\mathbf{J}$ in free space. The electric field satisfies the vector wave equation

$$\nabla\times\nabla\times\mathbf{E}(\mathbf{r}) - k_0^2\mathbf{E} = -\mathrm{i}\omega\mu_0\mathbf{J}. \tag{3.6.12}$$

The free space dyadic Green's function of the electric type $\overline{\overline{\mathrm{G}}}$ relates the field $\mathbf{E}$ and $\mathbf{J}$ by

$$\mathbf{E}(\mathbf{r}) = -\mathrm{i}\omega\mu_0 \iiint_{\mathcal{V}} \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,\mathrm{d}V', \tag{3.6.13}$$

where $\mathcal{V}$ is the support of the current $\mathbf{J}(\mathbf{r})$. Following [166], integrating (3.6.12) over $\mathcal{V}$ and applying (3.6.13), we have that

$$\begin{aligned}\iiint_{\mathcal{V}} \nabla\times\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}') - k^2\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,\mathrm{d}V'& \\ = \iiint_{\mathcal{V}} \overline{\overline{\mathrm{I}}}\delta(\mathbf{r}-\mathbf{r}')\cdot\mathbf{J}(\mathbf{r})\,\mathrm{d}V',&\end{aligned} \tag{3.6.14}$$

which for arbitrary $\mathbf{J}(\mathbf{r})$ can be satisfied only if

$$\nabla\times\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') - k_0^2\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') = \overline{\overline{\mathrm{I}}}\delta(\mathbf{r}-\mathbf{r}'). \tag{3.6.15}$$

Taking the (posterior) scalar product of (3.6.15) with an arbitrary constant vector $\mathbf{p}$, we have

$$\nabla\times\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{p} - k_0^2\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{p} = \mathbf{p}\delta(\mathbf{r}-\mathbf{r}'). \tag{3.6.16}$$

By the vector identity $\nabla \times \nabla \times \mathbf{f} \equiv \nabla(\nabla \cdot \mathbf{f}) - \nabla^2 \mathbf{f}$, we have

$$\nabla^2 \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p} + k_0^2 \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p} = \nabla \left[\nabla \cdot \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p} \right] - \mathbf{p}\delta(\mathbf{r} - \mathbf{r}'). \tag{3.6.17}$$

Taking the divergence of (3.6.17),

$$\nabla \cdot \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p} = -\frac{1}{k_0^2} \nabla \cdot [\mathbf{p}\delta(\mathbf{r} - \mathbf{r}')], \tag{3.6.18}$$

since $\nabla \cdot (\nabla \times \mathbf{f}) \equiv 0$. Substituting (3.6.18) into (3.6.17), we find

$$\nabla^2 \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p} + k_0^2 \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p} = -\left(1 + \frac{\nabla\nabla \cdot}{k_0^2} \right) [\mathbf{p}\delta(\mathbf{r} - \mathbf{r}')]. \tag{3.6.19}$$

By making use of the scalar Green's function, which solves (3.6.1), we find that

$$\overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p} = \left(1 + \frac{\nabla\nabla \cdot}{k_0^2} \right) [\mathbf{p} G_0(\mathbf{r}, \mathbf{r}')] = \left(\overline{\overline{\mathrm{I}}} + \frac{\nabla\nabla}{k_0^2} \right) G_0(\mathbf{r}, \mathbf{r}') \cdot \mathbf{p}. \tag{3.6.20}$$

Since $\mathbf{p}$ is an arbitrary vector, we see the free space dyadic Green's function is given by

$$\overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') = \left(\overline{\overline{\mathrm{I}}} + \frac{\nabla\nabla}{k_0^2} \right) G_0(\mathbf{r}, \mathbf{r}'). \tag{3.6.21}$$

Following a similar procedure, the free space dyadic Green's function of the magnetic type can be shown to be given by [166, pp 60]

$$\begin{aligned} \overline{\overline{\mathrm{G}}}_m(\hat{\mathbf{r}}, \hat{\mathbf{r}}') =& \nabla \times \left[\overline{\overline{\mathrm{I}}} G_0(\hat{\mathbf{r}}, \hat{\mathbf{r}}') \right] \\ =& [\nabla G_0(\hat{\mathbf{r}}, \hat{\mathbf{r}}')] \times \overline{\overline{\mathrm{I}}}. \end{aligned} \tag{3.6.22}$$

From Maxwell's equations, it is clear that the electric and magnetic dyadic Green's functions are related by

$$\nabla \times \overline{\overline{\mathrm{G}}} = \overline{\overline{\mathrm{G}}}_m \tag{3.6.23a}$$

$$\nabla \times \overline{\overline{\mathrm{G}}}_m = \overline{\overline{\mathrm{I}}}\delta(\mathbf{r} - \mathbf{r}') + k_0^2 \overline{\overline{\mathrm{G}}}. \tag{3.6.23b}$$

which tells us that the curl of $\overline{\overline{\mathrm{G}}}$ is given by

$$\nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') = \nabla \times \left[\overline{\overline{\mathrm{I}}} G_0(\mathbf{r}, \mathbf{r}') \right] = [\nabla G_0(\mathbf{r}, \mathbf{r}')] \times \overline{\overline{\mathrm{I}}}. \tag{3.6.24}$$

The free-space dyadic Green's function of the electric type is symmetric, and $\mathbf{r}$ and $\mathbf{r}'$ are interchangeable,

$$\overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') = \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}) = \left[\overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \right]^T = \left[\overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}) \right]^T, \tag{3.6.25}$$

whereas the free-space dyadic Green's function of the magnetic type is antisymmetric,

$$\overline{\overline{\mathrm{G}}}_m(\mathbf{r}, \mathbf{r}') = \overline{\overline{\mathrm{G}}}_m(\mathbf{r}', \mathbf{r}) = \left[\overline{\overline{\mathrm{G}}}_m(\mathbf{r}, \mathbf{r}') \right]^T = \left[\overline{\overline{\mathrm{G}}}_m(\mathbf{r}', \mathbf{r}) \right]^T. \tag{3.6.26}$$

3.7 Surface equivalence principle

The surface equivalence states that the fields exterior (interior) to a (possibly fictitious) surface $\mathcal{S}$ may be exactly represented by equivalent electric and magnetic currents, $\mathbf{J}_s$ and $\mathbf{M}_s$, placed on the surface and allowed to radiate into the external (internal) region - hence why it was useful to include the fictitious magnetic current in our formulation of Maxwell's equations (3.2.6). Love's equivalence principle [102], [186, pp9-10], states that these equivalent currents are given in terms of the total exterior fields, $\mathbf{E}^{\text{ext}}$ and $\mathbf{H}^{\text{ext}}$, as

$$\begin{aligned} \hat{\mathbf{n}} \times \mathbf{H}^{\text{ext}} =& \mathbf{J}_s \\ \mathbf{E}^{\text{ext}} \times \hat{\mathbf{n}} =& \mathbf{M}_s, \end{aligned} \tag{3.7.1}$$

where the interior fields are assumed to be zero. The radiated fields due to the equivalent currents are given by the integral equations [134, 186]

$$\begin{aligned} \mathbf{E}^{\text{ext}}(\mathbf{r}) =& \oint_{\mathcal{S}} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{M}_s(\mathbf{r}') \,\mathrm{d}S' \\ &+ \mathrm{i}k_0 Z_0 \oint_{\mathcal{S}} \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}_s(\mathbf{r}') \,\mathrm{d}S' \end{aligned} \tag{3.7.2a}$$

$$\begin{aligned} \mathbf{H}^{\text{ext}} =& -\oint_{\mathcal{S}} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}_s(\mathbf{r}') \,\mathrm{d}S' \\ &+ \mathrm{i}k_0 Y_0 \oint_{\mathcal{S}} \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{M}_s(\mathbf{r}') \,\mathrm{d}S'. \end{aligned} \tag{3.7.2b}$$

In equations (3.7.2), we have assumed that the surface $\mathcal{S}$ is radiating into free space with constant permittivity and permeability ϵ_0 and μ_0. Where they are radiating into a homogeneous region of permittivity and permeability ϵ and μ, the values k_0, Z_0 and Y_0 should be replaced by $k = \epsilon k_0$, $Z = \sqrt{\mu/\epsilon}$ and $Y = \sqrt{\epsilon/\mu}$, the wave number, intrinsic impedance and intrinsic admittance, respectively. In proceeding chapters, we may write equations (3.7.2) as

$$\mathbf{E}^{\text{ext}} = K(\mathbf{M}_s) + Z_0 L(\mathbf{J}_s), \tag{3.7.3a}$$

$$\mathbf{H}^{\text{ext}} = -K(\mathbf{J}_s) + Y_0 L(\mathbf{M}_s), \tag{3.7.3b}$$

where the integral operators K and L are defined by

$$K(\mathbf{X}) = \oint_S \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{X} \,\mathrm{d}S \tag{3.7.4a}$$

$$L(\mathbf{X}) = \mathrm{i}k_0 \oint_S \overline{\overline{\mathrm{G}}} \cdot \mathbf{X} \,\mathrm{d}S. \tag{3.7.4b}$$

Poggio and Miller [134, pp 164] also note that when the surface S is placed within an infinite medium and subjected to incident fields $\mathbf{E}^{\text{in}}$, $\mathbf{H}^{\text{in}}$, in order to satisfy the radiation condition at infinity the contribution from the integral over S must be due entirely from sources outside of S. In other words, denoting $\mathbf{E}^{\text{ext}} = \mathbf{E}^{\text{in}} + \mathbf{E}^{\text{sc}}$, $\mathbf{H}^{\text{ext}} = \mathbf{H}^{\text{in}} + \mathbf{H}^{\text{sc}}$, where the superscript sc denotes the parts of the field scattered by the dielectric enclosed by S,

$$\mathbf{E}^{\text{ext}} = \mathbf{E}^{\text{in}} + K(\mathbf{M}_s) + Z_0 L(\mathbf{J}_s), \tag{3.7.5a}$$

$$\mathbf{H}^{\text{ext}} = \mathbf{H}^{\text{in}} - K(\mathbf{J}_s) + Y_0 L(\mathbf{M}_s), \tag{3.7.5b}$$

which is the same as (3.7.2) with the addition of the forcing terms.

Let us briefly examine how an electric field is produced by the two integral operators K and L. Expanding the dyadic Green's function to write L as

$$L(\mathbf{J}_s) = \mathrm{i}k_0 \oint_{\mathcal{S}} \left[\mathbf{J}_s(\mathbf{r}') + \frac{1}{k_0^2} \nabla\nabla' \cdot \mathbf{J}_s(\mathbf{r}') \right] G_0(\mathbf{r}, \mathbf{r}') \, \mathrm{d}S', \tag{3.7.6}$$

it is clear that L produces an electric field by two mechanisms [152]. The first term is induction, by which a current first produces a magnetic field, and a time-varying magnetic field in turn produces an electric one. The second term gives rise to an electric field by charge accumulation in the current. The integral $K(\mathbf{M}_s)$ gives rise to an electric field mainly through induction, but this induction term will be partially orthogonal to that produced by $L(\mathbf{J}_s)$ [152]. By duality, the same can be concluded for the magnetic fields.

One can use the far field expression of the free space Green's function (3.6.8), together with the curl of the dyadic Green's function (3.6.24), to obtain the far field expressions of the radiated currents as[1] [186, p 12]

$$\mathbf{E}^{\text{ext}}(\mathbf{r}) = \mathrm{i}k_0 \frac{\mathrm{e}^{-\mathrm{i}k_0 r}}{4\pi r} \oint_{\mathcal{S}} \left[\mathbf{M}_s(\mathbf{r}') \times \hat{\mathbf{r}} - Z_0 \hat{\mathbf{r}} \times (\hat{\mathbf{r}} \times \mathbf{J}_s(\mathbf{r}')) \right] \mathrm{e}^{\mathrm{i}k_0 \mathbf{r} \cdot \hat{\mathbf{r}}'} \, \mathrm{d}S' + \mathcal{O}\left(\frac{1}{r^2}\right), \tag{3.7.7a}$$

$$\mathbf{H}^{\text{ext}}(\mathbf{r}) = \mathrm{i}k_0 \frac{\mathrm{e}^{-\mathrm{i}k_0 r}}{4\pi r} \oint_{\mathcal{S}} \left[\hat{\mathbf{r}} \times \mathbf{J}_s(\mathbf{r}') - Y_0 \hat{\mathbf{r}} \times (\hat{\mathbf{r}} \times \mathbf{M}(\mathbf{r}')) \right] \mathrm{e}^{\mathrm{i}k_0 \mathbf{r} \cdot \hat{\mathbf{r}}'} \, \mathrm{d}S' + \mathcal{O}\left(\frac{1}{r^2}\right), \tag{3.7.7b}$$

as $r \to \infty$.

If the radiated fields are to satisfy some given boundary conditions to the surface $\mathcal{S}$, we can use the appropriate Green's function (if available) in the surface equivalence

[1] Note that in many texts there may be a sign difference due to the choice of Green's functions.

formulation. If $\mathbf{E}^{\text{ext}}$ is to satisfy Dirichlet boundary conditions on $\mathcal{S}$, one can use the electric dyadic Green's function of the first kind, $\overline{\overline{\mathrm{G}}}_1$, which is defined to satisfy the Dirichlet condition [166, p 66]

$$\hat{\mathbf{n}} \times \overline{\overline{\mathrm{G}}}_1 = 0 \qquad \text{on } \mathcal{S}, \tag{3.7.8}$$

and reduce the surface equivalence formulation to

$$\mathbf{E}^{\text{ext}}(\mathbf{r}) = \oint_{\mathcal{S}} \nabla \times \overline{\overline{\mathrm{G}}}_1(\mathbf{r}, \mathbf{r}') \cdot \mathbf{M}_s(\mathbf{r}')\, \mathrm{d}S', \tag{3.7.9a}$$

$$\mathbf{H}^{\text{ext}}(\mathbf{r}) = -\oint_{\mathcal{S}} \nabla \times \overline{\overline{\mathrm{G}}}_1(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}_s(\mathbf{r}')\, \mathrm{d}S', \tag{3.7.9b}$$

where we have used the fact that $\overline{\overline{\mathrm{G}}}_1$ has no tangential component on $\mathcal{S}$ to negate the second terms in the integrand.

If the radiated electric field is to satisfy Neumann boundary conditions, one can use the electric dyadic Green's function of the second kind, $\overline{\overline{\mathrm{G}}}_2$, which is defined to satisfy the Neumann condition [166, p 67]

$$\hat{\mathbf{n}} \times \nabla \times \overline{\overline{\mathrm{G}}}_2 = 0 \qquad \text{on } \mathcal{S}, \tag{3.7.10}$$

resulting in the expressions for the radiated fields reduces to

$$\mathbf{E}^{\text{ext}}(\mathbf{r}) = \mathrm{i}k_0 Z_0 \oint_{\mathcal{S}} \overline{\overline{\mathrm{G}}}_2(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}_s(\mathbf{r}')\, \mathrm{d}S', \tag{3.7.11a}$$

$$\mathbf{H}^{\text{ext}}(\mathbf{r}) = \mathrm{i}k_0 Z_0 \oint_{\mathcal{S}} \overline{\overline{\mathrm{G}}}_2(\mathbf{r}, \mathbf{r}') \cdot \mathbf{M}_s(\mathbf{r}')\, \mathrm{d}S'. \tag{3.7.11b}$$

Unlike (3.7.2), equations (3.7.9) and (3.7.11) require only knowledge of either the electric or magnetic current sources on $\mathcal{S}$ to calculate the respective fields. However, construction of the first and second kind of dyadic Green's functions is more involved, particularly for more complex surfaces where they will most likely be unknown. Some forms are given by Tai [166].

3.8 Summary

In this chapter, we have covered the well known electromagnetic theory relevant to GPR, which will be referred to throughout this thesis. In Chapter 4, we will use the Helmholtz equation to describe scattering of GPR waves in the 2D approximation as the forward model for the full-wave inversion problem. Chapter 7 will make use of the dyadic Green's functions and the surface equivalence principle to describe scattering from small inclusions in the subsurface, and to examine sensitivity in the 3D full-wave inversion problem. Chapter 6 covers numerical simulation of GPR data in 3D, and will use both the variational form of the curl-curl wave equation from Section 3.5 as the basis of a finite-element approximation, as well as the surface equivalence principle to describe the transmission and reflection of electromagnetic waves from the ground surface.

Chapter 4

Full-Wave Inversion in 2D

4.1 Introduction

We wish to be able to perform full-wave inversion on GPR data to obtain quantitative information about objects in the subsurface, as well as account for multiple scattering and changes in the wave speed which would affect the accurate locating and characterising of objects. As discussed in Section 2.5, 2D full-wave inversion of GPR data is now a fairly well established research topic. However, 2D inversions make unrealistic assumptions about the physical reality of the wave scattering problem, and so 2D inversion of data from a 3D world may not always be adequate for one's needs. Particularly, landmines are 3D objects, so we will need to implement 3D inversion schemes for our purpose. Nevertheless, it is useful for us to begin experimenting with 2D inversions to gain incite in to solving our specific problem of landmine detection. Since the 2D assumptions made reduce the forward problem to the Helmholtz equation, our results can be applicable also to ultrasound imaging.

This chapter is used to illustrate the inversion procedure, and so we bring together all the implementational details and numerical recipe in one place for the reader. We will first describe the forward problem in 2 dimensions, which we solve via a second order finite-difference scheme for ease of prototyping. Thereafter, we discuss the various aspects of the inversion scheme, including suitable regularization strategies. We implement Total Variation regularisation (TV), and we believe this is the first implementation for FWI of GPR data. TV has been used to deblur (non FWI) GPR images as a post-processing method [130], and as well as for FWI of seismic data with the Helmholtz equation as the forward model [15], so this is only a minor novelty.

4.2 The forward problem in 2D

4.2.1 Finite difference approximation of the Helmholtz equation

For 2D inversions, we make the assumption that the subsurface is invariant in one coordinate direction, say $\hat{\mathbf{y}}$. This assumption is made not only about the parameters describing the subsurface, but also about sources which are therefore line sources. Under this assumption, waves will not reflect in or out of the plane that we are imaging in. From Section 3.4.2, the y component of the electric wave satisfies the equation

$$\left[\nabla\left(\frac{1}{\mu_r}\nabla\right) + \left(k_0^2\epsilon_r - \mathrm{i}k_0 Z_0\sigma\right)\right] E_y = -\mathrm{i}k_0 Z_0 J_y, \tag{4.2.1}$$

which is recognised as 2D Helmholtz equation

$$\left[\nabla\left(\frac{1}{\mu_r}\nabla\right) + k^2\right] E_y = S, \tag{4.2.2}$$

where the complex wavenumber $k^2(\mathbf{x}) = k_0^2\epsilon_r(\mathbf{x}) - ik_0 Z_0\sigma(\mathbf{x})$. The y component of the wave decouples from the x and z components, since a $\hat{\mathbf{y}}$ polarized wave incident on an object invariant in the y direction will not change polarization state. The x and z components however remain coupled.

We have chosen to use a second-order finite difference method for the numerical solution of the Helmholtz equation, primarily for ease of prototyping. Assuming we discretise the subsurface into a regular grid with constant step length h, this is given by

$$-\frac{1}{h^2}(u_{m-1,n} + u_{m+1,n}) - \frac{1}{h^2}(u_{m,n-1} + u_{m,n+1}) + \left(\frac{4}{h^2}k_{m,n}^2\right) = S_{m,n}, \tag{4.2.3}$$

where $u_{m,n}$ is the discrete value of the component of electric field in the plane of homogeneity (e.g. E_y), or the pressure field if one considers ultrasound, and $k_{m,n}$ and $S_{m,n}$ are the discrete values of the wavenumber and source field, respectively.

Since we have an infinite domain, one must truncate the domain in order to numerically compute a solution to (4.2.2). The most straightforward way to do this is by using absorbing boundary conditions. First order accurate absorbing boundary conditions in 2D can be easily implemented by [21]

$$\frac{\partial u}{\partial \hat{\mathbf{n}}} - \mathrm{i}ku = 0 \tag{4.2.4}$$

where $\hat{\mathbf{n}}$ is the unit outward normal to the domain. It is further possibly to improve their accuracy by [21]

$$\frac{\partial u}{\partial \hat{\mathbf{n}}} - \mathrm{i}ku - \frac{\mathrm{i}}{2k}\frac{\partial^2 u}{\partial \hat{\mathbf{s}}^2} = 0 \tag{4.2.5}$$

on domain edges and

$$\frac{\partial u}{\partial \hat{\mathbf{s}}} = \frac{3}{2}\mathrm{i}ku \tag{4.2.6}$$

at the corners, where $\hat{\mathbf{s}}$ is a unit tangent vector to the domain. Equations (4.2.5) and (4.2.6) are second order accurate. To apply either the first or second order conditions, one uses their discrete approximation to substitute for $u_{m,n}$ in (4.2.3) which would lie one grid point outside of the domain. Alternatively, one could use perfectly matched layers [24] to truncate the domain, either as well as or instead of the absorbing boundary conditions.

The discretised system, coupled with the absorbing boundary conditions, is assembled as the linear system

$$A(\mathbf{k})\mathbf{u} = \mathbf{S}, \tag{4.2.7}$$

where the $A \in \mathbb{C}^{n^2 \times n^2}$ is a Hermitian highly sparse banded matrix. Solving the system (4.2.7) results in a discrete second-order accurate approximation to the solution to (4.2.2) (if the second-order absorbing boundary conditions have been used, else it will be first-order accurate).

4.2.2 Iterative solution of the linear system

For sufficiently large k, the system matrix A becomes indefinite due to it being the sum of a negative (∇^2) and positive (k^2) operator (i.e. it has both positive and negative eigenvalues) [57], and may have a large condition number which can result in large floating point errors in the solution to the system. The system (4.2.7) is traditionally solved via direct method such as Gaussian elimination, an $\mathcal{O}(n^3)$ operation for an $n \times n$ square matrix. Indeed, if we use MATLAB's `mldivide` or 'backslash' operator to solve the system if will perform some test on the form of A before deciding to calculate an

LU decomposition $A = LU$ (where L is a lower and U an upper triangular matrix), and solve the system $LU\mathbf{x} = \mathbf{b}$ via Gaussian elimination [169].

Calculating an LU decomposition of A may require an unacceptably large amount of storage, since it will not take account of the sparse banded nature of the matrix, thereby producing fill-in in the bandwidth of A (i.e., it will create non-zero entries where there were zeroes before, between the leading and furthest off-diagonals of A). Further, one may not need the machine-precision accuracy gained from using a direct method. This is particularly prevalent where one is solving an inverse problem with limited accuracy data: there is little point trying to simulate data with machine precision when one cannot record to this accuracy, and where modelling errors may be a greater source of inaccuracy in simulated data. We must also consider that errors in the forward model propagate into the derivative of the objective function when solving the inverse problem, since the derivative is a backprojection of data residuals, giving rise to another source of error on solution of a further linear system. We may then need to solve the forward problem with greater precision than the recorded data in order to obtain sufficiently accurate derivative information.

Instead of using a direct method, it is more appropriate to use an iterative one. Such a method builds up a sequence of improving approximations, $\mathbf{x}_k$, to the linear system. For large sparse banded matrices, iterative methods can take advantage of the fact that a matrix multiplication $A\mathbf{x}$ takes $m \times n$ operations, where m is the number of bands and n the size of the matrix. This can be done by taking the k^{th} approximation to be

$$x_k \in \operatorname{span}\left\{\mathbf{b}, A\mathbf{b}, \ldots, A^{k-1}\mathbf{b}\right\}, \quad k = 1, 2, \ldots, \tag{4.2.8}$$

which is the Krylov subspace of A. The vectors $\mathbf{q}_i = A^i\mathbf{b}$ are referred to as the Arnoldi vectors. Iterative methods based on the Krylov subspace include conjugate gradients (CG) for Hermitian positive definite A, minimal residual (MINRES) for Hsermitian indefinite A, and the generalized minimal residual (GMRES) algorithm for general non-Hermitian matrices, as well as the bi-conjugate gradient (BiCG) and bi-conjugate gradient stabilized (BiCGSTAB) algorithms. Broadly speaking, these methods differ in how they find the $\mathbf{x}_k$ which minimises $A\mathbf{x}_k - \mathbf{b}$ [64].

One must ask how good an approximate solution is contained in (4.2.8). If it does not contain a good solution for any moderate k, one must use instead a preconditioned method, which solves the modified problem

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}. \tag{4.2.9}$$

Then, we instead generate the subspace

$$x_k \in \operatorname{span}\left\{M^{-1}\mathbf{b}, (M^{-1}A)M^{-1}\mathbf{b}, \ldots, (M^{-1}A)^{k-1}M^{-1}\mathbf{b}\right\}, \quad k = 1, 2, \ldots. \tag{4.2.10}$$

A good choice of preconditioner M should be one such that the linear system (4.2.9) is much easier to solve, broadly speaking resulting in the space (4.2.10) containing a good approximation after a moderate number of iterations. Generally this means M is a good approximation to A, such that $M^{-1}A$ is close (in some sense) to the identity. If $M = A$ though, we gain nothing from preconditioning since we will have solved the linear system directly.

The simplest form of preconditioner is to take M as the diagonal of S (the Jacobi preconditioner). This has a minimal memory footprint and it is simple to find M^{-1}, but it may be insufficient where A is not sufficiently diagonally dominant. This diagonal preconditioner can be generalised by forming M from blocks along the diagonal of A. For denser matrices than the 5-point stencil finite difference matrix, forming M from the elements A_{ij} where indices i and j lie close to one another can also be effective. Methods based on an incomplete factorization of A, such as incomplete LU (ILU) factorisation, can prove to be far more robust since they are much 'closer' to A. Their drawback is their additional storage requirement, as well as additional time to compute the decomposition. Multigrid methods are also possible, in which the problem is solved in a hierarchy of increasingly refined grids. However, since a sufficiently refined grid is required to prevent large numerical dispersion, it may not be possible to solve the problem on a much coarser grid which limits their applicability.

For the discretised Helmholtz system, with A close to indefinite the Krylov subspace may poorly represent the solution. This is because the solution is large throughout the domain, but with a 5 finite difference point stencil all the lower order Arnoldi basis vectors have only a few nonzero entries [57]. A pre-conditioned method is therefore necessary.

One appropriate method is BiCGSTAB [177] with a shifted-Laplace type preconditioner [2, 54, 141, 189, 199]. The shifted-Laplace the pre-conditioner M is given by

$$M(\mathbf{k}) = A(\hat{\mathbf{k}}), \qquad \hat{\mathbf{k}} = \mathbf{k}(\beta_r + i\beta_c), \tag{4.2.11}$$

where β_r is close to 1 and $0 < \beta_c \leq 1$. At $\beta_r = 1$ and $\beta_c = 0$, the shifted-Laplace pre-conditioner reduces to the system matrix and applying M^{-1} will solve the system (4.2.7). This is not efficient since it would bypass the iterative method. For $\beta_c > 0$ M is a damped system, and so the solution $M^{-1}\mathbf{b}$ has far more local perturbations than $A^{-1}\mathbf{b}$. $0 \ll \beta_r < 1$ can also be beneficial for large values k_0, as M will have a lower condition number but similar eigenvalue spectrum. In general, for our relatively small-scale problem we have found the values $\beta_r = 1$, $\beta_c = 0.2$ to yield convergence at an acceptable rate. Figure 4.1 shows the convergence of the finite-difference solution of the Helmholtz equation with $k = 15$ against the analytic Green's function, (3.6.9), using first order absorbing boundary conditions. Convergence is seen to be super-linear, not quadratic, owing to the first order absorbing conditions. At higher wavenumbers, an initial oscillatory period would be observed in the convergence plot before the solution begins to converge with grid refinement. This is due to high numerical dispersion caused by insufficient grid points (see, for example, Ainswork [3]).

4.3 Inversion procedure

4.3.1 Introduction

In this section we describe the full-wave inversion procedure, that is to solve numerically

$$\mathbf{m}_{\text{inv}} = \underset{\mathbf{m}}{\operatorname{argmin}} \frac{1}{2}\|\mathcal{F}(\mathbf{m}) - \mathbf{d}\| + \lambda R(\mathbf{m}), \tag{4.3.1}$$

giving the full numerical recipe so that the reader has all the required information for them to implement 2D FWI in one location. Following the discussions on inverse problems and iterative optimisation in Chapter 2, we first describe our chosen optimisation procedure (l-BFGS) and reasons for its selection. We also discuss implementational aspects of the iterative procedure, including stopping criteria and line-searching, in the context of the inverse problem. We then discuss a suitable regularisation strategy for landmine detection, and its implementation in the two-dimensional discretisation of the subsurface.

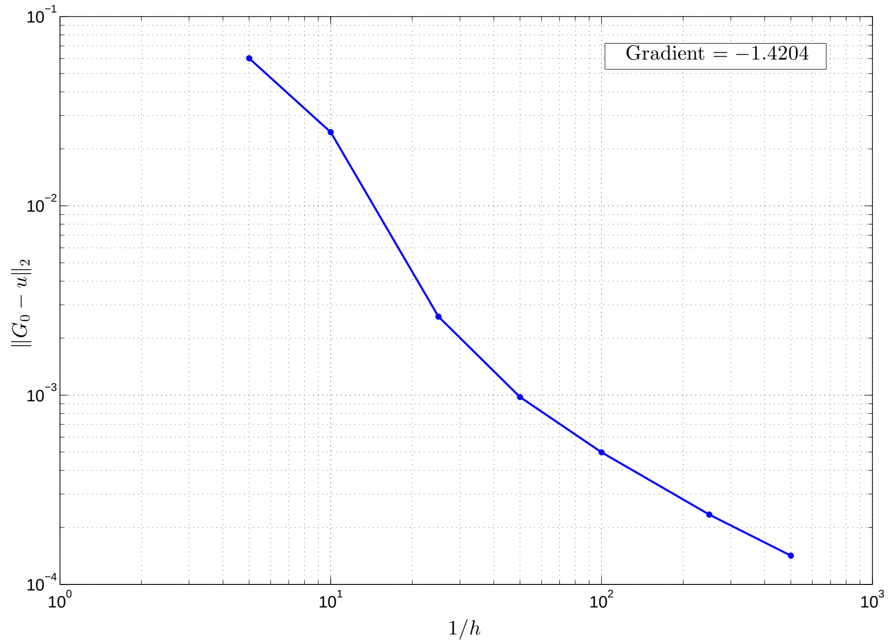


Figure 4.1: Convergence plot of the finite-difference solution of the Helmholtz equation with $k = 15$, for a 1×2 m grid with $\mathcal{O}(1)$ absorbing boundary conditions, compared to the analytic Greens function. The line of best fit has a gradient of −1.4204, showing super-linear convergence. Convergence is not quadratic due to the $\mathcal{O}(1)$ absorbing boundary conditions.

To highlight the affect of the various parameters or features of the inversion process, we have used a common test problem throughout this section containing two square scatterers (or rather, cuboids infinite in the $\hat{\mathbf{y}}$ direction), as shown in Figure 4.2. Unless stated otherwise, data was simulated at 100 frequencies between 1 and 3 GHz for a 1 source 3 receiver array with maximum source-receiver offset of 20 cm, at 16 source locations along the surface. The data then had Gaussian noise added to result in a signal-to-noise ratio of 20:1. The inversion was initialised at a homogeneous domain, used Total Variation regularisation (see section 4.3.6), was allowed to run for 20 iterations, and the objective function was weighted by $\frac{1}{\omega^2}$. When simulating data, the finite difference grid spacing was 1.43 mm, but when evaluating the objective function 2 mm was used. This strategy, in which course and fine grid points are not co-located (apart from at source locations), ensures we are not ignoring the effect of discretisation errors on the solution to the inverse problem by using the same simulation and inversion grid (this is often referred to as an ‘inverse crime’, see for example Kaipio and Somersalo [81]).

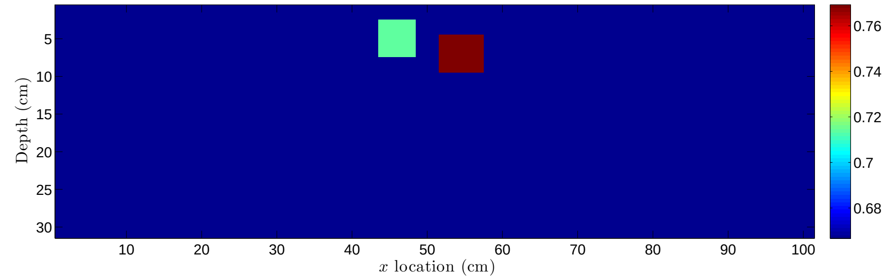


Figure 4.2: Test domain used to demonstrate the inversion process. Colour scale shows reciprocal of wavespeed $\frac{1}{c} \times 10^8$ s/m

4.3.2 Iterative solution of the full-wave problem

In Section 2.5.3, we gave an overview of several optimisation methods which can be suitable for solving the non-linear and large-scale optimisation problem of FWI: Newton; quasi-Newton; steepest descent; and non-linear conjugate gradient methods. These methods calculate a descent direction of the cost function $\mathcal{J}$ at each iteration based on a linear or quadratic model $\mathcal{M}_k(\mathbf{p}) \approx \mathcal{J}(\mathbf{x}_k)$, followed by a line-search in the descent direction to approximately minimise the cost function in this direction. The model $\mathcal{M}$ may not be a good approximation if we are far from the global minimiser (true solution). In Section 2.5.5, we also discussed the meaning of the Hessian matrix and its effect in the Newton descent direction: it incorporates non-linear effects such as first-order multiple scattering, and correctly rescales the descent direction to account for parameter illumination. Due to the importance of the Hessian matrix, we wish our optimisation method to include it. Thus we rule out steepest descent and conjugate gradient methods, leaving us with a choice of Newton, quasi-Newton and Newton-CG. Unfortunately, both the Hessian matrix itself and the Gauss-Newton approximation $H \approx J^T J$ are too large to store in full, or calculate in a reasonable time-frame, for a realistic sized full-wave problem. We therefore chose to use a quasi-Newton method, which approximates the Hessian matrix based on the finite differences in the gradient at each iteration.

One option is the BFGS method, as well as other methods in the Broyden Class [123, pp 149-152]. The BFGS update (2.5.10) requires the storage of a large dense Hessian approximation, which also rules the method out due to memory constraints.

Instead, we may use the limited memory BFGS method, l-BFGS, which stores a user-defined number m of vector pairs $\{\mathbf{s}_k, \mathbf{y}_k\}$, where

$$\mathbf{s}_k = \mathbf{m}_{k+1} - \mathbf{m}_k, \qquad \mathbf{y}_k = \nabla_{\mathbf{m}}\mathcal{J}_{k+1} - \nabla_{\mathbf{m}}\mathcal{J}_k, \tag{4.3.2}$$

and $\mathcal{J}_k := \mathcal{J}(\mathbf{m}_k)$. Repeated application of the BFGS formula (2.5.10) shows that at iteration k, the BFGS approximate Hessian satisfies

$$\begin{aligned} B_k = {} & \left(V_{k-1}^T \cdots V_{k-m}^T\right) B_{k-m} \left(V_{k-m} \cdots V_{k-1}\right) \\ & + \rho_{k-m} \left(V_{k-1}^T \cdots V_{k-m+1}^T\right) \mathbf{s}_{k-m}^T \mathbf{s}_{k-m}^T \left(V_{k-m+1} \cdots V_{k-1}\right) \\ & + \rho_{k-m+1} \left(V_{k-1}^T \cdots V_{k-m+2}^T\right) \mathbf{s}_{k-m+1}^T \mathbf{s}_{k-m+1}^T \left(V_{k-m+2} \cdots V_{k-1}\right) \\ & + \dots \\ & + \rho_{k-m+1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T, \end{aligned} \tag{4.3.3}$$

where B_{k-m} is the BFGS approximate inverse Hessian at iteration $k - m$, and

$$\rho_k = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}, \qquad V_k = I - \rho_k \mathbf{y}_k \mathbf{s}_k^T. \tag{4.3.4}$$

The l-BFGS approximate inverse Hessian is derived from (4.3.3) by choosing a new initial approximation B_k^0 in place of B_{k-m} in (4.3.3), thus neglecting contributions to the Hessian from more than m iterations previous. If B_k^0 is sufficiently sparse (for example, a diagonal matrix), then l-BFGS will have a vastly lower memory footprint then BFGS. (4.3.3) also prompts the efficient calculation of the product $B_k \nabla_{\mathbf{m}} \mathcal{J}_k$ given in Algorithm 1 by Nocedal and Wright [123, pp178].

Algorithm 1 l-BFGS two-loop recursion

```
$\mathbf{q} \leftarrow \nabla\mathcal{J}_k$;
for $i = k-1, k-2, \dots, k-m$ do
    $\alpha_i \leftarrow \rho_i \mathbf{s}_i^T \mathbf{q}$;
    $\mathbf{q} \leftarrow \mathbf{q} - \alpha_i \mathbf{y}_i$;
end for
$\mathbf{r} \leftarrow B_k^0 \mathbf{q}$;
for $i = k-m, \dots, k-1$ do
    $\beta \leftarrow \rho_i \mathbf{y}_i^T \mathbf{r}$;
    $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{s}_i(\alpha_i - \beta)$;
end for                                  ▷ end with result $\mathbf{r} = B_k \nabla\mathcal{J}_k$
```

l-BFGS requires an initial estimation of the inverse Hessian matrix B_k^0, which for computational simplicity is often chosen to be a multiple of the identity matrix. The first iteration of l-BFGS will then be the equivalent to steepest descent, and we can expect little reduction in our cost function. Since we are able to choose a different initial Hessian for each iteration, for iteration $k > 1$ it can be effective to choose

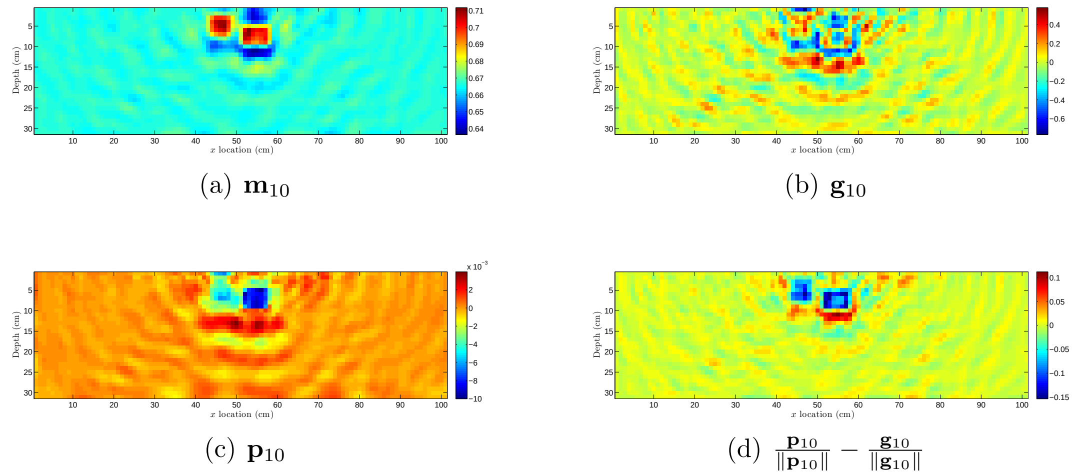
$$B_k^0 = \gamma_k I = \frac{\mathbf{S}_{k-1}^T \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^T \mathbf{y}_{k-1}} I, \tag{4.3.5}$$

which attempts to make the size of B_0 close to that of $\nabla^2 \mathcal{J}$ [123, pp143].

Algorithm 2 l-BFGS

```
Choose an $\mathbf{m}_0$, $m > 0$;
$k \leftarrow 0$;
while Convergence criteria not met do
    Choose $B_k^0$;                                  ▷ e.g. from (4.3.5)
    $\mathbf{p}_k \leftarrow -H_k \nabla \mathcal{J}_k$ from Algorithm 1;
    $\mathbf{m}_{k+1} \leftarrow \mathbf{m}_k + \alpha_k \mathbf{p}_k$, where $\alpha_k$ satisfies the (strong) Wolfe conditions;
    if $k > m$ then
        Discard the vector pair $\{\mathbf{s}_{k-m}, \mathbf{y}_{k-m}\}$ from storage;
    end if
    $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1}, \mathbf{x}_k$;
    $\mathbf{y}_k \leftarrow \nabla \mathcal{J}_{k+1} - \nabla \mathcal{J}_k$;
    $k \leftarrow k + 1$;
end while
```

The importance of the Hessian matrix in least-squares FWI, as discussed in Section 2.5.5, can be highlighted by comparing the l-BFGS descent direction to the gradient, as shown in Figure 4.3. Here we see the l-BFGS direction will add a contribution over the whole of the scattering objects and comparatively little contributions elsewhere; in contrast, the gradient term appears to only add contributions around the edge of the scatterers, with the (l_∞) size of artefacts almost as large as that of the scattering objects. This is easiest to see in the comparison between the normalised directions. The gradient is also very badly scaled, which may result in a line search requiring many objective evaluations in order to find a suitable steplength if a steepest descent method is used.



(a) $\mathbf{m}_{10}$

(b) $\mathbf{g}_{10}$

(c) $\mathbf{p}_{10}$

(d) $\frac{\mathbf{p}_{10}}{\|\mathbf{p}_{10}\|} - \frac{\mathbf{g}_{10}}{\|\mathbf{g}_{10}\|}$

Figure 4.3: Comparing the l-BFGS descent direction to the gradient after 10 iterations. a) shows the current iterate $\mathbf{m}_{10}$, b) the gradient $\mathbf{g}_{10}$, c) the l-BFGS descent direction $\mathbf{p}_{10}$, and d) the difference between normalised l-BFGS direction and normalised gradient.

4.3.3 Stopping conditions

The l-BFGS algorithm given in Algorithm 2 is stated without stopping criteria. We will generally want to stop when one of several things occurs:

$$\begin{aligned} \frac{\mathcal{J}_k}{\mathcal{J}_0} &\leq \text{tol}_{\mathcal{J}}, && (4.3.6a) \\ \nabla \mathcal{J}_k &\leq \text{tol}_g, && (4.3.6b) \\ \frac{\|\mathbf{m}_k - \mathbf{m}_{k-1}\|}{\|\mathbf{m}_k\|} &\leq \text{tol}_s, && (4.3.6c) \\ \mathcal{J}_{k-1} - \mathcal{J}_k &\leq \text{tol}_c. && (4.3.6d) \end{aligned}$$

The reasoning behind first two of these conditions is clear: if the gradient is sufficiently small we have reached a local minima, and if the cost value has reached a known minimum value we will not be able to reduce it further. $\text{tol}_{\mathcal{J}}$ should be set relative to the estimated noise level: given the accuracy of our data (and of our forward model), one should not try and approach this accuracy in the cost function, as we will be including noise in our solution. If the maximum estimated noise level is δ_ε, a suitable tolerance might be $\text{tol}_{\mathcal{J}} = 10 \times \delta_\varepsilon$ – i.e. we expect an order of accuracy in solution less, but other choices are possible.

(4.3.6c) identifies that the optimisation scheme has stagnated, and further iterations may simply be a waste of computational time. (4.3.6d) is similarly a sensible criteria, preventing excessively small change in cost reduction at large computational expense. It is guaranteed to be reached if $\mathcal{J}_k$ is strictly decreasing with k and bounded below. Both (4.3.6c) and (4.3.6d) can be implemented with a delay to prevent a premature stop, i.e. we stop only if the inequality holds for a given number of previous iterations. This can be useful if we expect the stagnation occurred due to a poor descent direction, or changing some other parameter (say, regularisation), which can be resolved in the next iteration.

We note that each tolerance is an additional regularisation parameter. Too small a tolerance and we over-solve the inverse problem, either fitting the noise in the data or simply making trivially small gains at large computational cost. Too large and we exit prematurely, far from the true solution and having only fitted the dominant parts of the data misfit. This is highlighted by Figure 4.4, comparing the iterate $\mathbf{m}_k$ at iterations 5, 15, 25, and 50, and by Figure 4.5, showing the relative residual, absolute error and relative change in solution for 50 iterations of l-BFGS.

From Figure 4.4 we see that at 5 and 15 iterations we are far from the true solution. With little difference between the iterates 25 and 50 though, for these final 25 iterations we are over-solving. This is confirmed by Figure 4.5, which shows that after 25 iterations there is little reduction in both the residual as well as the absolute error against the true solution. From Figure 4.5(c) we see that, in this example the stopping criteria $\mathrm{tol}_s = 10^{-4}$ would have been appropriate.

4.3.4 Line-searching

The l-BFGS algorithm requires a step length α_k at each iteration, to determine how far along each descent direction $\mathbf{p}_k$ one should travel to update the solution (i.e. a globalisation method). Often in FWI the step length is calculated via a linearisation [55, 132]

$$\alpha = \kappa \frac{\big(\mathcal{F}(\mathbf{m}_k + \kappa\mathbf{p}_k) - \mathcal{F}(\mathbf{m}_k)\big)^T \big(\mathcal{F}(\mathbf{m}_k) - \mathbf{d}_{\mathrm{obs}}\big)}{\big(\mathcal{F}(\mathbf{m}_k + \kappa\mathbf{p}_k) - \mathcal{F}(\mathbf{m}_k)\big)^T \big(\mathcal{F}(\mathbf{m}_k + \kappa\mathbf{p}_k) - \mathcal{F}(\mathbf{m}_k)\big)}, \tag{4.3.7}$$

where κ is chosen to be small enough such that linearity in the forward operator holds, but large enough to avoid numerical round-off error. For details see Appendix A.3. However, carrying out a line-search which satisfies the (strong) Wolfe

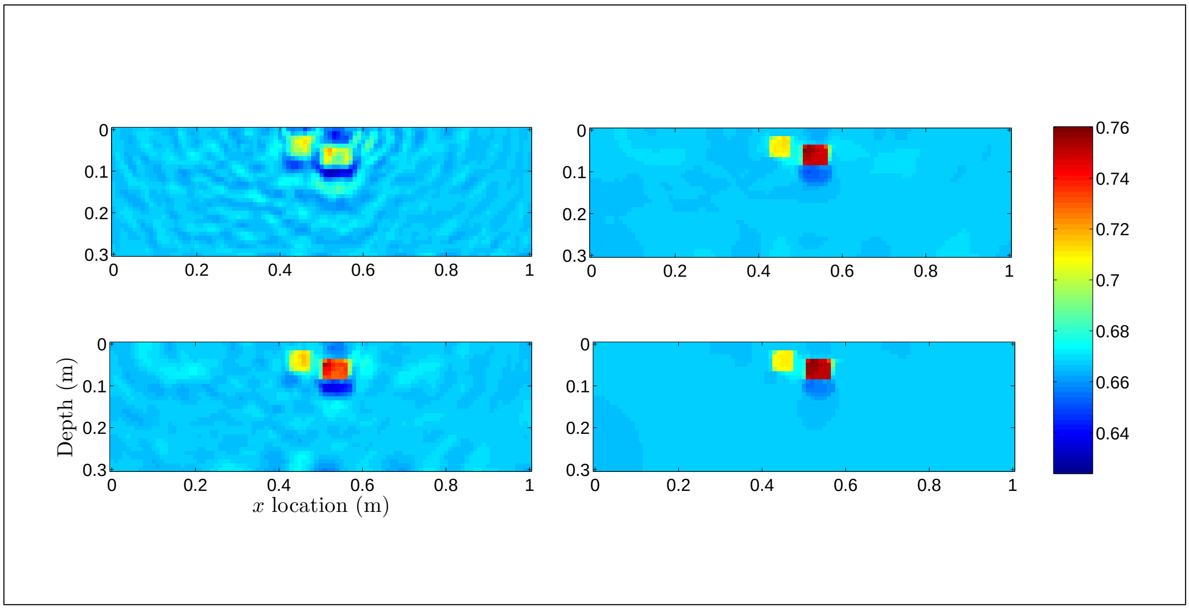


Figure 4.4: Reconstruction of the 2D test problem in Figure 4.2 at iterations 5 (top left), 15 (bottom left), 25 (top right) and 50 (bottom right)

conditions (2.5.15) may result in either faster or more stable convergence if the linearisation (4.3.7) is poor. Given a descent direction $\mathbf{p}_k$, defining $\phi(\alpha) := \mathcal{J}(\mathbf{m}_k + \alpha\mathbf{p}_k)$ the strong Wolfe conditions can be written as

$$\phi(\alpha) \leq \phi(0) + c_1\alpha\phi'(0) \tag{4.3.8a}$$

$$|\phi'(\alpha)| \leq c_2|\phi'(0)|. \tag{4.3.8b}$$

As mentioned in Section 2.5.3, (4.3.8a) ensures the objective function has decreased sufficiently, preventing us from testing larger steps when the decrease is poor; and (4.3.8a) ensures α lies close to a local minimum.

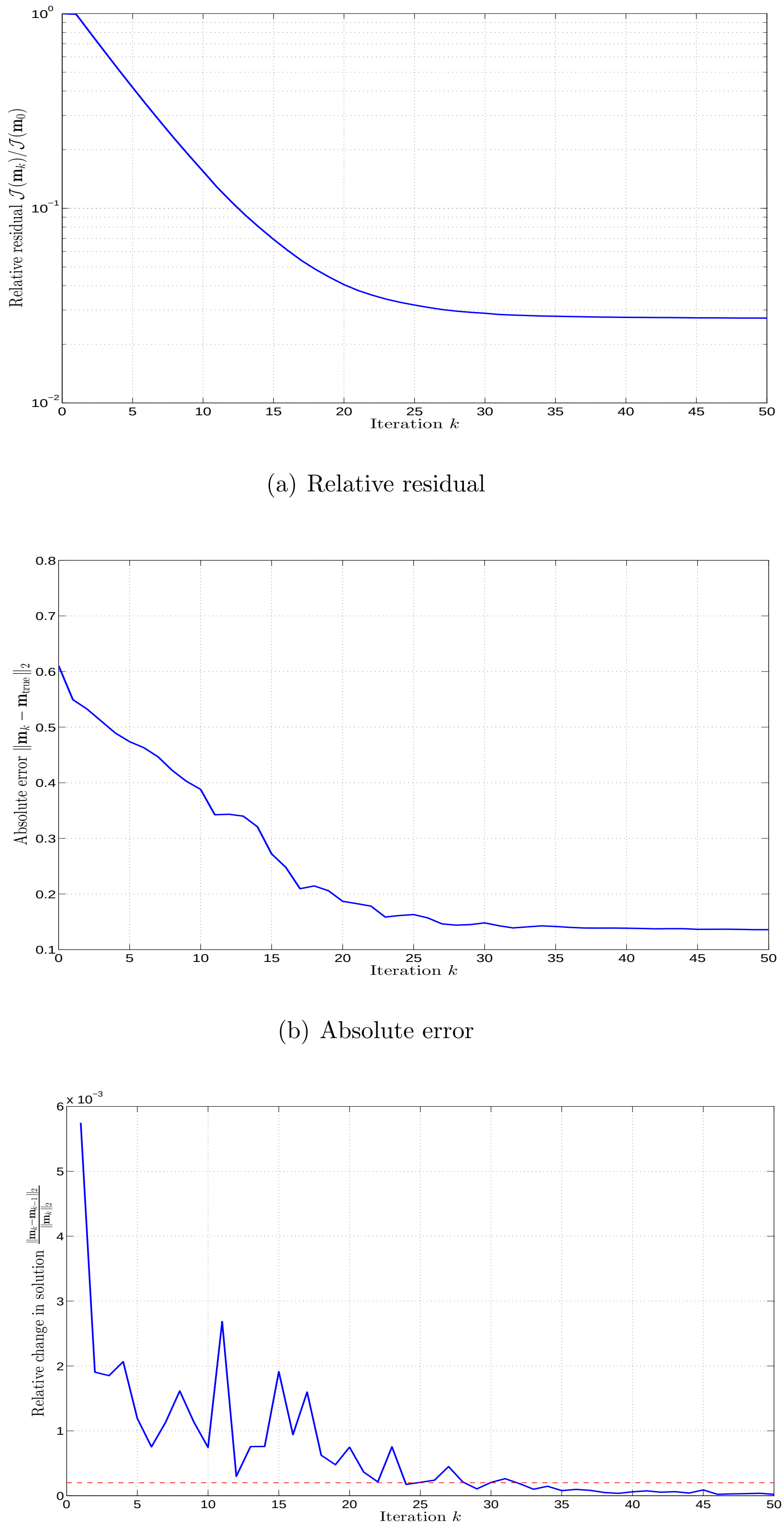
To understand why the curvature condition (4.3.8b) must be satisfied, and not just sufficient decrease condition (4.3.8a), consider that the Hessian update is positive definite only if

$$\mathbf{s}_k^T\mathbf{y}_k > 0. \tag{4.3.9}$$

By (4.3.8b), we have

$$\begin{aligned}\mathbf{y}_k^T\mathbf{s}_k =& \alpha(\nabla\mathcal{J}_{k+1} - \nabla\mathcal{J}_k)^T\mathbf{p}_k \\ \geq& (c_2 - 1)\alpha\nabla\mathcal{J}^T\mathbf{p}_k > 0 \end{aligned} \tag{4.3.10}$$

since $0 < c_2 < 1$ and $\mathbf{p}_k$ is a descent direction. Hence, satisfying the (strong) Wolfe guarantees a positive definite Hessian update, and therefore that the $\mathbf{p}_k$ will be descent



(a) Relative residual

(b) Absolute error

(c) Relative change in solution

Figure 4.5: Relative objective residual, absolute 2-norm error of the solution and relative change in solution for the first 50 iterations of l-BFGS. The horizontal line in c) shows where the relative change in solution crosses 10^{-4}.

directions. For l-BFGS, it is sensible to perform a fairly loose search such as $c_1 = 10^{-4}$ and $c_2 = 0.9$ [123, pp. 142]. This way, more effort is placed into finding a step which will give a positive definite Hessian update, improving the descent direction for later iterations.

Nocedal and Wright [123, pp 60-61] give a line search algorithm which first finds a region $[\alpha_1, \alpha_2]$ in which an α satisfying () lies, then interpolating between these points. For completeness, this is included here as Algorithms 3 and 4.

Algorithm 3 Line Search Algorithm

```
$\alpha_0 \leftarrow 0$, choose $\alpha_{\max} > 0$ and $\alpha_1 \in (0, \alpha_{\max})$;
$i \leftarrow 1$;
loop
    Evaluate $\phi(\alpha_i)$
    if $\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0)$ or $[\phi(\alpha_i) \geq \phi(\alpha_{i-1})$ and $i > 1]$ then
        $\alpha_* \leftarrow \mathrm{zoom}(\alpha_{i-1}, \alpha_i)$;
        return $\alpha_*$;
    end if
    Evaluate $\phi'(\alpha_i)$;
    if $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$ then
        $\alpha_* \leftarrow \alpha_i$;
        return $\alpha_*$;
    end if
    if $\phi'(\alpha_i) \geq 0$ then
        $\alpha_* \leftarrow \mathrm{zoom}(\alpha_i, \alpha_{i-1})$;
        return $\alpha_*$;
    end if
    Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\max})$;
    $i \leftarrow i + 1$;
end loop            ▷ Loop ends with an $\alpha_*$ satisfying the strong Wolfe conditions
```

The trial step lengths generated by Algorithm 3 are monotonically increasing, and $\phi(\alpha)$ is assumed to a continuous function with continuous first derivative. If there exists an $\alpha \in (0, \alpha_{\max})$ such that

$$\phi(\alpha) = \phi(0) + c_1\alpha\phi'(0) =: \psi(\alpha), \tag{4.3.11}$$

then there is guaranteed to be at least one $\alpha_* \in (0, \alpha_{\max})$ satisfying the strong Wolfe conditions [123, pp 35]. To show this, first let α' smallest such α satisfying (4.3.11), then all $\alpha \in (0, \alpha')$ satisfy the sufficient decrease condition. By the mean value theorem, $\exists\alpha'' \in (0, \alpha')$ such that

$$\phi'(\alpha') - \phi(0) = \alpha'\phi'(\alpha''). \tag{4.3.12}$$

Combining (4.3.11) and (4.3.12), it is clear that α'' satisfies the curvature condition, and also satisfies the sufficient decrease condition since $\alpha'' < \alpha'$.

Assuming $\alpha_{\max}$ is sufficiently large that the above holds, the line search algorithm successively increases α_i until:

1. α_i no longer satisfies the sufficient decrease, or $\phi(\alpha_i) > \phi(\alpha_{i-1})$;
2. α_i satisfies the strong Wolfe conditions;
3. the cost function has positive gradient at α_i.

In the first and third cases, we have overstepped a (local) minimum of $\phi(\alpha)$, so there must be an $\alpha_* \in (\alpha_{i-1}, \alpha_i)$ satisfying the strong Wolfe conditions, which is found by the function zoom by interpolation between α_i and α_{i-1}. Otherwise, the function is still decreasing at α_i, and a larger step size is chosen via

$$\alpha_{i+1} = \lambda \alpha_i, \qquad \lambda > 1. \tag{4.3.13}$$

λ should be sufficiently large that the algorithm makes sufficient progress, but not so large that zoom is given too wide an interval to interpolate effectively. $\lambda \in (1.05, 1.5)$ can be an effective choice.

Note that Algorithm 3 only calculates a gradient where it is necessary: if α_i violates the sufficient decrease condition we needn't check if it satisfies the curvature condition. Note also that the order in which α_i and α_{i-1} are supplied to zoom varies: the first argument is always the one which is closest to satisfying the Wolfe conditions.

Algorithm 4 selects a new step length α_j between α_{lo} and α_{hi}. If this satisfies the strong Wolfe conditions it is returned, but if it does not a new step is selected between α_j and α_{lo}. Whichever gives the higher function value of α_j and α_{lo} is renamed to be α_{hi}, and the other to be α_{lo} for the next iteration.

Note that at the start of every iteration, we always have knowledge of $\phi(\alpha_{\text{lo}})$, $\phi'(\alpha_{\text{lo}})$ and $\phi(\alpha_{\text{hi}})$. We can therefore take the trial step length to be the minimiser of the quadratic interpolation of ϕ between α_{lo} and α_{hi},

$$\alpha_q = -\frac{b}{2a}, \tag{4.3.14}$$

Algorithm 4 Zoom

```
Input $\alpha_{\text{lo}}$, $\alpha_{\text{hi}}$;
loop
    Interpolate $\phi$ between $\alpha_{\text{lo}}$ and $\alpha_{\text{hi}}$ to find a trial step $\alpha_j$;
    Evaluate $\phi(\alpha_j)$;
    if $\phi(\alpha_j) > \phi(0) + c_1\alpha_j\phi'(0)$ or $\phi(\alpha_j) \geq \phi(\alpha_{\text{lo}}$ then
        $\alpha_{\text{hi}} \leftarrow \alpha_j$;
    else
        Evaluate $\phi'(\alpha_j)$;
        if $|\phi'(\alpha_j)| \leq -c_2\phi'(0)$ then
            $\alpha_* \leftarrow \alpha_j$;
            return $\alpha_*$;
        end if
        if $\phi'(\alpha_j)(\alpha_{\text{hi}}) - \alpha_{\text{lo}}) \geq 0$ then
            $\alpha_{\text{hi}} \leftarrow \alpha_{\text{lo}}$;
        end if
        $\alpha_{\text{lo}} \leftarrow \alpha_j$;
    end if
end loop            ▷ Loop ends with an $\alpha_*$ satisfying the strong Wolfe conditions
```

where

$$\begin{bmatrix} \alpha_{\text{lo}}^2 & \alpha_{\text{lo}} & 1 \\ \alpha_{\text{hi}}^2 & \alpha_{\text{hi}} & 1 \\ 2\alpha_{\text{lo}} & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \phi(\alpha_{\text{lo}}) \\ \phi(\alpha_{\text{hi}}) \\ \phi'(\alpha_{\text{lo}}) \end{bmatrix}.$$

If $\alpha_{\text{lo}} \neq 0$, we also always have the additional information of $\phi'(0)$ ($\phi(0)$ may not be additional information if $\alpha_{\text{hi}} = 0$), so can instead interpolate a cubic. A cubic may also be interpolated between $\phi(\alpha_{\text{lo}})$, $\phi'(\alpha_{\text{lo}})$, $\phi(\alpha_{\text{hi}})$ and $\phi(\alpha_{\text{old}})$ at later iterations, where $\alpha_{\text{old}} \neq \alpha_{\text{lo}}, \alpha_{\text{hi}}$ is a previously tested and rejected step.

This interpolation should be safeguarded to ensure the trial value is sufficiently far from either α_{lo} and α_{hi}.Similarly to Moré and Thuente [120], we take

$$\alpha_j = \begin{cases} \dfrac{\alpha_{\text{lo}} + \alpha_{\text{hi}}}{2}, & \text{if any} \begin{cases} \alpha_q < \min(\alpha_{\text{lo}}, \alpha_{\text{hi}}) + \delta \\ \alpha_q > \max(\alpha_{\text{lo}}, \alpha_{\text{hi}}) - \delta \\ |\alpha_q - \alpha_{\text{hi}}| < |\alpha_q - \alpha_{\text{lo}}| \end{cases} \\ \alpha_q, & \text{otherwise} \end{cases}, \tag{4.3.15}$$

where $\delta > 0$ is a small user-defined number. The selection (4.3.15) ensures a sufficiently large step is taken, and that it is closer to α_{lo} than α_{hi}, that it lies in the given range (if a cubic interpolation has been used it may not). We also return $\alpha_* = \alpha_{\text{lo}}$ if $|\alpha_{\text{lo}} - \alpha_{\text{hi}}| < \delta$ and $\alpha_{\text{lo}} \neq 0$, or if some maximum number of iterations have passed.

For l-BFGS, the step length $\alpha_0 = 1$ should be tried first by the line-search algorithm, as such a step length helps super-linear convergence to be observed [123, pp158-160]. In early iterations though the Hessian approximation may be poor, and so the descent direction poorly scaled. Choosing

$$\alpha_0 = \begin{cases} \dfrac{\mathcal{J}_k}{\lambda\phi'(0)} & \text{for } k = 1 \\ \dfrac{2(\mathcal{J}_k - \mathcal{J}_{k-1})}{\phi'(0)} & \text{otherwise} \end{cases} \tag{4.3.16}$$

interpolates a quadratic to $\mathcal{J}_k$, $\mathcal{J}_{k-1}$ and $\phi'(0)$, or a line in the first iteration. The contribution $\lambda > 0$ in the first iteration serves to rescale α_0, since the linear interpolant is a poor approximation. Adjusting this by

$$\alpha_0 = \min(1, 1.01\alpha_0), \tag{4.3.17}$$

which forces the selection of $\alpha_0 = 1$ as soon as it is reasonable.

Our final concern is with the maximum step length. Via equations (4.3.11) and (4.3.12), we showed that step lengths satisfying the Wolfe conditions were guaranteed if $\phi(\alpha') = \psi(\alpha')$ for some $\alpha' \in (0, \alpha_{\max}]$. This in turn is guaranteed if we take

$$\alpha_{\max} = \frac{\phi(0)}{c_1\phi'(0)} \tag{4.3.18}$$

since $\psi(\alpha_{\max}) = 0$, and ϕ is continuous and bounded below by some $\phi_{\min} > 0$, so it must intersect ψ. However, this range may be too great, breaking physical bounds on the parameters. In such cases $\alpha_{\max}$ must be reduced, but we cannot then guarantee the curvature condition will be satisfied by α_*.

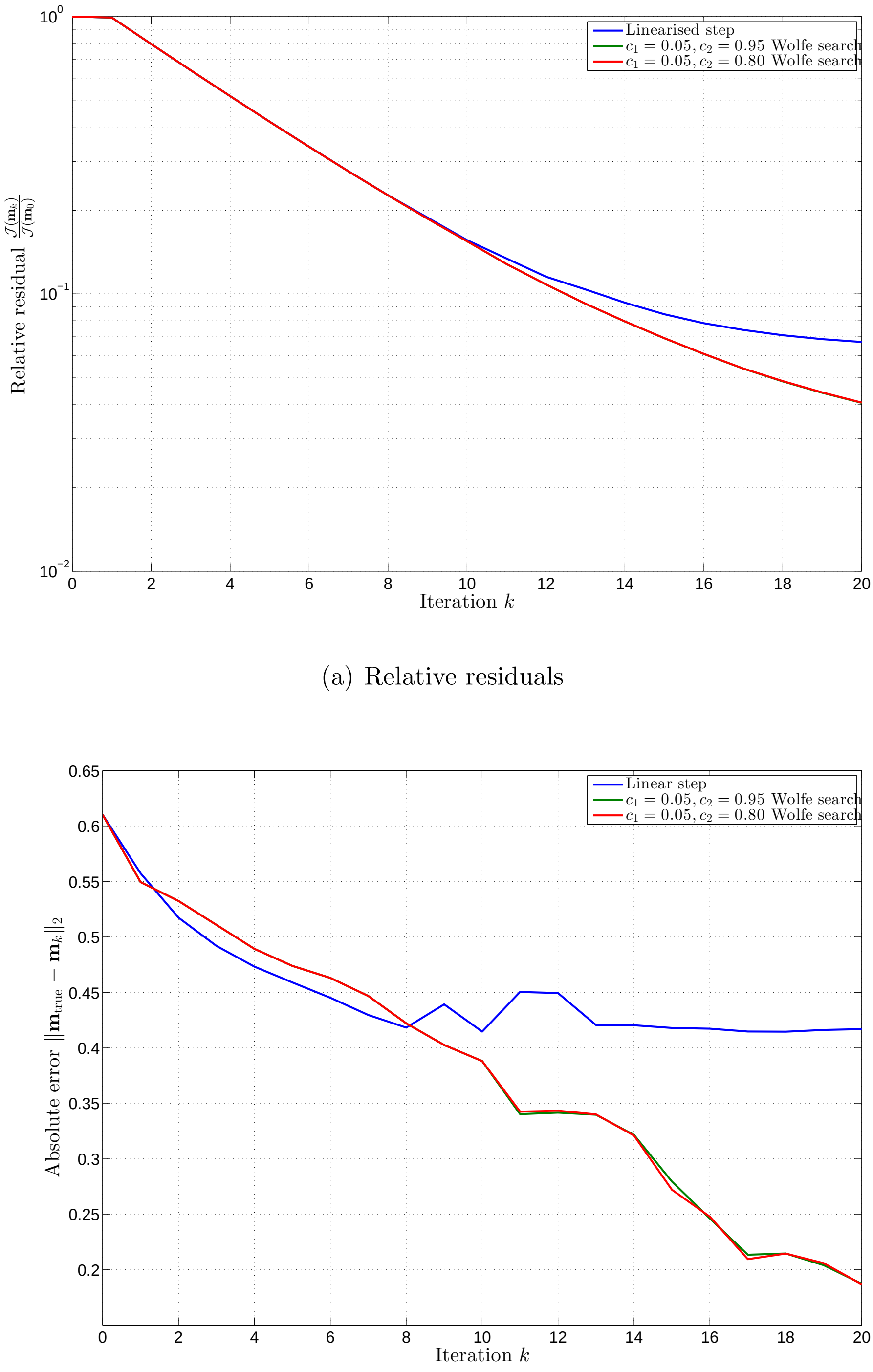
The line search described in this section may appear to be computationally expensive, but the actual cost will depend on well scaled the descent direction $\mathbf{p}_k$ is, and how well $\phi(\alpha)$ can be approximated by a quadratic or cubic function. Note that the linearisation (4.3.7) always takes 1 additional forward solve to calculate a step length α, before the objective and gradient are evaluated at $\mathbf{m}_k - \alpha\mathbf{p}_k$. In the strong Wolfe search, if the initial step length is accepted there is no additional cost. If it is not, Algorithm 4 interpolates this step length to find a more desirable selection, similar to calculating the linearised step length (4.3.7). A gradient is only calculated if a step length sufficiently reduced the objective function, and only rejected if the step length would result in a non positive definite l-BFGS update. (4.3.7) does not guarantee either a positive definite update or sufficient decrease in objective function. It may be particularly unsuitable if the objective function is highly nonlinear.

To demonstrate the necessity of a more involved line search than the linear step (4.3.7), we compare the first 20 iterations of the test problem detailed in Section 4.3.1 using a linearised step, to using Algorithms 3 and 4. For the linearised step, we take $\kappa = \|\mathbf{p}_k\|_\infty/100$ (as suggested by Pica [132]), and for the strong Wolfe conditions we take $c_1 = 0.05$ (a fairly tight condition), and both $c_2 = 0.95$ and $c_2 = 0.8$ (a loose and tight condition, respectively). The relative residuals and absolute errors of the optimisation procedures are shown in Figure 4.6, and Figure 4.7 the reconstruction results after 20 iterations.

Figure 4.6(a) shows us that the linearised step length prevents l-BFGS from reaching its optimal rate of convergence. Moreover, Figure 4.6(b) shows that after 8 iterations, while the residual continues to decrease, we fail to resolve any more details of the subsurface. From the reconstruction shown in Figure 4.7(a), which is fairly oscillatory and so will have a large TV norm, we hypothesise that this is due to failing to account for the non-linearity of TV regularisation by imposing a linearised step length. As we will see in Section 4.3.6, the way in which we have implemented TV regularisation means the non-linearity increases with each iteration, making the linearisation (4.3.7) a poor choice. Using the strong Wolfe line search however, we do more to approximately minimise $\mathcal{J}$ along $\mathbf{p}_k$, thus accounting for the non-linearity in the objective functional.

It is interesting to note that the additional cost of implementing a strong Wolfe line search was minimal in this example. Each iteration of l-BFGS requires two forward solves (one data simulation and one adjoint solve for the gradient). Using (4.3.7) required 20 additional forward solves (1 per iteration). The loose Wolfe search required only 9 additional forward solves, because in later iterations $\mathbf{p}_k$ was well scaled and $\alpha = 1$ was accepted, requiring no additional forward solves required. The stricter Wolfe search required 18 additional forward solves, which is still fewer than using the linearised step, though there is negligible difference between the results of the two reconstructions using Wolfe line searches.

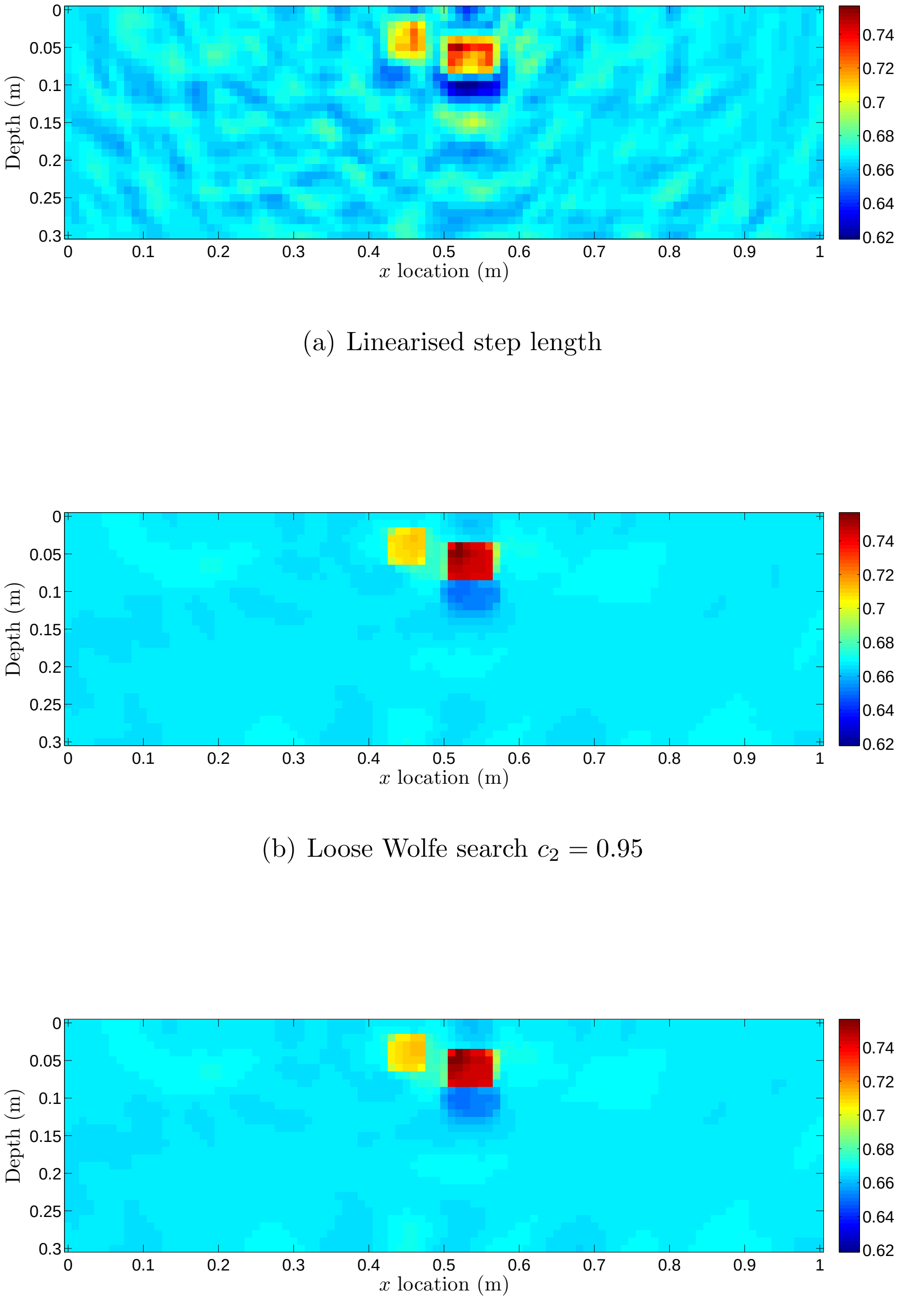
Note that our implementation of (4.3.7) is not thorough: we do not dynamically set κ as is more usual [55], and there are no safeguards implemented to ensure sufficient



(a) Relative residuals

(b) Absolute error

Figure 4.6: Comparison of (a) relative residuals and (b) absolute error for 20 iterations l-BFGS using a linearised step lengths to step lengths satisfying the strong Wolfe conditions



(a) Linearised step length

(b) Loose Wolfe search $c_2 = 0.95$

(c) Tight Wolfe search $c_2 = 0.80$

Figure 4.7: Reconstruction results after 20 iterations of l-BFGS for (a) linearised line steps, to strong Wolfe line searches with (b) $c_2 = 0.95$ and (c) $c_2 = 0.80$. Both strong Wolfe line searches used $c_1 = 0.05$.

decrease. Nevertheless, it seems fruitless to calculate forward solutions which will always be discarded, and it would appear necessary to satisfy the curvature condition to ensure a good l-BFGS descent direction and to properly account for non-linearity in the objective function.

4.3.5 Modifications for bounded optimisation

The line-search described above produces step lengths guaranteed to satisfy the Strong Wolfe conditions wherever $\alpha_{\max}$ is chosen by (4.3.18). If $\alpha_{\max}$ is less than that in (4.3.18), for example due to physical bounds on parameters, the strong Wolfe conditions are not guaranteed to be satisfied if

$$\alpha_{\max} = \underset{\alpha \in (0, \alpha_{\max}]}{\operatorname{argmin}} \; \phi(\alpha), \tag{4.3.19}$$

for example when both ϕ and ϕ' are monotonically decreasing functions of $\alpha \in (0, \alpha_{\max}]$.

The l-BFGS-b scheme [33] has been developed for problems with box constraints $m_{\min} \leq m_i \leq m_{\max}$. The method identifies free and fixed variables at each iteration via a steepest descent method, then updates only the free variables using the l-BFGS method. In our experience, since we are dealing with measurements caused by physical parameters which will be far from physical bounds (for example, we won't have in general a vacuum or perfect conductors present), the unbounded l-BFGS method will be unlikely to come close to violating the physical constraints. We have found it sufficient to identify the fixed variables as those for which $\mathbf{m}_k + \alpha_{\min}\mathbf{p}_k$ would violate the constraints, for some prescribed minimum step length $\alpha_{\min}$.

To ensure a positive definite Hessian update, Byrd et al [33] discard any vector pairs $(\mathbf{s}_k, \mathbf{y}_k)$ which would result in a non positive definite Hessian update. One can instead use a damped BFGS update

$$H_{k+1} = H_k - \frac{H_k \mathbf{s}_k \mathbf{s}_k^T H_k}{\mathbf{s}_k^T H_k \mathbf{s}_k} + \frac{\mathbf{r}_k \mathbf{r}_k^T}{\mathbf{s}_k^T \mathbf{r}_k}, \tag{4.3.20}$$

where

$$\mathbf{r}_k = \theta_k \mathbf{y}_k + (1 - \theta_k) H_k \mathbf{s}_k, \tag{4.3.21}$$

and Nocedal [123, pp 537] proposes the scalar θ_k is given by

$$\theta_k = \begin{cases} 1 & \mathbf{s}_k^T \mathbf{y}_k \geq 0.2\mathbf{s}_k^T H_k \mathbf{s}_k \\ (0.8\mathbf{s}_k^T H_k \mathbf{s}_k)/(\mathbf{s}_k H_k \mathbf{s}_k - \mathbf{s}_k^T \mathbf{y}_k) & \mathbf{s}_k^T \mathbf{y}_k < 0.2\mathbf{s}_k^T H_k \mathbf{s}_k \end{cases} \tag{4.3.22}$$

With this choice, $\theta_k = 0$ gives $H_{k+1} = H_k$, discarding the latest update, and $\theta_k = 1$ gives the possibly indefinite BFGS update. θ_k then interpolates between the previous Hessian approximation and the new update, ensuring H_{k+1} is positive definite but that as much new information is included as possible.

Equation (4.3.20) gives the dense BFGS Hessian update, and not the l-BFGS inverse Hessian. The damped limited memory BFGS update is similarly calculated by replacing $\mathbf{y}_k$ with $\mathbf{r}_k$, as implemented by Baali et al [6]. Calculation of θ_k (specifically $H_k\mathbf{s}_k$) is a little more involved since we do not store H_k nor wish to form the dense matrix. Instead, we solve

$$H_k \mathbf{s}_k = (B_k)^{-1} \mathbf{s}_k \tag{4.3.23}$$

for $H_k\mathbf{s}_k$ using an iterative method such as CG,with an implementation that can be passed a function returning products $B_k\mathbf{x}$ rather than the matrix B_k itself (e.g. MATLAB's inbuilt `cgs`). These products can then be calculated using the same 2-loop l-BFGS recursive method as used for the inverse Hessian gradient product, Algorithm 1. The cost of solving for $H_k\mathbf{s}_k$ is small, since if B_k is formed by m vector pairs $(\mathbf{s}_k, \mathbf{y}_k)$ and B_0 is taken to be a multiple of the identity, then B_k has at most $r = m+1$ distinct eigenvalues. It can easily be shown that conjugate gradient methods will terminate in at most r iterations [123, pp. 115]. The distinct eigenvalues of the l-BFGS inverse Hessian for the first 10 iterations are shown in Figure 4.8.

We reiterate that the damping only ever occurs where the line search fails to return step lengths satisfying the strong Wolfe conditions, which is a rare occurrence. While more complex choices than (4.3.20) are possible, for example switching to an SR1 update [5], they are not worth implementing for such occasional gain.

4.3.6 Regularisation

The regularisation functional R plays two main roles: preventing over-solving of an ill-posed system, thereby reducing the contribution of noise in the solution; and adding a-priori knowledge, restricting the solution of our ill-posed problem to one which fits

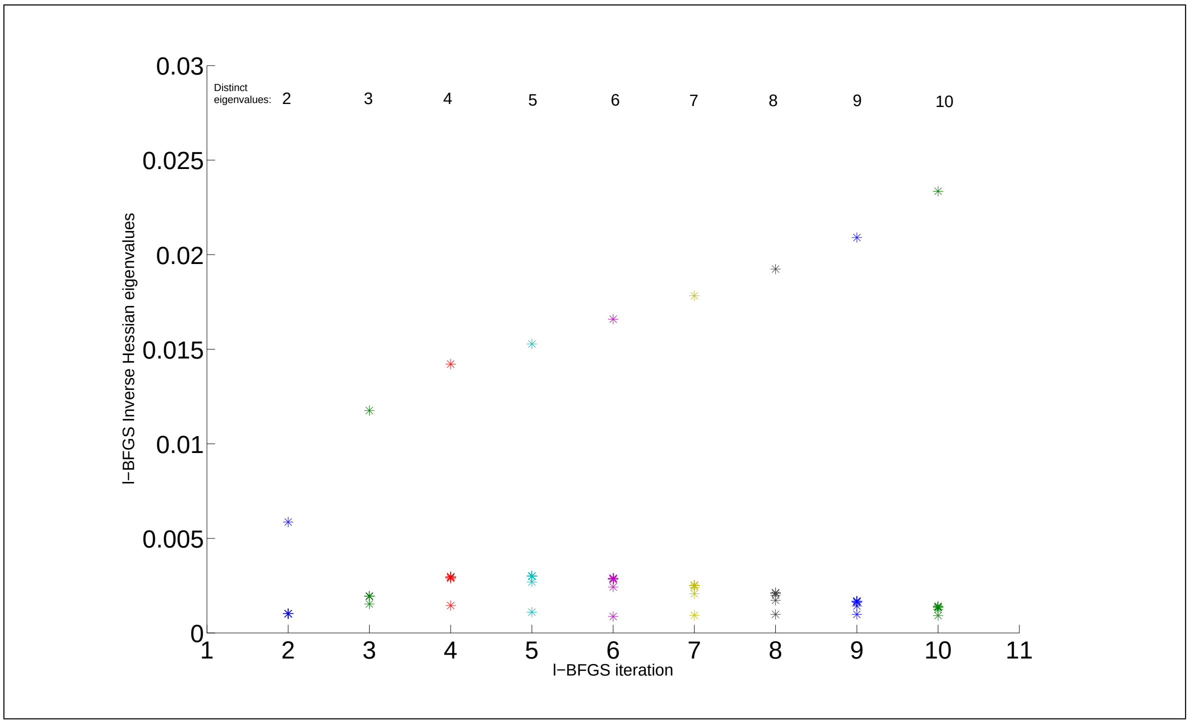


Figure 4.8: Eigenvalues of the l-BFGS inverse Hessian after the first 10 iterations. The number of numerically distinct (15 decimal places) eigenvalues is displayed, which is seen to be 1 more than the number of BFGS updates (which is 1 less than the iteration number)

some expected distribution of the subsurface parameters. For the landmine detection problem, we are looking for distinct objects in the ground with a solid boundary. A suitable regularisation functional for this is the Total Variation semi-norm,

$$TV(f) = \int_{\Omega} |\nabla f(\mathbf{x})| \, \mathrm{d}V, \tag{4.3.24}$$

where $f : \Omega \to \mathbb{R}$ for some domain $\Omega \subset \mathbb{R}^3$. The total variation semi-norm penalises high-frequency changes in f, but allows for jump changes. It therefore promotes solutions $f(\mathbf{x})$ to the inverse problem which vary little in some neighbourhood around $\mathbf{x}$, but that distinct jumps are promoted where this is not possible.

The gradient of the TV semi-norm is given by

$$\nabla TV(f) = -\nabla \cdot \left(\frac{\nabla f}{|\nabla f|} \right). \tag{4.3.25}$$

It is clear from (4.3.25) that the TV semi-norm is non-differentiable for a homogeneous function, which makes it unsuitable for gradient based optimisation methods (particularly so since we are likely to start from a homogeneous background). The gradient is also strongly non-linear [28]. One method to overcome this is to approximate the

Euclidean norm with [183, 184]

$$|x| \approx \sqrt{|x|^2 + \varepsilon^2},$$

for some $|\varepsilon| \ll 1$, which has a well-defined smooth gradient at $|x| = 0$. We then have the smoothed Total Variation semi-norm

$$TV_\varepsilon(f) = \int_\Omega \sqrt{|\nabla f(\mathbf{x})|^2 + \varepsilon^2}\, \mathrm{d}V. \tag{4.3.26}$$

In the 2D discrete setting, this is given by [184, pp 133]

$$TV_\varepsilon(\mathbf{m}) = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} \psi\left((D^x_{ij}m)^2 + (D^z_{ij}m)^2\right) h_x h_z, \tag{4.3.27}$$

where h_x and h_z are the step sizes in the $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$ directions, which have n_x and n_z total grid points;

$$D^x_{ij}m = \frac{m_{i,j} - m_{i-1,j}}{h_x}, \quad \text{and} \quad D^z_{i,j}m = \frac{m_{i,j} - m_{i,j-1}}{h_z}; \tag{4.3.28}$$

and

$$\psi(t) = 2\sqrt{t + \varepsilon^2}. \tag{4.3.29}$$

Following C. Vogel [184, pp. 133-134], the derivative of (4.3.27) in the direction $\mathbf{v}$ is given by

$$\begin{aligned}\left.\frac{\partial TV_\varepsilon}{\partial \mathbf{v}}\right|_{\mathbf{m}} =& \sum_{i=1}^{n_x}\sum_{j=1}^{n_z} \psi'_{ij}\left[(D^x_{ij}m)(D^x_{ij}v) + (D^z_{ij}m)(D^z_{ij}v)\right] h_x h_z, \\ =& \left\langle \mathrm{diag}(\psi'(\mathbf{m}))D_x\mathbf{m}, D_x\mathbf{v}\right\rangle + \left\langle \mathrm{diag}(\psi'(\mathbf{m})D_z\mathbf{m}, D_z\mathbf{v}\right\rangle, \end{aligned} \tag{4.3.30}$$

where $\psi'_{ij} = \left((D^x_{ij}m)^2 + (D^z_{ij}m)^2\right)$ and $\mathrm{diag}(\psi')$ is the diagonal matrix whos entries are the ψ'_{ij}, D_x and D_z correspond to the grid operators in (4.3.28), and $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. From this, we obtain that the gradient with respect to model parameters is given by

$$\nabla_{\mathbf{m}} TV_\varepsilon(\mathbf{m}) = L(\mathbf{m})\mathbf{m}, \tag{4.3.31}$$

where

$$L(\mathbf{m}) = \begin{bmatrix} D_x^T & D_z^T \end{bmatrix} \begin{bmatrix} \mathrm{diag}(\psi'(\mathbf{m})) & 0 \\ 0 & \mathrm{diag}(\psi'(\mathbf{m})) \end{bmatrix} \begin{bmatrix} D_x \\ D_z \end{bmatrix}. \tag{4.3.32}$$

Implementing a smoothed Total-Variation regularisation in this manner, one can reduce the parameter ε after each iteration of l-BFGS, so that $TV_\varepsilon \to TV$. In early iterations this must be done gradually, since the gradient may be close to singular if $\mathbf{m}_i$ is close to homogeneous.

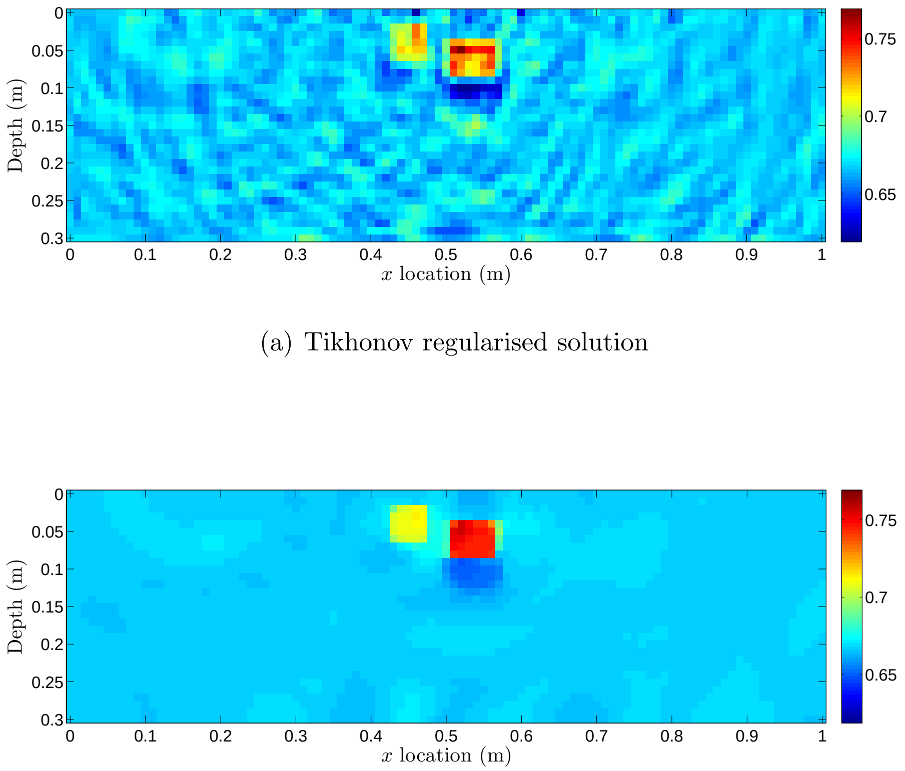
A. Borsic [28] has compared various starting values for ε as well as the rate in which they are decreased for solving the electrical impedance tomography (EIT) problem using a Newton method, as well as for fixed values of ε. In general, a larger value results in a more stable scheme, but the approximation of the TV semi-norm is poor and so final results will suffer.

It can be more efficient to solve the TV regularised inverse problem with a Primal Dual method, in which an alternative problem is constructed using the Lagrangian of the objective function, referring to A. Borsic [28] for an overview of Primal Dual methods for the TV regularised inverse problem up to 2002. More recently, it has been shown that using a split Bregman method can converge quickly as well as being suitable for large-scale problems [62]. For the simpler test problems considered in this chapter, it is sufficient for us to implement an iteratively reweighted quasi-Newton method for the smoothed TV functional, in which ε is reduced only after sufficient decrease in objective function, and is increased if a sufficient reduction cannot be found. Pseudocode for this method is given in Algorithm 5. Note that when ε is increased due to an insufficient reduction in the objective function, the iterate $\mathbf{m}_{k+1}$ is thrown away (i.e. iteration k is recalculated).

Algorithm 5 Iteratively Reweigheted l-BFGS

```
set $\varepsilon$, $\mathbf{m}_0$;
$k = 0$;
while Convergence criteria not met do
    Calculate $\mathbf{p}_k$                                              ▷ via l-BFGS Algorithm 1
    Set $\mathbf{m}_{k+1} \leftarrow m_k + \alpha_k \mathbf{p}_k$ satisfying Strong Wolfe conditions    ▷ via Algorithm 3
    if sufficient decrease of $\mathcal{J}(\mathbf{m}_{k+1})$ then
        decrease $\varepsilon$;
        $k \leftarrow k + 1$;
    else
        Increase $\varepsilon$;
    end if
end while
```

Figure 4.9 compares a Total Variation to a Tikhonov regularised solution (2.2.6) with a homogeneous prior. The reconstruction with Tikhonov regularisation is poor because it does not add relevant prior information about the solution: the background



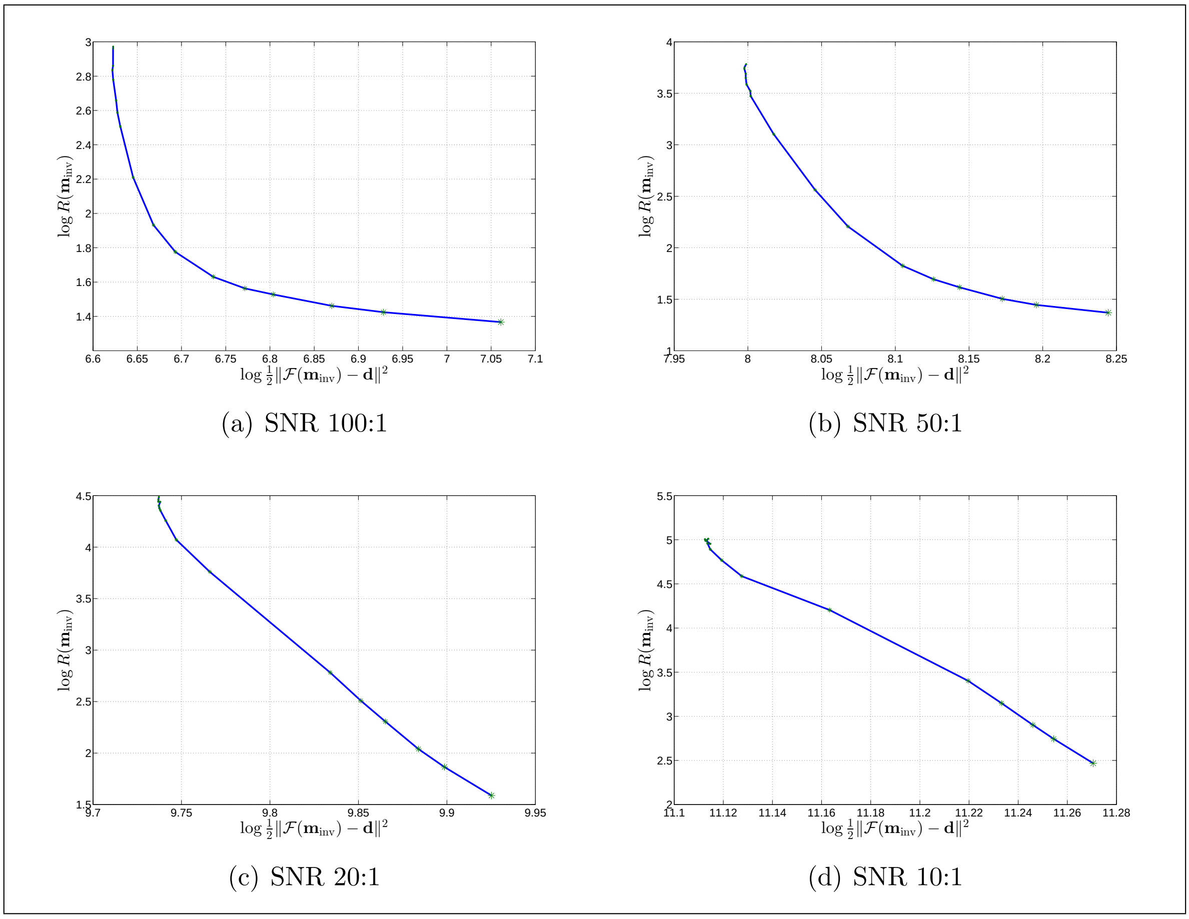
(a) Tikhonov regularised solution

(b) Total Variation regularised solution

Figure 4.9: (a) Tikhonov and (b) Total Variation regularised solutions to the FWI problem, both with $\lambda = 1.5$.

is close to the correct value, but the inclusions are restricted away from the correct values. Tikhonov regularisation also does a poor job at preventing oscillatory artefacts, which arise due to the gradient being formed by solutions to wave equations.

As we discussed in Section 2.2.2, the amount of regularisation used (choice of $\lambda > 0$ in (4.3.1)) provides a trade-off between fitting the model to the data, and incorporating *a priori* knowledge to ensure uniqueness and stability of the solution. This can be highlighted by l-curve plots of the logs of the data misfit $\|\mathcal{F}(\mathbf{m}_{\text{inv}}) - \mathbf{d}\|^2$ against the regularisation misfit $R(\mathbf{m}_{\text{inv}})$, for various values of λ. Such plots generally show a large regularisation term and small data residual for small λ, owing to the dominance of noise in the solution, and vice-versa for large λ, with an area of high curvature for intermediate values of λ (the "corner" of the L). According to Hansen and O'Leary [72] (see also [70] and [71, pp. 187-193]), this point of high curvature



(a) SNR 100:1

(b) SNR 50:1

(c) SNR 20:1

(d) SNR 10:1

Figure 4.10: L-curve plots for the TV regularised FWI problem with varying signal-to-noise ratio, with regularisation parameters $\lambda \in [0, 500]$.

provides a qualitative means of choosing the regularisation parameter λ. This criterion does not always hold [68, 182], the L-curve plot does not always form an L shape (for example when the underlying assumption that the data residual and the smoothing regularisation function are not monotonic functions of λ).

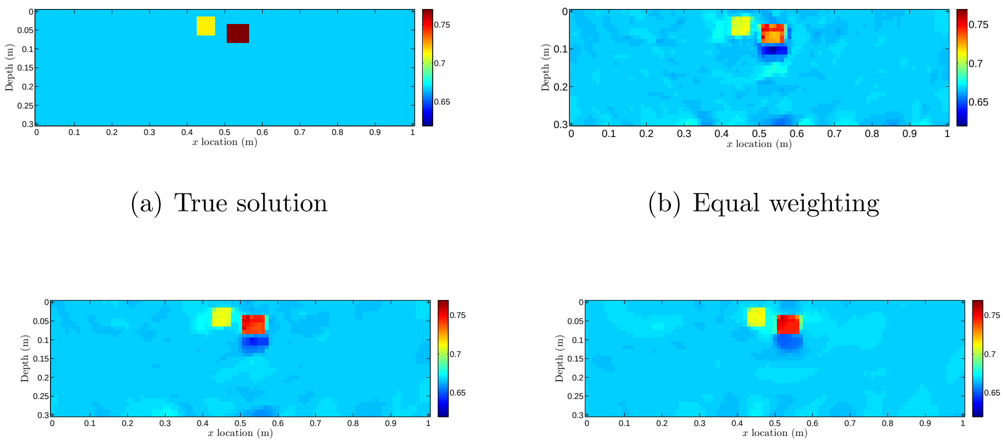
Figure 4.10 gives L-curve plots for the TV regularisd FWI problem with signal-to-noise ratios between 100 and 10. While the trade-off between data residual and TV norm is clear, there is not a distinctly high point of curvature in any of the plots. Moreover, for large amounts of noise the plot does not show a convex corner at all for the non-linear Total-Variation regularisation. The cluster of points in Figures 4.10(c) and 4.10(d) varying discontinuously with $\lambda \ll 1$ highlights the instability of the inverse solution to noise.

4.3.7 Frequency selection

The scattering of waves in the subsurface largely occurs due to features large compared to the wavelength, with features smaller than the wavelength having a much smaller effect. Lower frequency components of the recorded data then will largely be due to larger scatterers in the subsurface, whereas higher frequency components will contain the detail of smaller objects. With higher frequency components affected by more length scales of objects, they are likely to have been scattered more times, and so fitting higher frequency may be a more non-linear problem. Contrarily, it may be much easier to determine large-scale features to lower frequency data, which has only been reflected from large objects (ignoring some features), but reconstructing the detail of features small compared to the wavelength becomes a severely ill-posed problem (compare to EIT). There may also be differences in the noise level at different frequencies, perhaps with higher frequency data being more susceptible to background thermal noise and so having a greater variance.

In light of this, one often attempts first to reconstruct large-scale features (expecting a better conditioned problem) before resolving the detail, and there are two common approaches: invert all the data simultaneously but with a different weighting for each frequency component, for example Hu et al [76] weight proportionally to $1/\omega^2$ for seismic inversion; or begin with lower frequency components, and adding higher frequency components to the objective function (removing or not the lower frequencies) in later iterations, for example as implemented by Ravaut et al [139] for seismic data, or Meles et al [112, 113] for GPR data. The latter has the additional benefit of reducing the computational cost of evaluating the objective function in early iterations, while the former uses a more over-determined dataset so may be more robust to noisy data.

Figure 4.11 compares the reconstruction, and Figure 4.12 the relative residuals and 2-norm absolute error, after 20 iterations of l-BFGS for different weights of frequency components in the data. We see from Figure 4.11 that the $\frac{1}{\omega}$ and $\frac{1}{\omega^2}$ weightings have acted as ‘smoothing operators’, reducing the contribution of noise when compared with the equal weighting reconstruction. The artefact beneath the greater contrasted object is also reduced, possibly due to internal multiple scattering (at higher frequencies) having less of an impact. Moreover, Figure 4.12 shows that with the frequency



(a) True solution

(b) Equal weighting

(c) $\dfrac{1}{\omega_i}$ weighting

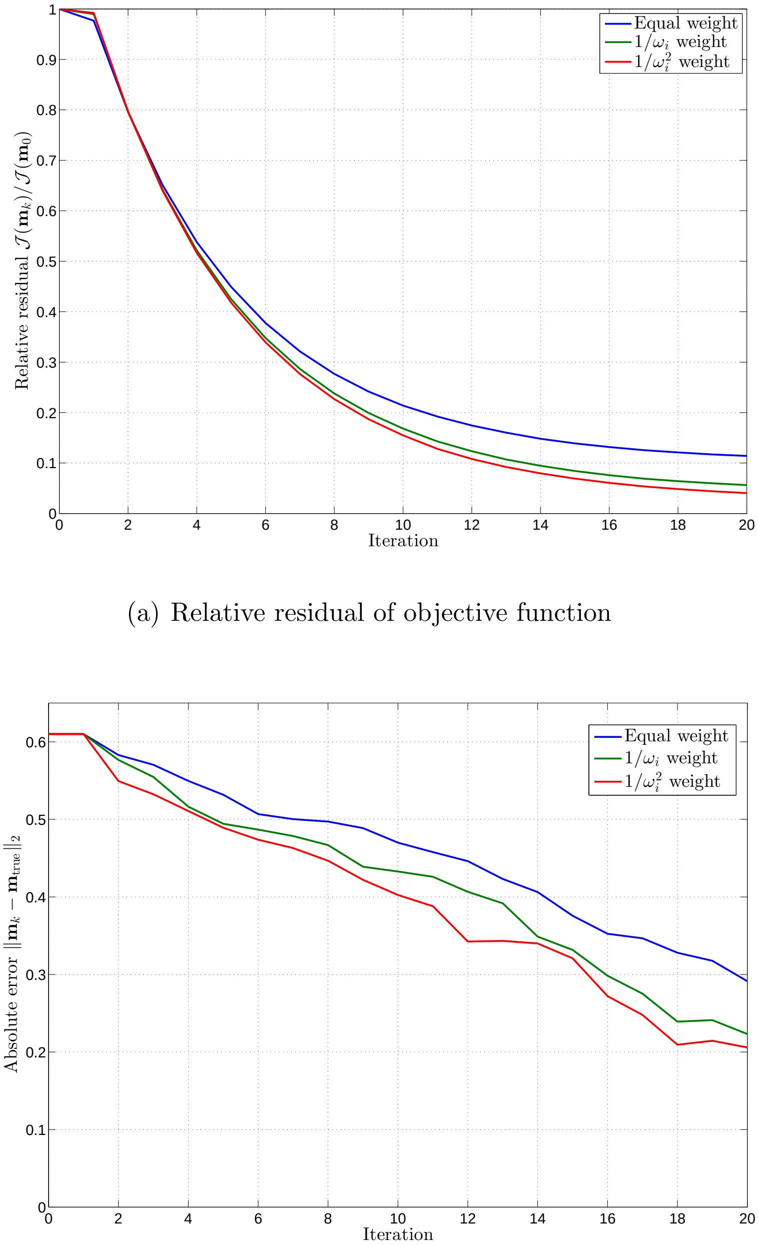
(d) $\dfrac{1}{\omega_i^2}$ weighting

Figure 4.11: Comparison of 2D inversions after 20 iterations of l-BFGS for different weights of frequency components. a) is the true image, b) the result for equal weighting, c) the result for reciprocal of frequency weighting, and d) the result for reciprocal of frequency squared. Colour scale (same in each) showing reciprocal of wavespeed $\dfrac{1}{c} \times 10^8$ s/m

weighted objective functionals, l-BFGS is able to achieve more per iteration, both in reducing the residual and in reducing the absolute error compared with the true solution.

4.4 Numerical experiment

We present here a numerical experiment to reconstruct a domain with more complex objects: a target with a hollow; and a cross shaped target. The cross shape was used by Meles et al [111] to increase multiple scattering, and so the complexity of the inverse problem, in a cross-borehole experiment. Similarly, the hollow target will increase the non-linearity of the inverse problem due to the increased presence of internal multiples and smaller-scale features. We are particularly interested in whether we are able to resolve the hollow target, since landmines generally have some form of air gap or multiple material types. The domain is discretised into 0.25 cm square voxels. Approximately 10% of the voxels had a small normally distributed random perturbation in wave speed added, with a variance of 5×10^5 ms^{-1} – a more realistic stochastic model of granular soil is later used in Chapter 8. The domain is shown in



(a) Relative residual of objective function

(b) 2-norm absolute error of iterate compared to true solution

Figure 4.12: Comparison of relative residual and absolute error for different weights of frequency components for the first 20 iterations of l-BFGS.

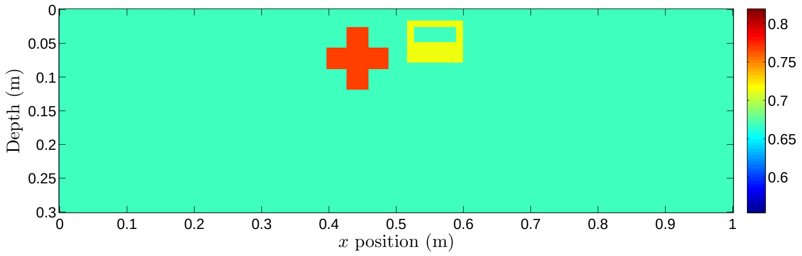


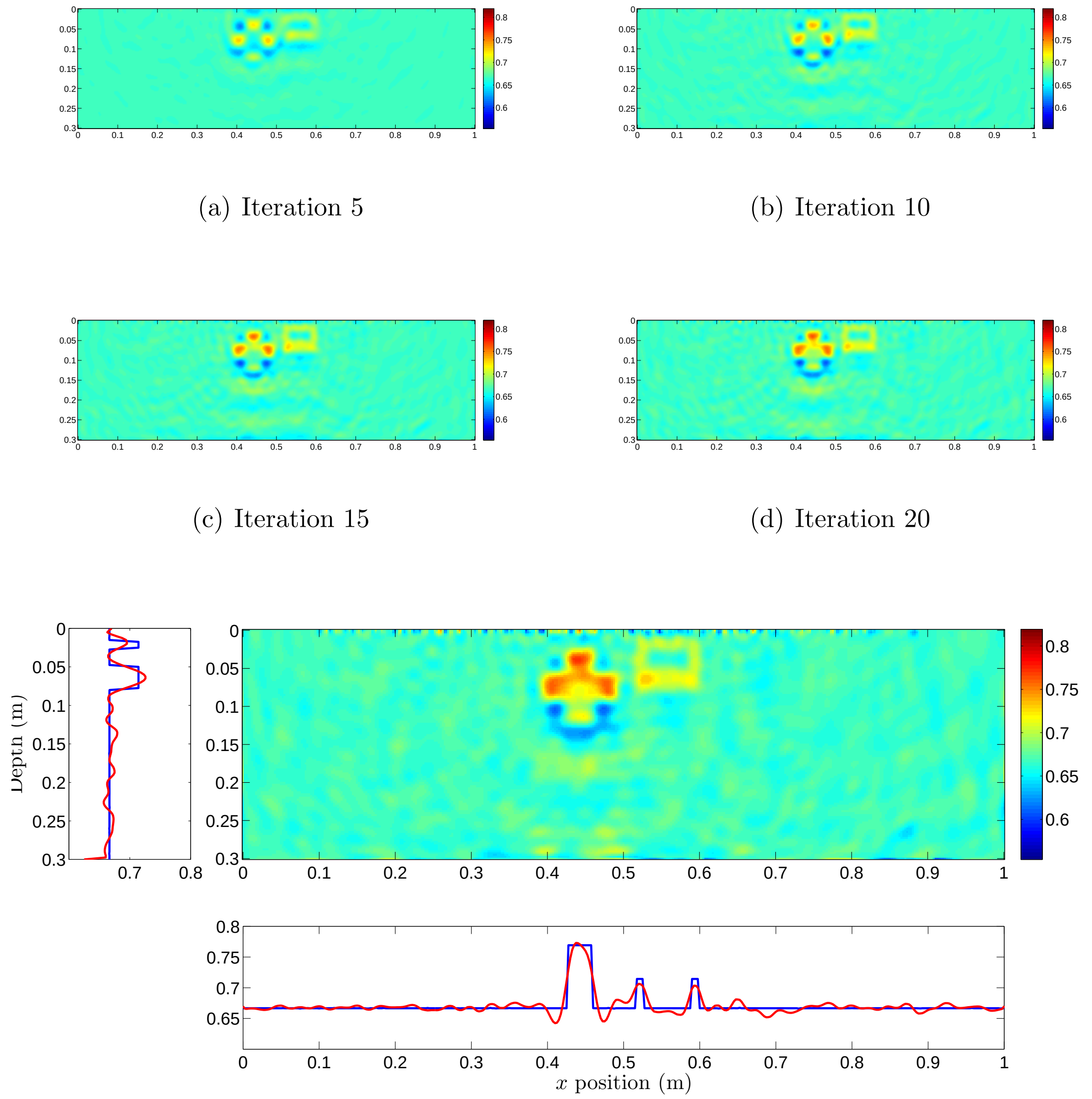
Figure 4.13: Test domain in 2D with one cross-shaped and one hollow target.

Figure 4.13.

We simulate data at 100 frequencies linearly spaced between 1 and 3 GHz, at 100 source locations across the surface with 3 receivers evenly spaced across a 20 cm maximum source-receiver offset: 30,000 measurements for the 48,521 parameters, and so a definite nullspace which we need to account for with prior knowledge (regularization). Following Section 4.3.6, we use TV regularisation with $\lambda = 2$. Gaussian white noise was added to the simulated data at a signal to noise ratio of 20:1. Figure 4.14(e) shows the reconstruction after 5, 10, 15, 20 and 30 iterations of l-BFGS, with a horizontal and vertical slice through the hollow object at 30 iterations.

We are able to reconstruct the approximate wavespeed and shape of the objects, including the interior of the hollow object. We are not aware of hollow objects such as this being considered in other GPR FWI numerical experiments. Reconstructing objects which vary internally is difficult, as one must necessarily account for internal (multiple) scattering. The lower part of the cross has not yet been fully resolved, which is to be expected as ray paths to this area must first pass through some other part of the cross (i.e. data due to reflections from the bottom point of the cross is not linear). We also see that the top three points of the cross (and not the centre or bottom) are resolved first, as there is a direct reflection in the data from them.

Either more iterations of l-BFGS, a larger amount of regularisation (λ) or smaller TV small number ϵ is needed to account for the wave-like artefacts in the reconstruction, which are due to (a combination of) multiple scattering and incomplete data coverage.



(a) Iteration 5

(b) Iteration 10

(c) Iteration 15

(d) Iteration 20

(e) Iteration 30

Figure 4.14: Reconstruction of the more complex domain in Figure 4.13 after 5, 10, 15, 20 and 30 iterations of l-BFGS. The vertical slice in (e) is at $x = 0.55$ cm, and the horizontal at $z = 0.04$ cm. Blue line shows the true domain, and red the reconstructed.

4.5 Summary and conclusions

In this chapter, we have implemented a finite-difference solver for the Helmholtz equation to simulate GPR data in 2D, and used this as the forward solver for 2D FWI. The 2D Helmholtz problem was approximated by a finite difference scheme, and the resulting sparse linear system solved using a pre-conditioned BiCGSTAB method. A sufficiently fine grid spacing is required to prevent high numerical dispersion. We have described the FWI numerical recipe in full, with simulated examples to introduce to the reader to all the implementational aspects of inversion. We have shown that, in so far as the Helmholtz equation is a sufficient model for GPR data, it is possible to carry out 2D FWI for (simulated) GPR data for hand-held arrays.

We are not aware of other examples of GPR FWI which consider data from small-scale arrays, as generally a large array is used for data redundancy. While Lavoué et al [90, 91] have used the l-BFGS-b optimisation method for GPR FWI, we have modified the method to give a damped Hessian update, rather than skipping it, on occasions when the update would not be positive-definite. This way we still retain as much additional information about the Hessian as possible, whilst ensuring the l-BFGS direction $-\mathbf{p}_k$ is a descent direction. We did not quantify the benefits of such an approach, as in numerical experiments we have undertaken it is rare that bounds are reached.

We are the first to apply Total Variation regularisation to the GPR FWI problem, though as previously noted it has been applied to the more general inverse Helmholtz problem. The TV regularised FWI problem was solved via an iteratively re-weighted l-BFGS scheme, which we found to converge in a reasonable number of iterations (~ 30), and did indeed lead to solutions with the jump discontinuities desired. For more complex problems it may be desirable to implement a primal-dual method to solve the TV regularised problem. While the l-BFGS method has led to a good rate of convergence, it would also be interesting to implement other quasi-Newton methods such as other members of the Broyden class (e.g. SR1), or hybrid quasi-Newton CG methods such as those given Babaie-Kafaki et al [19].

We have also given a more thorough treatment of the step length calculation, where more often a step length is chosen via a simple linearisation. This can provide improved convergence by ensuring a non-trivial step is taken, while ensuring the step is not so large that the quadratic model $\mathcal{M}(\mathbf{m}_k)$ (and thus descent direction) has become poor, at little or no additional computational cost. This is particularly important when the inverse problem is highly non-linear, and so we deem it necessary when using a non-linear regularisation method such as Total Variation. Performing a line search satisfying the strong Wolfe conditions ensures the Hessian update will be positive definite, whilst first testing a unit step length helps the super-linear convergence of l-BFGS to be observed. We have also found no additional cost of implementing a strong Wolfe line search compared with calculating a linearised step length.

In our 2D numerical experiments, it was always possible to obtain good approximations of the wavespeed of scatterers given simulated noisy data, including the cross-shaped and hollow targets in the numerical experiment of Section 4.4. We are not aware of hollow targets such as this being considered in other GPR FWI numerical experiments, which are difficult to reconstruct as this necessarily involves accounting for internal multiple scattering. This is a positive result, though we acknowledge that the GPR landmine detection/classification problem is indeed a 3D problem which limits its applicability. In the proceeding chapters we consider the full 3D inversion problem, using the same iteratively re-weighted l-BFGS scheme for the TV regularised inverse problem, but with a more physically complete forward model.

Chapter 5

SVD analysis of GPR Full-Wave Inversion

5.1 Introduction

Following Watson and Lionheart [192], in this chapter we will use singular value decomposition (SVD) as a tool to investigate the map from data to image, helping us understand what it is possible for a given antenna array to resolve. Oberröhrmann et al [124] have, similarly, used a chequerboard analysis to determine optimal acquisition setup for coverage and resolution of a cross-borehole experiment. Meles et al [111] also used singular values as a tool to measure the reliability of inverted GPR images from full-waveform data, for 1 to 4-sided experiments. Our results differ in that we will also use the singular vectors to analyse the imaging and null-spaces of the FWI system. Silvestrov and Tcheverda [157] carry out a similar analysis for a cross-borehole seismic experiment, though they use the method to analyse an appropriate parametrisation of the problem rather than the acquisition system.

The landmine detection problem motivates us to get the best possible information out of the most affordable equipment, and less complex acquisition systems with fewer antenna are of course more affordable. Acknowledging that FWI is generally carried out using multi-static arrays for data redundancy, we ask whether enough information can be gained from bi-static systems (or multi-static systems with few receive antenna) for a reliable FWI by taking measurements at many more source locations. To put the question another way, do these additional measurements, from many more physical locations and at many frequencies, yield enough independent information suitable for an inversion? We of course expect that we will be unable to fully resolve the sub-surface parameters using only bi-static data, but following our inversion results in Chapter 4 we believe it possible to perform FWI with a small-scale hand-held array. While we concentrate on the specific case of a hand-held GPR system, our methodology can readily be used to analyse the suitability of other acquisition systems.

To compensate for the lack of multi-offset data we must finely sample the frequency spectrum of data in our experiment, and so take 100 frequencies between 1 and 3 GHz to ensure a correct coverage of imaged wavenumbers [159].

5.2 Statement of the problem

One of the difficulties of solving equation (2.5.1) is that many different parameter sets $\mathbf{m}$ can give the same measurements, up to the noise level, so we need sufficiently strong *a-priori* information to obtain a unique and stable solution. Much emphasis has been placed on developing stable and efficient optimisation procedures for solving the FWI problem, particularly in the seismic community [137, 159]. However, unlike in the seismic imaging case, where data is recorded on large arrays of receivers but over a small frequency band (generally in the range 1–20 Hz with $\mathcal{O}(10)$ sampled frequencies), in the hand-held GPR case we can have only a small array of antennas but a wider frequency band (in this case, 1–3 GHz with 100 sampled frequencies). We therefore wish to analyse what such an acquisition system is able to resolve, and if a more complex antenna array (i.e. large multi-static) is required for good results.

The full-wave inversion problem is solved iteratively with a suitable nonlinear optimisation scheme, such as the (l-)BFGS method [123, p. 177], and as such requires calculation of the gradients of the data residuals $\mathcal{J} = \frac{1}{2}\|\mathcal{F}[\mathbf{m}] - \mathbf{d}\|_2^2$ with respect to the parameters $\mathbf{m}$, which we recall are given by

$$\nabla_{\mathbf{m}}\mathcal{J} = \Re\left\{J^T \delta\mathbf{d}^*\right\}, \tag{5.2.1}$$

where $\delta\mathbf{d} = \mathcal{F}[\mathbf{m}] - \mathbf{d}$ are the data residuals, and $J(\mathbf{m})$ is the Jacobian matrix of partial derivatives $J_{ij} = \partial\mathcal{F}_i/\partial m_j$, while ^{T} denotes a (non-conjugate) transpose and * denotes complex conjugation. Computation of the Jacobian matrix is not mandatory in order to calculate the gradient, which can be efficiently computed via the adjoint state method (2.5.22). We calculate the Jacobian matrix via the adjoint state method (2.5.21) for this work.

If we assume the use of a (quasi-) Newton method to solve the FWI problem, then the descent direction $\mathbf{p}$ is given by

$$H\mathbf{p} = -\nabla_{\mathbf{m}}\mathcal{J}(\mathbf{m}), \tag{5.2.2}$$

where H is (an approximation to) the Hessian matrix.

The solution to the linearised inverse problem, neglecting regularisation, is [40]

$$\begin{aligned} \mathbf{m}_{\text{lin}} &= \mathbf{m}_0 + \Re\{\delta\mathbf{m}\}, \\ J\delta\mathbf{m} &= \delta\mathbf{d}, \end{aligned} \tag{5.2.3}$$

where $\mathbf{m}_0$ is the initial estimate, and $\delta\mathbf{d}$ is the data residual $\mathcal{F}[\mathbf{m}_0] - \mathbf{d}$. The least-squares solution is given by

$$\delta\mathbf{m} = (J^T J)^{-1} J^T \delta\mathbf{d}. \tag{5.2.4}$$

The similarities between a linear inversion, gradient calculation and iterative updates to the solution of the nonlinear inverse problem are made clear on comparison of equations (5.2.1), (5.2.2) and (5.2.4), with the gradient and solution to the linearised inverse problem differing by a scale factor [39, 133]. Note also that $J^T J$ is the Gauss-Newton approximation to the Hessian matrix H. In our example of the (l-)BFGS method, the first iteration is a gradient descent in which the linear inversion is realised. Since we expect the dominant part of the data to be from first order reflections, we assert that the Jacobian matrix exhibits the dominant features of the map from data to image, and in particular the Jacobian of the first iteration of the inversion. Since our optimisation method will always update the image in a direction close to $J^T\delta\mathbf{d}$, we cannot expect to be able to resolve anything in the nullspace of J^T.

5.3 Inverse problems and singular value decomposition

A singular value decomposition (SVD) of $J \in \mathbb{R}^{m\times n}$ is given by

$$J = U\Sigma V^T = \sum_{i=1}^{q} \mathbf{u}_i \sigma_i \mathbf{v}_i^T, \tag{5.3.1}$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthonormal matrices with columns $\mathbf{u}_i$ and $\mathbf{v}_i$, respectively, and $\Sigma \in \mathbb{R}^{m \times n}$, $q = \min(m, n)$, is a diagonal matrix of non-negative values $\Sigma_{ii} = \sigma_i$, ordered such that

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_q \geq 0. \tag{5.3.2}$$

The σ_i are called the singular values, the $\mathbf{u}_i$ and $\mathbf{v}_i$ are called the left- and right-singular vectors of J, respectively. The left singular vectors span the data space, and the right span the image space. Note that the σ_i are not conductivity σ_i.

Linear ill-posed problems are characterised by the singular values decaying rapidly to zero, so that [28, 71]

$$\begin{aligned} \sigma_1 \geq \ldots \geq \sigma_a > \mu \geq \sigma_{a+1} \geq \ldots \geq \sigma_b > \epsilon \geq \ldots \\ \ldots \sigma_{b+1} \geq \ldots \geq \sigma_c > 0 = \sigma_{c+1} = \ldots = \sigma_q, \end{aligned} \tag{5.3.3}$$

where μ is the accuracy threshold set by the measurement instrumentation, and ϵ is the threshold for which observations are negligible. Note that μ and ϵ are not permeability μ and permittivity ϵ. The singular values $\sigma_1, \ldots, \sigma_a$ thus relate to reliable measurements which will be correctly mapped by the Jacobian to a descent direction introducing model parameters responsible for the recorded data. The singular values $\sigma_{a+1}, \ldots, \sigma_b$ relate to unreliable measurements, and the rest are null (to machine precision). The former will result in noise being introduced to the solution, and the latter will have no effect.

If we could measure a continuum of data the linearised forward problem would be an operator $\mathcal{A} : X \to Y$ between infinite dimensional Banach spaces X and Y. As we refine our discrete model and $q \to \infty$, the singular values of J converge uniformly to those of $\mathcal{A}$ [69, 71, pp. 43-44]. In common with most inverse problems, the operator $\mathcal{A}$ is compact and so does not have a bounded inverse, reflected in the decay of singular values, and their rate of decay gives us a useful measure of the ill-posedness of the inverse problem. If there is a $\gamma > 0$ such that the singular values satisfy $\sigma_n = \mathcal{O}(n^{-\gamma})$, then by Hofmann [74] (see also [121, pp 44-45]) the degree of ill-posedness is defined as:

1. *mildly* ill-posed if $0 < \gamma \leq 1$;

2. *moderately* ill-posed if $\gamma > 1$;

3. *severely* ill-posed if $\sigma_n = \mathcal{O}\big(\mathrm{e}^{-\gamma n}\big)$.

These definitions are related to the Picard condition, that the data $d \in Y$ must satisfy

$$\sum_{i=1}^{\infty} \left(\frac{\langle u_i, d \rangle_Y}{\sigma_i} \right)^2 < \infty \tag{5.3.4}$$

in order for a square-integrable solution to the inverse problem $\mathcal{A}^{-1}d$ to exist, which is a stronger statement than $d \in L^2(Y)$. Here, $u_i \in Y$ are the singular functions of $\mathcal{A}$ forming a basis for the (infinite dimensional) data space, and σ_i the singular values. We cannot guarantee that the noise component of d satisfies (5.3.4), and so the contributions of noise to the naive solution $\mathcal{A}^{-1}d$ will 'blow up' as $1/\sigma_n$.

Note that while J may represent an over-determined system (5.2.3) (i.e. we have more measurements than parameters we are trying to find, $m > n$), a will typically be far less than the number of parameters n. In this sense, we are left with an *effective* Jacobian,

$$J_{\text{eff}} = \sum_{i=1}^{a} \mathbf{u}_i \sigma_i \mathbf{v}_i^T, \tag{5.3.5}$$

which maps measurements to the image space spanned by the first a singular vectors $\mathbf{v}_i$ only. Therefore, we can in fact only expect to be able to resolve targets represented by the vectors in the row space of J_{eff}, rather than the full matrix J.

Given measurements of relative accuracy r, we can estimate the accuracy threshold μ by using the 'rule of thumb' that the condition number of J_{eff} should not be greater than $1/r$, and noting that $\text{cond}(J_{\text{eff}}) = \sigma_1/\sigma_a$. This rule of thumb ensures that J_{eff} satisfies the discrete Picard condition [71, p. 82]. More precisely though, the meaning of μ is [71, p. 22]

$$\text{rank}(J_{\text{eff}}) \equiv \min_{\|E\|_2 \leq \mu} \text{rank}(J + E), \tag{5.3.6}$$

in which we may consider the arbitrary $m \times n$ matrices E to represent modeling errors. Solving the linearised inverse problem via the truncated singular value (TSVD) solution

$$\delta\mathbf{m}_{\text{inv}} = \underset{\delta\mathbf{m}}{\text{argmin}} \, \| J_{\text{eff}} \delta\mathbf{m} - \delta\mathbf{d} \|_2^2 = (J_{\text{eff}}^T J_{\text{eff}})^{-1} J_{\text{eff}}^T \delta\mathbf{d} \tag{5.3.7}$$

is equivalent to a Tikhonov regularised solution [71, pp. 109-111], [25, pp. 257]

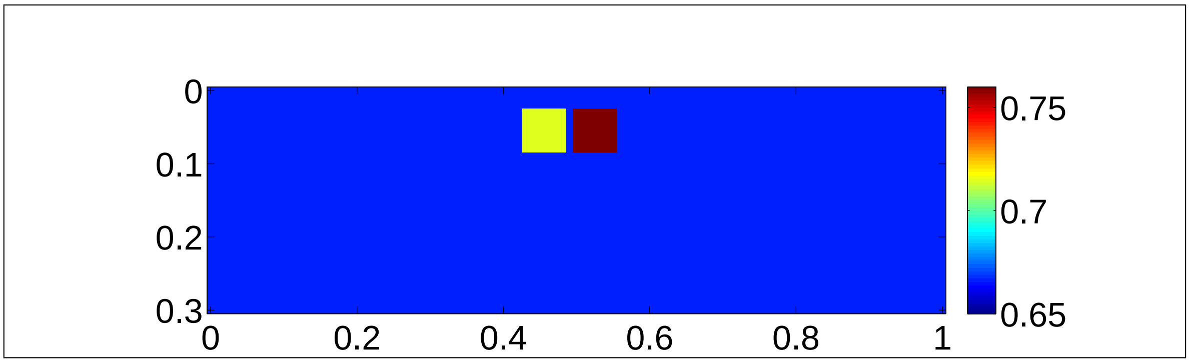


Figure 5.1: Test domain $\mathbf{m}_{\text{true}}$ used for projections and simulating data for inversion. Colour scale showing reciprocal of wave speed $c^{-1} \times 10^{-8}\text{sm}^{-1}$. Note that projections are onto $\delta\mathbf{m} = \mathbf{m}_{\text{true}} - \mathbf{m}_0$ for a homogeneous background $\mathbf{m}_0$ of wave speed 1.5×10^8 m/s.

5.4 Numerical results

We now calculate the Jacobian matrices using an adjoint-state formulation [137], before taking an SVD and projecting objects of interest onto the row space of the respective effective Jacobians to illuminate whether the system would be able to resolve such a target.

Our simulated systems all have a 20 cm maximum source-receiver offset, a limit chosen to represent our need for hand-held devices. Measurements are taken at 100 frequencies linearly sampled between 1 and 3GHz, at source positions evenly spaced between 10 cm and 70 cm along the surface of a 1 m wide by 30 cm deep domain (so that the final receiver position is 10 cm from the other end of the domain). These limitations were chosen represent possible hand-held GPR measurements.

The test domain, shown in Figure 5.1, is 1 m wide by 0.3 m deep with absorbing boundary conditions on the sides and bottom, and has a background wave speed of $1.5 \times 10^8\text{ms}^{-1}$. It is discretised to 1 cm for imaging purposes, however it is refined further to 0.33 cm for both forward and adjoint solutions of the Helmholtz equation to ensure accuracy in the higher frequencies. This gives a minimum of 15 grid points per wavelength, which is sufficient for the purpose of this experiment.

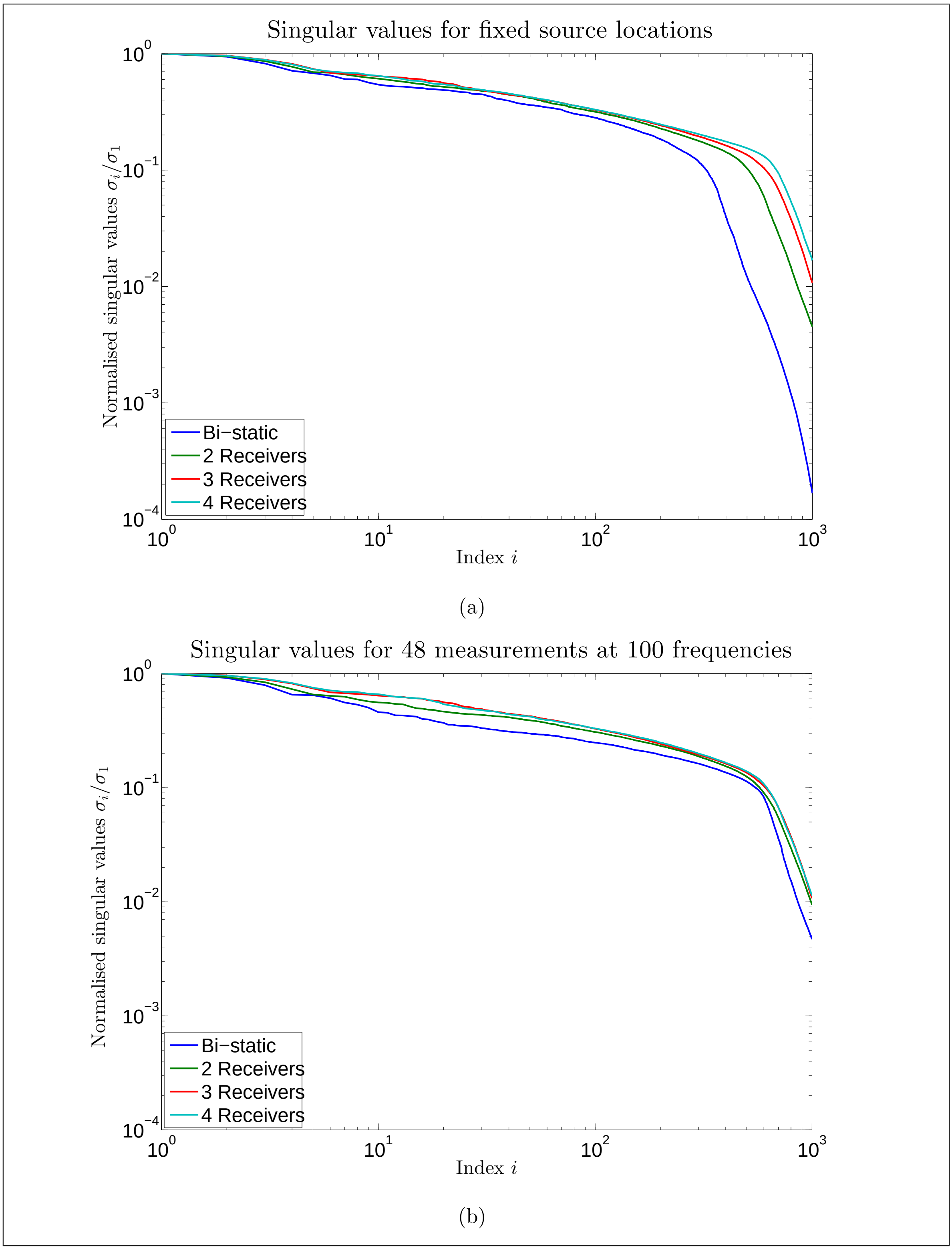
There are several parameters describing the acquisition array which we could investigate with an SVD analysis, including the number or offset of antennae; and the number or position of source locations. In this experiment we consider number of receive antennae, which we evenly space in the 20 cm interval, fixing other parameters.

We analyse both fixing the total number of measurements for our comparison, reducing the number of source locations with an increased number of receive antennas, and fixing the source locations. In the former case we keep $n_s \times n_r = 48$, where n_s is the number of source locations and n_r the number of receive antennas, and in the latter we keep $n_s = 16$. For our accuracy threshold, we accept singular vectors $\mathbf{v}_i$ associated with singular values σ_i satisfying $\sigma_i > 10^{-2}\sigma_1$, which represents a possible accuracy of noisy measurements with 2 significant figures accuracy. Our results are not sensitive to a (reasonable) change in this accuracy threshold, as the number of singular vectors above the threshold for each test case remain in proportion.

5.4.1 SVD analysis

Figure 5.2 shows us the first 1000 singular values for Jacobian matrices calculated for the homogeneous domain, representing the first linear step of the inversion. The near straight lines in these log-log scale plots show that the singular values decay like $i^{-\gamma}$, for $\gamma > 0$ a positive real constant. Initially this is with $0 < \gamma < 1$, indicating a mildly ill-posed problem, but increasing to $\gamma > 1$ at around $i = 500$ (depending on the set of singular values in question), which indicates a moderately ill-posed problem. This fits the rapid decay typical of an inverse problem, as mentioned in Section 5.3. The steepening at $i = 500$ is an effective drop in rank, implying less independent information can be gained from subsequent measurements.

Moreover, the steepening in decay of singular values indicates that the reconstruction of some types of perturbations in the subsurface – those which can be represented by $\mathbf{v}_i$ for $i < 500$ – may be better posed than others. Figures 5.3 and 5.4 show some of the singular vectors for the bistatic and 4 receiver systems, and we remark that those corresponding to smaller σ_i become more oscillatory: a phenomenon often observed but which there is no known proof in general [121, pp. 45]. We therefore expect recovery of larger (less oscillatory) perturbations to be generally more stable to noise. This is in line with the the analytic results of Karamehmedović [82], who found (roughly) that, for the inverse source problem of the Helmholtz equation in 2D, reconstruction of objects large compared with the wavelength is stable (through (2.4.3), it is easy to see the inverse source problem is equivalent to the linearised inverse parameter problem). Referring to Karamehdović's results, we expect the drop in rank occurs when



(a)

(b)

Figure 5.2: The first 1000 normalised singular values of the Jacobian matrices for bi-static and multi-static acquisition systems. 5.2(a) show the singular values for the case where we have fixed the source locations, and 5.2(b) the case where we have kept the total the number of measurements constant.

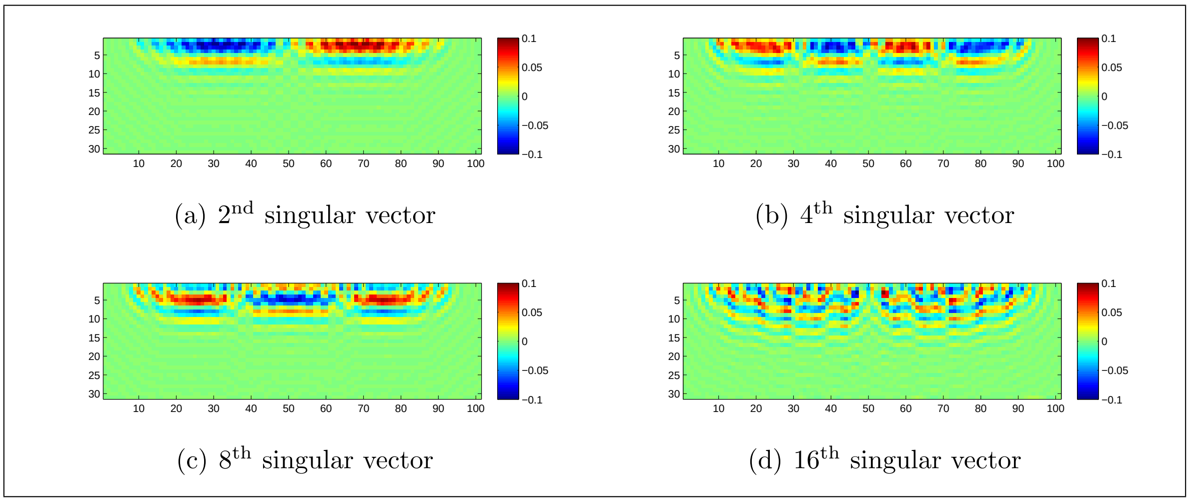
the singular vectors have sufficiently small length-scale perturbations compared with the wavelength.

Figure 5.2(a) shows a slower decay in singular values for systems with more receivers, which is as we would expect. However, when we fix the total number of measurements, Figure 5.2(b) shows a similar rate of decay for all four systems. Particularly, the σ_i for the bi-static system decay almost as slowly as those for the multi-static systems, and those for the 3 and 4 receiver case are almost indistinguishable. This suggests we may be able to at least partially compensate for a lack of multi-offset data by increasing the number of source locations, as each results in a Jacobian matrix with a similar condition number (or, equivalently, a J_{eff} of similar rank).

However, it is not just the rank and condition of J_{eff} that are important, or equivalently amount of data above the noise level, but the suitability of these measurements to image our target. One can only expect to image something which makes a significant enough impact to observed data and better measurements do not necessarily just mean more observations above the noise level. To understand the suitability of our measurements, one must also look at the singular vectors $\mathbf{v}_i$, and how they map the imaging space. Figures 5.3 and 5.4 show the singular vectors for the bi-static and 4 receiver systems with a fixed total number of measurements. We remark that the bi-static vectors show a skew-symmetry, the multi-static show anti-symmetry, and both systems show a greater sensitivity nearer sources/receivers. As we mentioned above, earlier singular vectors are also less oscillatory, and so different systems will likely lead to different stabilities in the recovery of certain parameters with small length-scale perturbations (such as nearby objects presenting jump changes).

For a realistic noise level J_{eff} will always have a large nullspace, and so taking more measurements to result in a better conditioned Jacobian is not going to help if what we want to be able to image still lies in this nullspace. To help us understand the image and nullspaces of J_{eff}, we can project test targets onto singular vectors $\mathbf{v}_i$ corresponding to singular values above the accuracy threshold. As previously mentioned, we take the accuracy threshold to be given by the σ_i satisfying $\sigma_i > 10^{-2}\sigma_1$.

The targets we use are 5 cm squares placed 3 cm below the surface and 2 cm apart, and have wave speeds of 1.3×10^8 and 1.4×10^8 ms^{-1} respectively, as shown in Figure 5.1. These are chosen as we are interested in whether a system is able



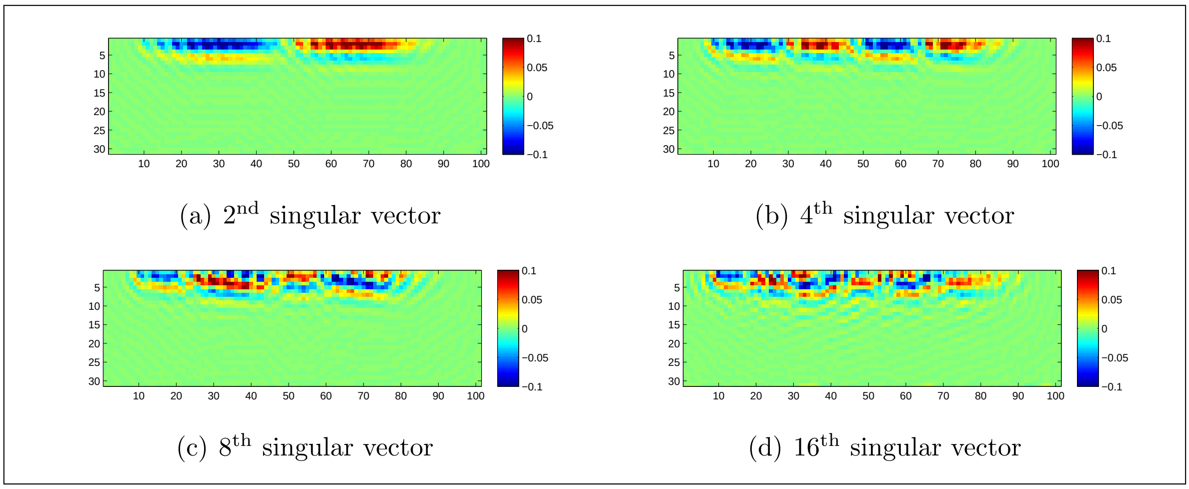
(a) 2nd singular vector

(b) 4th singular vector

(c) 8th singular vector

(d) 16th singular vector

Figure 5.3: The 2nd, 4th, 8th and 16th singular vectors for the bi-static system with 48 source locations and 100 frequencies between 1 and 3 GHz.



(a) 2nd singular vector

(b) 4th singular vector

(c) 8th singular vector

(d) 16th singular vector

Figure 5.4: The 2nd, 4th, 8th and 16th singular vectors for the 4 receiver system with 12 source locations and 100 frequencies between 1 and 3 GHz.

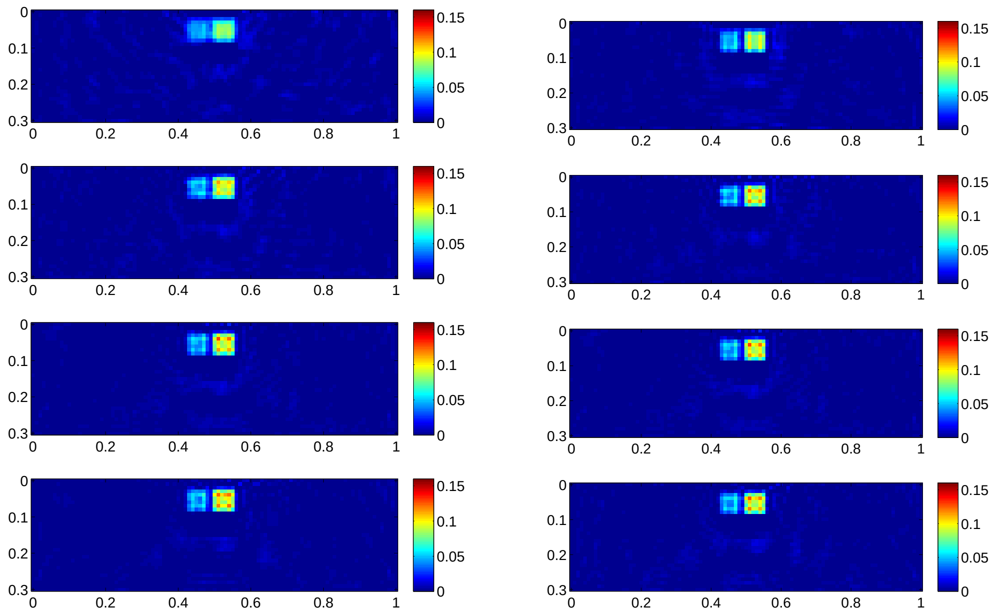


Figure 5.5: Projections of the two low contrast targets onto the right singular vectors associated with singular values σ_i which satisfy $\sigma_i > 10^{-2}\sigma_1$. From top to bottom: projections associated with an acquisition system with 1, 2, 3 and 4 receive antennas with a maximum source/receiver offset of 20 cm. Column a) contains the projections for fixed source locations, and column b) for a fixed number of measurements. Colour scale showing reciprocal of wave speed $c^{-1} \times 10^{-8}\text{sm}^{-1}$. Note that projections are onto $\delta\mathbf{m} = \mathbf{m}_{\text{true}} - \mathbf{m}_0$ for a homogeneous background $\mathbf{m}_0$ of wave speed 1.5×10^8 m/s.

to distinguish nearby objects based on quantitative information, particularly with a low contrast to one another. The low contrast, both between the objects and to the background, also means that the linear approximation should be a good one.

The results of the projections are shown in Figure 5.5, and the relative errors of the projection compared to the true $\delta\mathbf{m}$ are given in Table 5.1. As we ought to expect, for fixed source locations we get better results for the systems with more receivers, with a progressively better contrast between the two objects. The improvement going from bi-static to 2 receivers is significant, but increasing further to 3 or 4 receivers less so. Particularly, the bi-static projection gives a lower contrast and a ‘noisier’ background. This is clear from Table 5.1, which shows the projection of the 4 receiver system is 15.9% closer to the true solution than the bi-static case, with the 3 and 4 receiver projections on par.

For the projections with a fixed total number of measurements, only the bi-static

Acquisition system	A	B
Bi-static	37.5%	32.7%
2 receivers	29.1%	26.0%
3 receivers	26.5%	26.5%
4 receivers	24.1%	26.7%

Table 5.1: Relative error of projections of low contrast targets onto the right singular vectors of acquisition systems, for fixed source locations (column A) and fixed total number of measurements (column B)

result remains poor, with all multi-static projections giving results to within 1% relative error of each other. The implication is that, provided we have some degree of multi-static data in 20 cm offset limit, we ought to be able to at least partially compensate for a lack of large array by taking more measurements.

5.4.2 Inversion results

We now invert data simulated for each of the four acquisition systems, for both 16 source locations and 48 total measurements (locations × receivers), at 100 frequencies between 1 and 3GHz, with 5% Gaussian noise added. We use a small weight of total variation regularisation [184, p. 129] for stability, and allow 20 iterations of the l-BFGS optimisation procedure.

The inversion results are shown in Figure 5.6, and Figure 5.8 shows the absolute model error with each iteration. It is clear that the inversion of bi-static data reaches a local minima much farther from the true solution than the multi-static cases, and we are also almost unable to distinguish the two objects from one-another in the end result. Conversely, it is easy to distinguish the two objects from the inversions of multi-static data, as highlighted by the horizontal slices of the inversion shown in Figure 5.7.

When we fix the total number of measurements (an increase in measurements for 2 receivers, reduction for 4, and the same data set for 3 receivers), the results from the multi-static inversion do become slightly more similar, though there is little improvement in the bi-static case. This is most apparent from the plots of model error, Figure 5.8. This backs up the result of our projections onto singular vectors, in which

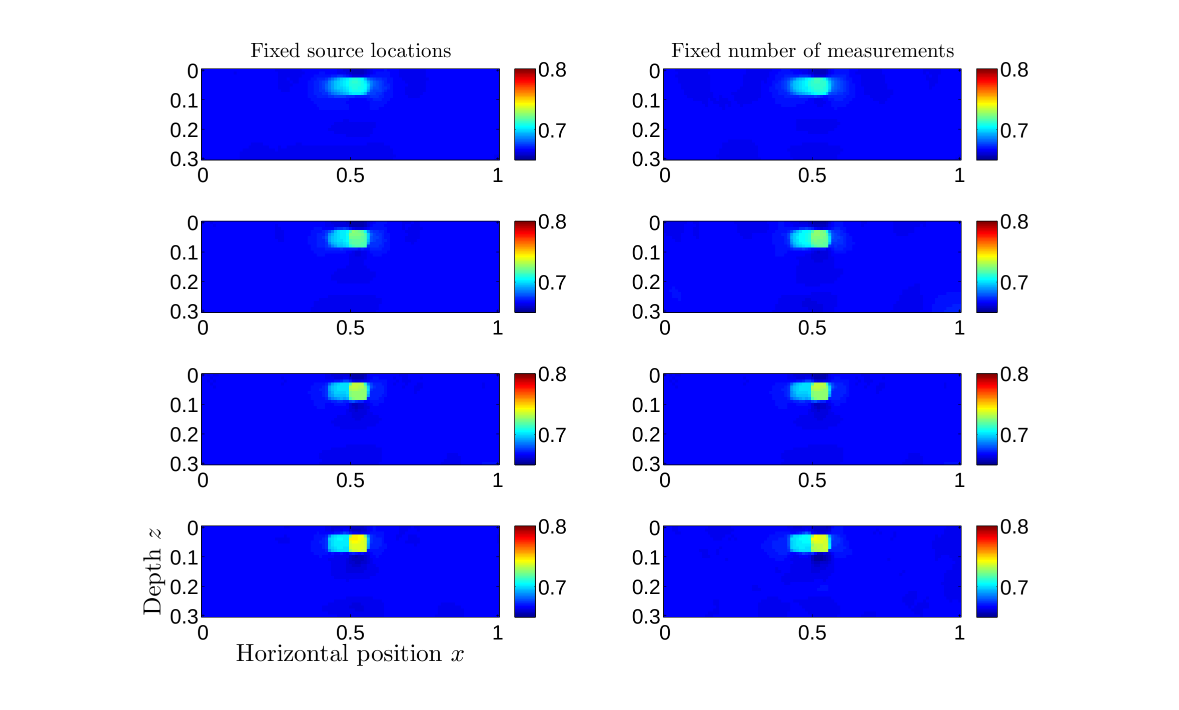
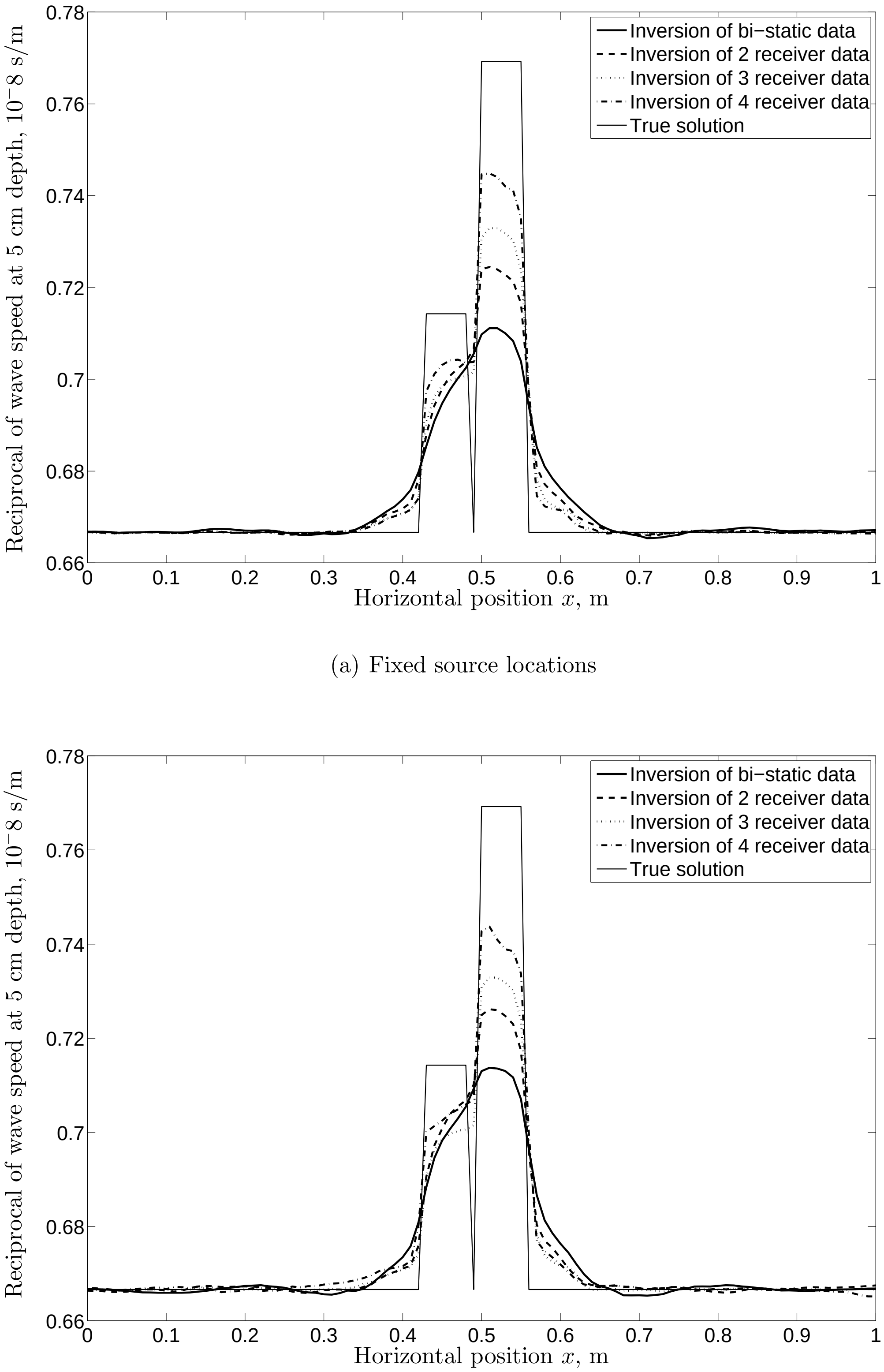


Figure 5.6: From top to bottom: inversion results for 1, 2, 3 and 4 receiver data sets, respectively. The left column is for a fixed source locations, and the right for a fixed total number of measurements.

we saw similar results when keeping the total number of measurements fixed. We suspect the difference between the multi-static inversion results with a fixed number of measurements is due to the non-linear part of the inversion (i.e. inversion of the part of the data caused by diffraction and multiple reflections between the two objects), since this part is ignored by the SVD analysis.

5.5 Discussion and further work

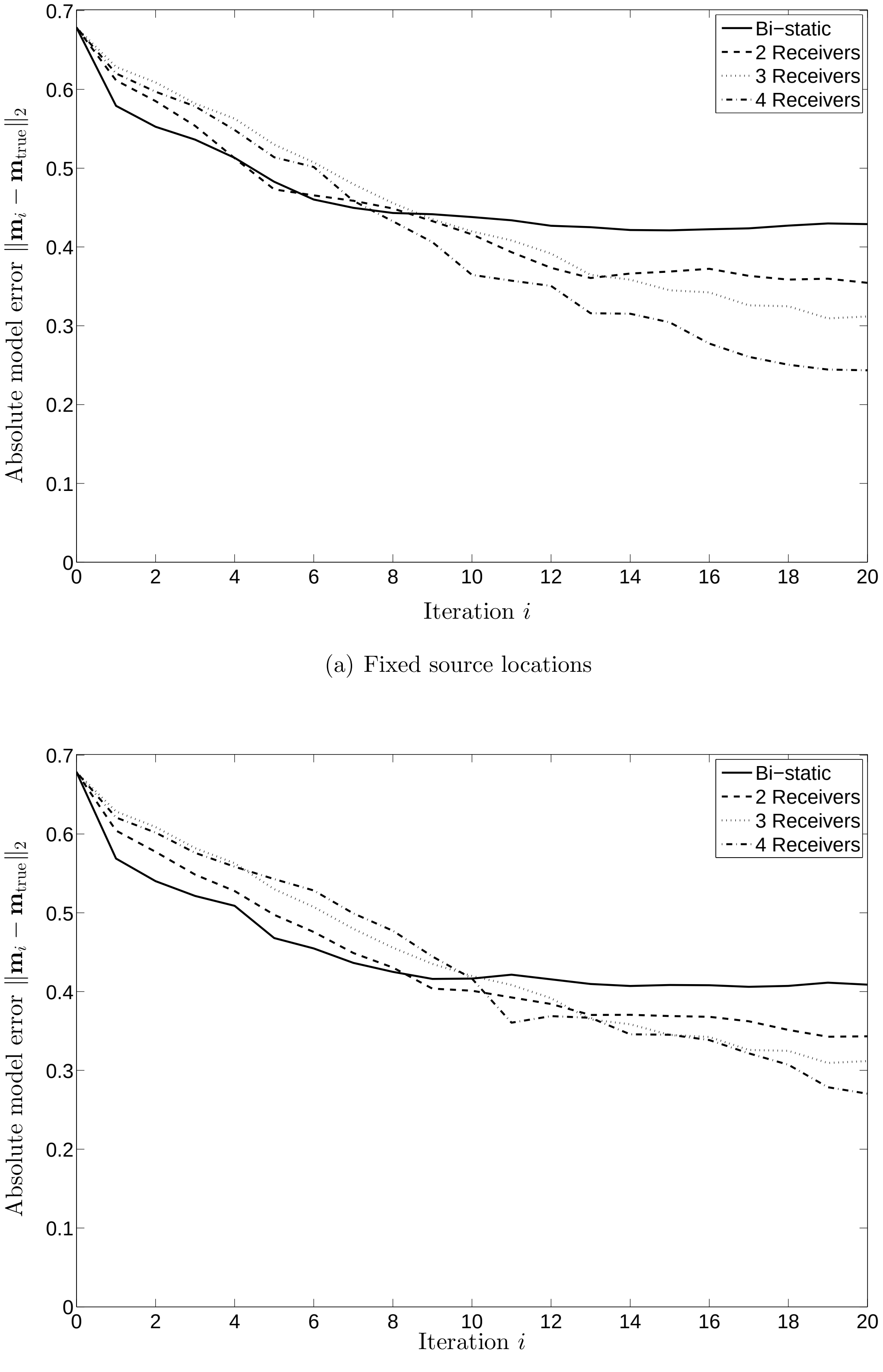
Taking an SVD of the Jacobian matrices shows us how data can be mapped to the gradient in our inversion procedure, allowing us to see what can (or can not) be resolved in a linear step of the inversion for a given acquisition system. Since the inversion procedure consists of many more non-linear updates, a linear step being unable to resolve a certain target does not mean we cannot resolve it at all, but that we will be relying on non-linear steps to do so – e.g. relying on data caused by multiply scattered or refracted waves. Further, we expect the singly scattered waves to give the biggest contribution to the data [157], and so the first linear step ought to give the largest contribution. Because of this, our SVD analysis gives a good indication of



(a) Fixed source locations

(b) Fixed total number of measurements

Figure 5.7: Horizontal slice at 5 cm depth of inversion results compared to true solution, for (a) fixed source locations and (b) fixed total number of measurements. Showing reciprocal of wave speed. $c^{-1} \times 10^{-8}$ sm^{-1}



(a) Fixed source locations

(b) Fixed total number of measurements

Figure 5.8: Absolute error of inversion procedures for (a) measurements from fixed source locations and (b) a fixed total number of measurements.

the row- and null-spaces of the full non-linear inversion, without having to calculate the Jacobian matrix for subsequent iterations (the Jacobian $J(\mathbf{m}_i)$ changes with each iteration).

In a similar fashion to our SVD analysis, with an aim to determine the information content of data and the extent to which an image was fully resolved, Meles et al [111] formed a cumulative sensitivity plot from Jacobian matrices to help to gain understanding about what a system may be capable of imaging. If there is no sensitivity in an area, then one cannot detect an object there. However, an object may lie in an area of sensitivity, but still lie (partially) in the nullspace of J_{eff}, since the singular vectors $\mathbf{v}_i$ above the noise level may not be able to adequately represent the shape of the object or its contrast against the surrounding ground. This is where our analysis differs, as we have also explored these singular vectors, as did Silvestrov and Tcheverda [157] when considering appropriate model parametrisation for a seismic cross-borehole experiment.

Our analysis fixed the maximum source-receiver offset at 20 cm, and it would be interesting to see how the minimal offset affects our results. We also considered only a simple test problem with two square objects. This was useful for us since one of our aims is to use FWI to distinguish objects through additional quantitative information, but other (more realistic and complex) domains should be considered. Further, we have neglected to include antenna characteristics, as well as simplifying our forward map to the 2D Helmholtz equation, and so must be careful in how we apply any conclusions to the design of a real GPR system. To fully utilise SVD as a tool to aid designing an acquisition system, we ought to work with the 3D Maxwell equations, including antenna characteristics as well as different layouts. That said, if we find poor results in this simplified case then we cannot reasonably expect better results in the more complex 3D Maxwell case, since we will be adding more degrees of freedom to the inversion. There is therefore little point in studying a bi-static system for FWI, but it would be interesting to continue the study with small-scale arrays.

5.6 Conclusions

We presented an SVD analysis of Jacobian matrices for acquisition systems with 1 to 4 receive antennas. This was used as to analyse the ill-posedness of the FWI problem for the resulting data set, as well as being a tool for analysing the ability of such a system to image objects of interest (or, equivalently, to explore their nullspaces and inability to image). The method can readily be applied to the analysis of other acquisition systems, provided calculation of the Jacobian matrix is not prohibitive.

In our analysis, when we fixed the source locations for the different acquisition systems, the decay of singular values showed Jacobians for systems with more receivers have a greater rank (up to a given noise level), or equivalently the system is better conditioned. However, when we fix the total number of measurements (within our hand-held limitations of up to 4 receivers in a 20 cm maximum offset), the difference between the decays of singular values for all systems is almost negligible. Since these Jacobian matrices then have approximately the same rank and condition, this implies that we ought to be able to compensate for a lack of multi-offset data by taking measurements at more locations.

When we include noise though, regardless of how many measurements we are able to take, we still have a large nullspace of the resulting effective Jacobian. It is therefore important not just to look at the singular values to analyse the condition of a system, but also to analyse the span of singular vectors above the noise level. The results of projections of our objects of interest onto these singular vectors imply that, when we have at least some degree of multi-static data (within our imposed hand-held limits), we can compensate for a having fewer offsets by taking more measurements. This is not the case for the bi-static system which, despite having approximately the same rank and condition, is unable to destinguish the objects. These results are backed up anecdotally by our inversion results, and we see that a small scale multi-static array ought to be sufficient to gain quantitative information able to distinguish objects, though not necessarily fully resolve a more realistic domain.

This highlights the fact that it is not just having more singular vectors available to span our image space that are important (i.e. obtaining a better conditioned system), and taking more measurements to average over noise. Rather, better measurements are also those that result in singular vectors better able to resolve what we are interested in, as we need our data to be sufficiently sensitive to their presence. It is the latter that has led to all the multi-static systems behaving comparably to each other, and the bi-static system falling short.

Chapter 6

Modelling the forward problem in 3D

6.1 Introduction

In Chapter 4, we used a finite difference approximation of the Helmholtz equation to simulate GPR data in 2D. From Chapter 3 though, we know that the Helmholtz equation only describes electromagnetic scattering in a domain which is homogeneous in one direction (say $\hat{\mathbf{y}}$) and only for the component of electric field polarised in that direction. We also neglected any proper treatment of boundary conditions and source terms. If we wish to perform FWI on real GPR field data, we must address this and model the full 3D electromagnetic scattering problem.

In this chapter, we first derive the finite element approximation of the electromagnetic vector wave equation using linear edge (or Nédélec) elements. We then discuss truncation of an infinite subdomain by using a perfectly matched layer (PML), and model the ground transmission/reflection of electromagnetic waves by a coupling the finite element system to a boundary integral over the ground surface. Finally, we discuss a preconditioned iterative method to solve the partially dense, partially sparse coupled finite difference boundary integral (FE-BI) system, which is suitable for the scale of problem we are interested in, and give compare simulated results to some real GPR data.

6.2 Finite Element Formulation

In this section we present the finite element formulation of Maxwell's equations, following P. Monk [118,119], JL. Volakis et al [186], J. Jin [80] and N. Polydorides [136], in which the solution to weak formulation of the vector wave equation (3.5.5) is approximated numerically by a finite set of basis functions. We begin by partitioning Ω into a set of n_{el} non-overlapping elements, $\Omega_h = \cup_{i=1}^{n_{el}} \triangle_i$. The weak problem (3.5.5) is then approximated on Ω_h by: find $\mathbf{E}_h \in X_h$ such that

$$\int_{\Omega_h} \left(\frac{1}{\mu_r} (\nabla \times \mathbf{F}) \cdot (\nabla \times \mathbf{E}_h) - k_0^2 \epsilon_r \mathbf{F} \cdot \mathbf{E}_h \right) dV = \mathrm{i} k_0 Z_0 \oint_{\partial\Omega_h} \mathbf{F} \cdot (\hat{\mathbf{n}} \times \mathbf{H})\, dS \quad (6.2.1)$$

for all $\mathbf{F} \in Y_h$, where $X_h \subset X$, $Y_h \subset Y$ are finite dimensional subsets of X and Y given in (3.5.6).

Letting Ω_h be a partitioning of Ω into tetrahedral elements, we choose for X_h, Y_h the standard basis of linear edge shape elements $\{\mathbf{L}_i\}_{i=1}^{n_e}$, namely Neédélec (edge) elements [122], which were first described by Whitney [197]. Here n_e is the number of edges of tetrahedra in Ω_h, and the $\mathbf{L}_i \in H^1_{\mathrm{curl}}(\Omega_h)$ are given by

$$\mathbf{L}_i = l_i (N_m^i \nabla N_n^i - N_n^i \nabla N_m^i) \qquad i = 1 : n_e. \quad n, m \in \mathcal{V}. \quad (6.2.2)$$

Here, l_i is the length of the i^{th} edge, $\mathcal{V}$ is the set of indices for the nodes in the model, and N_m^i, N_n^i are the definitions of the nodal shape functions at the two vertices across the edge. In local nodal numbering on the k^{th} element in Ω_h, these scalar-valued nodal shape functions are defined as

$$N_1(x,y,z) = \frac{1}{6V} \begin{vmatrix} 1 & x & y & z \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}, \qquad N_2(x,y,z) = \frac{1}{6V} \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x & y & z \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix},$$

$$N_3(x,y,z) = \frac{1}{6V} \begin{vmatrix} 1 & x_1 & y_2 & z_3 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x & y & z \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}, \qquad N_1(x,y,z) = \frac{1}{6V} \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x & y & z \end{vmatrix},$$

where V is the volume of the k^{th} tetrahedron. It is easy to show that the basis functions $\mathbf{L}_i$ have a constant tangential component upon the edge on which they are defined, and that they are divergence-free [80, pp. 295]. Higher order Nédélec elements are possible, see for example Monk [126], Ainsworth and Coyle [4], and Schoberl [150].

The solution to (6.2.1) is then written as

$$\mathbf{E}_h = \sum_{i=1}^{n_e} E_i \mathbf{L}_i, \tag{6.2.3}$$

where $\mathbf{E}_h \in H^1_{\text{curl}}(\Omega_h)$, E_i is the tangential component of the electric field on the i^{th} edge. To solve (6.2.1), we must evaluate the integrals over each tetrahedral element, which gives the local contribution to the stiffness and mass matrices from tetrahedra e

$$K^e_{ij} = \int_{V^e} (\nabla \times \mathbf{L}^e_i) \cdot (\nabla \times \mathbf{L}^e_j)\, dV \tag{6.2.4}$$

$$M^e_{ij} = \int_{V^e} \mathbf{L}^e_i \cdot \mathbf{L}^e_j\, dV. \tag{6.2.5}$$

Here, $\mathbf{L}^e_i$ is the i^{th} edge shape function on tetrahedra e (i.e. using local numbering). Carrying out integrals (6.2.4) and (6.2.5), we are left with n_e local mass and stiffness matrices, the elements of which must be assembled into the global mass and stiffness matrices (i.e. using the global element numbering). See e.g. Howard, Silvester and Wathen [53] for a description of the assembly procedure, as well as a more general introduction to finite element methods. Additional care must be taken when assembling elemental contributions to ensure the local edge orientation is accounted for to avoid sign conflicts, which is not necessary for nodal elements.

The curl of the linear edge elements are given by

$$\begin{aligned}\nabla \times \mathbf{L}^e_i &= 2 l^e_i \nabla N^e_{i_1} \times \nabla N^e_{i_2} \\ &= \frac{l^e_i}{(6V^e)^2}\left[(c^e_{i_1} d^e_{i_2} - d^e_{i_1} c^e_{i_2})\hat{\mathbf{x}} + (d^e_{i_1} b^e_{i_2} - b^e_{i_1} d^e_{i_2})\hat{\mathbf{y}} + (b^e_{i_1} c^e_{i_2} - c^e_{i_1} b^e_{i_2})\hat{\mathbf{z}}\right],\end{aligned} \tag{6.2.6}$$

where $N^e_{i_j}$ is the nodal shape function for the j$^{\text{th}}$ node of edge shape function i on tetrahedra e, and the $b.$, $c.$ and $d.$ are the coefficients of x, y and z in the nodal elements, respectively. substituting into (6.2.4) results in

$$\begin{aligned}K^e_{ij} = \frac{l^e_i l^e_j V^e}{(6V^e)^4}&[(c^e_{i_1} d^e_{i_2} - d^e_{i_1} c^e_{i_2})(c^e_{j_1} d^e_{j_2} - d^e_{j_1} c^e_{j_2}) \\ &+ (d^e_{i_1} b^e_{i_2} - b^e_{i_1} d^e_{i_2})(d^e_{j_1} b^e_{j_2} - b^e_{j_1} d^e_{j_2}) \\ &+ (b^e_{i_1} c^e_{i_2} - c^e_{i_1} b^e_{i_2})(b^e_{j_1} c^e_{j_2} - c^e_{j_1} b^e_{j_2})].\end{aligned} \tag{6.2.7}$$

For the mass matrices (6.2.5), we have that

$$\mathbf{L}_i^e \cdot \mathbf{L}_j^e = \frac{l_i^e l_j^e}{(6V^e)^2}[N_{i_1}^e N_{j_1}^e f_{i_2 j_2} - N_{i_1}^e N_{j_2}^e f_{i_2 j_1} - N_{i_2}^e N_{j_1}^e f_{i_1 j_2} + N_{i_2}^e N_{j_2}^e f_{i_1 j_1}], \tag{6.2.8}$$

where $f_{ij} = b_i^e b_j^e + c_i^e c_j^e + d_i^e d_j^e$. The expressions for M_{ij}^e are straightforward to compute and implement, though all 21 combinations of i and j are rather involved so we refer the reader to [80, pp. 301-302]. Alternatively, the evaluation of (6.2.4) and (6.2.5) can be carried out numerically, with the result being exact for a 2nd order scheme (since the basis functions vary linearly).

6.3 Verifying the finite element system

We verify our forward solver is working correctly by comparing the numerical to the analytic result of a plane wave,

$$\mathbf{E}_{\text{exact}} = \mathbf{p}\mathrm{e}^{\mathrm{i}\mathbf{k}_0 \cdot \mathbf{x}}, \tag{6.3.1}$$

where $\mathbf{p}$ is the electric wave polarization, $\mathbf{k}_0$ is the free-space vector wavenumber with $|\mathbf{k}_0| = k_0$ and $\mathbf{p} \perp \mathbf{k}_0$. Plane waves are simulated in a cubic domain by applying the Neumann boundary condition

$$\hat{\mathbf{n}} \times \mathbf{H} = \hat{\mathbf{n}} \times \left(\frac{\mathrm{i}}{\omega \mu_0} \nabla \times (\mathbf{p}\mathrm{e}^{\mathrm{i}\mathbf{k}_0 \cdot \mathbf{x}}) \right), \quad \mathbf{x} \in \partial\Omega. \tag{6.3.2}$$

Our FE solution should converge linearly in the $H^1_{\text{curl}}(\Omega)$ norm with edge-length h [118],

$$\|\mathbf{e}\|_{\text{curl}} \leq Ch\|\mathbf{E}_{\text{exact}}\|_{\text{curl}} \tag{6.3.3}$$

where C is a constant which does not depend on h, and the $H^1_{\text{curl}}(\Omega)$ norm is given by

$$\|\mathbf{e}\|_{\text{curl}} = \sqrt{\|\mathbf{e}\|^2_{L^2} + \|\nabla \times \mathbf{e}\|^2_{L^2}}. \tag{6.3.4}$$

Here, $\|\cdot\|_{L^2}$ is the $(L^2(\Omega))^3$ function norm over the domain Ω,

$$\|\mathbf{e}(\mathbf{x})\|^2_{L^2} = \int_\Omega |\mathbf{e}(\mathbf{x})|^2 \,\mathrm{d}V, \tag{6.3.5}$$

and $|\cdot|$ the Euclidean norm of vectors in $\mathbb{C}^3$. (6.3.5) can be calculated numerically by Gauss integration over the discrete space Ω_h.

To test convergence initial tetrahedral meshes were generated using Netgen [149], which were then progressively refined, and the relative error between the FE and analytic solution calculated in the $H^1(\text{curl}, \Omega)$ norm calculated for each mesh. The standard uniform tetrahedral mesh refinement involves halving each edge in the mesh, dividing each tetrahedron is divided into 8 new tetrahedral volumes: 1 on each corner, and the remaining octahedron in the centre divided into a further 4 [125]. There are three possible diagonals which can divide the central octahedron, and hence three possible sets of 4 tetrahedrons, and the diagonal must be chosen carefully to ensure non-degeneracy [163]. For the purposes of this test, we make the assumption that the tetrahedral meshes produced by Netgen will be well conditioned, and so tetrahedra are close to equilateral (so that the 4 diagonals have approximately the same length), and then always chose the first diagonal node numbering for ease of implementation.

Figure 6.1 shows the convergence of the FE solution to the analytic plane wave with $1/h$ for three different starting meshes. Since the meshes produced by Netgen are unstructured, and so the tetrahedra do not all have a common edge length, we defined h as the maximum edge length in the whole mesh. The initial meshes used had a maximum edge length of $h_0 = 0.165, 0.111$ and 0.088 respectively. The straight lines of best fit on the log-scaled graphs had gradients of $\alpha = -0.95, -0.96$ and -0.98 respectively: just sub-optimal convergence. We expect this is largely due to the non-optimal refinement of the mesh (i.e. not selecting the best diagonal of the octahedron). We do not have the super-convergence often observed when refining a structured mesh, for example [77, 118].

6.4 Perfectly matched layers

A perfectly matched layer (PML) is an interface between two half-spaces, one of which is lossy, which does not reflect a plane wave for all frequencies and angles of incidence. Such a layer, which will absorb but not reflect incoming waves, can then be used to artificially truncate an unbounded domain. The concept was first proposed by Berenger [24], who found it gave superior numerical results compared with several other methods of truncating an infinite domain. In this section we show how such a layer can be described either by a change of coordinates or as an anisotropic absorber, for us to use to truncate the infinite subsurface.

First, we follow Chew and Weedon [38] and Jin [80, pp 375-406] to derive a PML

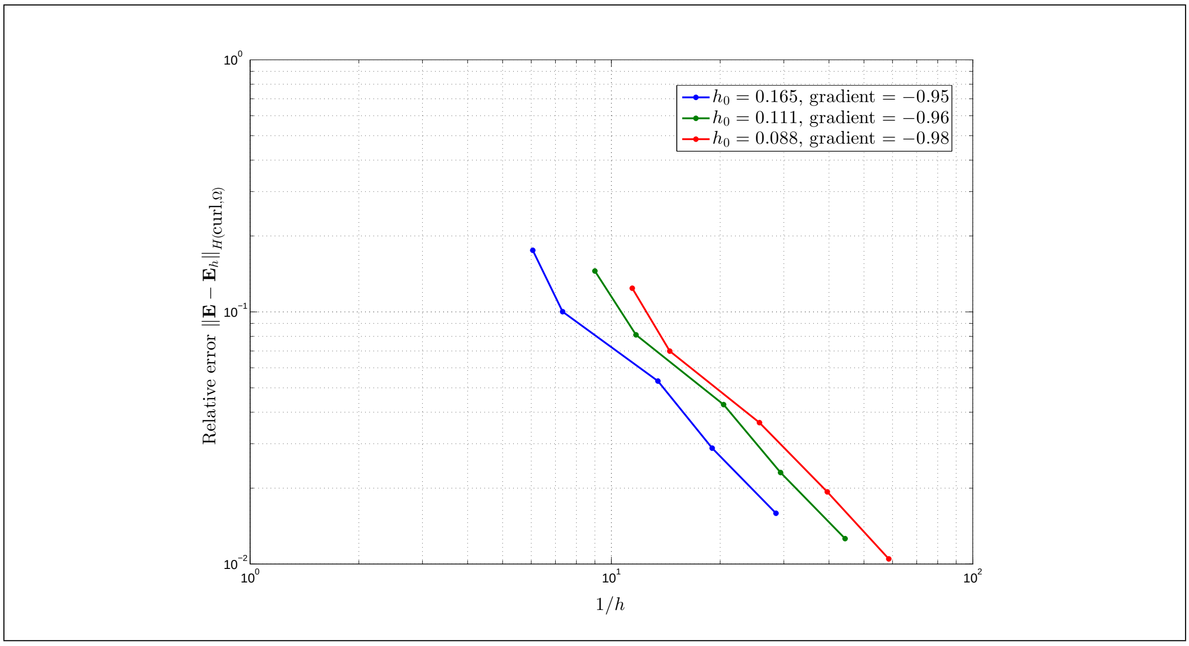


Figure 6.1

based on the change of co-ordinates

$$\begin{aligned} x_s &= \int_0^x s_x(x')\,\mathrm{d}x', \\ y_s &= \int_0^y s_y(y')\,\mathrm{d}y', \\ z_s &= \int_0^x s_z(z')\,\mathrm{d}z'. \end{aligned} \tag{6.4.1}$$

Under this change of variables, we have

$$\frac{\partial}{\partial x_s} = \frac{1}{s_x}\frac{\partial}{\partial x}, \qquad \frac{\partial}{\partial y_s} = \frac{1}{s_y}\frac{\partial}{\partial y}, \qquad \frac{\partial}{\partial z_s} = \frac{1}{s_z}\frac{\partial}{\partial z}, \tag{6.4.2}$$

and so Maxwell's equations become

$$\nabla_s \times \mathbf{E} = -\mathrm{i}\omega\mu\mathbf{H}, \tag{6.4.3a}$$

$$\nabla_s \times \mathbf{H} = \mathrm{i}\omega\epsilon\mathbf{E}, \tag{6.4.3b}$$

$$\nabla_s \cdot (\epsilon\mathbf{E}) = 0 \tag{6.4.3c}$$

$$\nabla_s \cdot (\mu\mathbf{H}) = 0, \tag{6.4.3d}$$

where

$$\nabla_s = \hat{\mathbf{e}}_x \frac{1}{s_x}\frac{\partial}{\partial x} + \hat{\mathbf{e}}_y \frac{1}{s_y}\frac{\partial}{\partial y} + \hat{\mathbf{e}}_z \frac{1}{s_z}\frac{\partial}{\partial z}. \tag{6.4.4}$$

So, ∇_s is the standard ∇ operator in Cartesian co-ordinates stretched by a factor of s_x, s_y and s_z in the x, y and z directions, respectively.

To show that this describes a lossy medium, consider a plane wave within such a PML region, which has electric and magnetic fields

$$\mathbf{E} = \mathbf{E}_0 \mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{r}} \tag{6.4.5}$$

$$\mathbf{H} = \mathbf{H}_0 \mathrm{e}^{-\mathrm{i}\mathbf{k}\cdot\mathbf{r}}, \tag{6.4.6}$$

$\mathbf{k} = k_x\hat{\mathbf{e}}_x + k_y\hat{\mathbf{e}}_y + k_z\hat{\mathbf{e}}_z$. Substituting into the Maxwell's equations for the stretched co-ordinate system (6.4.3) gives

$$\mathbf{k}_s \times \mathbf{E} = \omega\mu\mathbf{H}, \tag{6.4.7a}$$

$$\mathbf{k}_s \times \mathbf{H} = -\omega\epsilon\mathbf{E}, \tag{6.4.7b}$$

$$\mathbf{k}_s \cdot \mathbf{E} = 0, \tag{6.4.7c}$$

$$\mathbf{k}_s \cdot \mathbf{H} = 0, \tag{6.4.7d}$$

where

$$\mathbf{k}_s = \hat{\mathbf{e}}_x \frac{k_x}{s_x} + \hat{\mathbf{e}}_y \frac{k_y}{s_y} + \hat{\mathbf{e}}_z \frac{k_z}{s_z}. \tag{6.4.8}$$

Taking the cross product of (6.4.7a) with $\mathbf{k}_s$,

$$\mathbf{k}_s \times (\mathbf{k}_s \times \mathbf{E}) = \omega\mu\mathbf{k}_s \times \mathbf{H} = -\omega^2\mu\epsilon\mathbf{E}, \tag{6.4.9}$$

and by the identity $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$ and (6.4.7c) we have

$$(\mathbf{k}_s \cdot \mathbf{k}_s)\mathbf{E} = \omega^2\mu\epsilon\mathbf{E}. \tag{6.4.10}$$

This gives the dispersion relation

$$\mathbf{k}_s \cdot \mathbf{k}_s = \omega^2\mu\epsilon = \kappa^2, \tag{6.4.11}$$

the solution to which is

$$k_x = \kappa s_x \sin\theta \cos\varphi \tag{6.4.12a}$$

$$k_y = \kappa s_y \sin\theta \sin\varphi \tag{6.4.12b}$$

$$k_z = \kappa s_z \cos\theta, \tag{6.4.12c}$$

where θ) and φ are the polar and azimuthal angles in spherical polar coordinates. We note that if s_i is a complex number, the wave will be attenuated in the $\hat{\mathbf{e}}_i$ direction, and the attenuation in each direction can be independently controlled.

To show that this medium does not cause a reflection, consider again a plane wave of arbitrary polarisation, this time obliquely incident on the interface $z = 0$ in the stretched coordinate system. This wave can be decomposed into the sum of a wave with electric field transverse to $\hat{\mathbf{z}}$, (TE^z), and one with magnetic field transverse to $\hat{\mathbf{z}}$, (TM^z). For the TE^z case, the incident, reflected and transmitted fields are

$$\mathbf{E}^{\mathrm{in}} = \mathbf{E}_0 \mathrm{e}^{-\mathrm{i}\mathbf{k}^{\mathrm{in}}\cdot\mathbf{r}} \tag{6.4.13a}$$

$$\mathbf{E}^{\mathrm{r}} = R^{\mathrm{TE}} \mathbf{E}_0 \mathrm{e}^{-\mathrm{i}\mathbf{k}^{\mathrm{r}}\cdot\mathbf{r}} \tag{6.4.13b}$$

$$\mathbf{E}^{\mathrm{t}} = T^{\mathrm{TE}} \mathbf{E}_0 \mathrm{e}^{-\mathrm{i}\mathbf{k}^{\mathrm{t}}\cdot\mathbf{r}}, \tag{6.4.13c}$$

where $\mathbf{E}_0$ is a constant vector perpendicular to $\hat{\mathbf{e}}_z$, and R^{TE} and T^{TE} are the reflection and transmission coefficients. By tangential continuity boundary conditions (3.3.2), we have

$$R^{\mathrm{TE}} = \frac{k_{1z} s_{2z} \mu_2 - k_{2z} s_{1z} \mu_1}{k_{1z} s_{2z} \mu_2 + k_{2z} s_{1z} \mu_1}. \tag{6.4.14}$$

Here, the subscripts 1 and 2 indicate the upper and lower half-spaces respectively. From phase-matching conditions on the interface $k_{1x} = k_{2x}$ and $k_{1y} = k_{2y}$, we have

$$\kappa_1 s_{1x} \sin\theta_1 \cos\varphi_1 = \kappa_2 s_{2x} \sin\theta_2 \cos\varphi_2 \tag{6.4.15a}$$

$$\kappa_1 s_{1y} \sin\theta_1 \sin\varphi_1 = \kappa_2 s_{2y} \sin\theta_2 \sin\varphi_2. \tag{6.4.15b}$$

$$\tag{6.4.15c}$$

Choosing $\epsilon_1 = \epsilon_2$, $\mu_1 = \mu_2$, $s_{1x} = s_{2x}$ and $s_{1y} = s_{2y}$, then we have $\theta_1 = \theta_2$ and $\varphi_1 = \varphi_2$, and can show that $R^{\mathrm{TE}} = 0$, and similarly $R^{\mathrm{TM}} = 0$. This holds regardless of the choice of s_{1z} and s_{2z}, as well as the choice of frequency and the angle of incidence. For complex s_{2z}, we therefore have a layer which is lossy in the $\hat{\mathbf{z}}$ direction, but which is reflectionless for waves incident obliquely to the interface from above.

Above, we described a PML which absorbs waves in the $\hat{\mathbf{z}}$ direction. In order to truncate an infinite 3D domain with a PML, the choice of PML parameters s_ζ depends on position. For a PML perpendicular to the $\hat{\mathbf{x}}$ axis,

$$s_x = s'_x - \mathrm{i}s''_x, \tag{6.4.16}$$

and for a PML perpendicular to the $\hat{\mathbf{y}}$ axis

$$s_y = s'_y - \mathrm{i}s''_y. \tag{6.4.17}$$

Wherever the PML is not active in a direction $\zeta = x, y, z$, $s_\zeta = 1$. This way, the PML can then be attenuating in several directions (i.e. in the corners and edges of the 3D domain).

Consider an anisotropic medium with permittivity and permeability

$$\bar{\epsilon} = \begin{bmatrix} \epsilon_{xx} & 0 & 0 \\ 0 & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_{zz} \end{bmatrix} \qquad \bar{\mu} = \begin{bmatrix} \mu_{xx} & 0 & 0 \\ 0 & \mu_{yy} & 0 \\ 0 & 0 & \mu_{zz} \end{bmatrix}. \tag{6.4.18}$$

Let us denote the solution to the Maxwell equations for such a medium as $\mathbf{E}^{\text{an}}$ and $\mathbf{H}^{\text{an}}$, and choose $\bar{\epsilon}$ and $\bar{\mu}$ to be

$$\bar{\epsilon} = \epsilon\bar{\Lambda} \qquad \bar{\mu} = \mu\bar{\Lambda}, \tag{6.4.19}$$

where

$$\bar{\Lambda} = \begin{bmatrix} \frac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_z s_x}{s_y} & 0 \\ 0 & 0 & \frac{s_x s_y}{s_z} \end{bmatrix}. \tag{6.4.20}$$

It can then readily be shown that

$$E_\zeta^{\text{an}} = s_\zeta E_\zeta^{\text{sc}}, \qquad H_\zeta^{\text{an}} = s_\zeta H_\zeta^{\text{sc}}, \tag{6.4.21}$$

where $\zeta = x, y, z$, and $\mathbf{E}^{\text{sc}}$ and $\mathbf{E}^{\text{sc}}$ are the solutions to the Maxwell equations in the stretched co-ordinate system (6.4.3) with permittivity and permeability ϵ and μ. Hence we can interpret a PML as an anisotropic absorbing material, with it being first derived in this way by Sacks et al [143].

An anisotropic absorber PML is referred to as a uniaxial PML (UPML), and its numerical implementation is particularly straightforward where the domain discretisation is aligned with the coordinate system – e.g. for rectangular brick elements. However, Shin and Fan [155] found that in many applications, implementing the SC-PML is more favourable due to resulting better conditioned system matrices, and therefore more accurate solutions with direct solvers and faster convergence for iterative ones.

For ease of implementation, we have used a UPML backed by a PEC surface (natural boundary conditions). More robust choices can be made, such as first or second order absorbing boundary conditions [80, pp. 387-390], which may allow a smaller PML region to be used for the same attenuation. Also note that while the

PML is reflectionless in the continuous case described above, when we discretise a *perfectly-matched* interface does not exist due to discretisation errors and so may cause reflection [37]. To reduce this effect, we increase the lossy coefficients s'' gradually over a short distance at the start of the PML region.

6.5 Boundary conditions and sources above the ground

6.5.1 Introduction

In Section 6.2, we have prescribed the boundary condition

$$\hat{\mathbf{n}} \times \hat{\mathbf{H}} = \mathbf{g} \qquad \mathbf{r} \in \partial\Omega.$$

For GPR, we have knowledge of sources above the domain of interest (i.e. above ground), and so must apply transmission boundary conditions (3.3.2) at the ground surface. In order to apply these conditions we couple the FEM system with a boundary element approximation of the surface equivalence theorem. In this section we present the formulation of the boundary element method, and give the analytic results required for numerical evaluation of the integration of resulting singular integrands. While we could instead have included the air in the computational region for the FEM solver, it was felt at the outset of development that the use of a boundary element would allow easier application of arbitrary source terms without the difficulty of requiring a finer discretised mesh near source locations.

6.5.2 Field Integral Equations

Recall the surface equivalence principle from Section 3.7, that *two sources that produce the same field within a region are said to be equivalent within that region.* This states that the field exterior/interior to a given and possibly fictitious surface S may be exactly represented by equivalent currents placed on that surface and allowed to radiate. The sources which produce equivalent interior fields are given in terms of the exterior electromagnetic fields as

$$\mathbf{J}_s(\mathbf{r},\omega) = \hat{\mathbf{n}} \times \mathbf{H}^{\text{ext}} \quad \text{and} \quad \mathbf{M}_s(\mathbf{r},\omega) = \mathbf{E}^{\text{ext}} \times \hat{\mathbf{n}}. \tag{6.5.1}$$

Here, the exterior fields are given by the sum of the incident, reflected (directly from the surface) and scattered (from interactions in the subsurface) fields,

$$\begin{aligned} \mathbf{E}^{\mathrm{ext}} &= \mathbf{E}^{\mathrm{inc}} + \mathbf{E}^{\mathrm{r}} + \mathbf{E}^{\mathrm{sc}}, \\ \mathbf{H}^{\mathrm{ext}} &= \mathbf{H}^{\mathrm{inc}} + \mathbf{H}^{\mathrm{r}} + \mathbf{H}^{\mathrm{sc}}. \end{aligned} \tag{6.5.2}$$

Recall the radiated fields due to these equivalent currents are given by the surface equivalence principle

$$\begin{aligned} \mathbf{E}^{\mathrm{ext}} &= -\oint_S \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{M}_s \, \mathrm{d}S \\ &\quad + \mathrm{i}k_0 Z_0 \oint_S \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}_s \, \mathrm{d}S, \end{aligned} \tag{6.5.3a}$$

$$\begin{aligned} \mathbf{H}^{\mathrm{ext}} &= -\oint_S \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}_s \, \mathrm{d}S \\ &\quad + \mathrm{i}k_0 Y_0 \oint_S \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{M}_s \, \mathrm{d}S. \end{aligned} \tag{6.5.3b}$$

Having an equivalence relation for harmonic fields and current sources, we are seeking to apply the boundary condition

$$\hat{\mathbf{n}} \times \mathbf{H} = \hat{\mathbf{n}} \times \frac{1}{\mu_r} \nabla \times \mathbf{E} = -\mathrm{i}k_0 \mathbf{J}_s, \qquad r \in \partial\Omega \tag{6.5.4}$$

on the ground surface $\partial\Omega$. However, the total electric field on the boundary is not known a-priori since it is the sum of the incident, reflected and scattered fields, and only the incident and reflected can be calculated trivially. Following Volakis et al [186, pp. 229-233] (see also [80,134,136,187,188]), we cast the unknown quantities $\mathbf{E}^{\mathrm{sc}}$ and $\mathbf{H}^{\mathrm{sc}}$ as a boundary integral equation, obtaining

$$\begin{aligned} -\hat{\mathbf{n}} \times [\mathbf{H}^{\mathrm{in}} + \mathbf{H}^{\mathrm{r}}] &= -\frac{\mathbf{J}_s}{2} - \mathrm{P.V.} \oint_S \hat{\mathbf{n}} \times [\nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}_s] \, \mathrm{d}S' \\ &\quad + \mathrm{i}k_0 Y_0 \oint_S \hat{\mathbf{n}} \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{M}_s \, \mathrm{d}S' \end{aligned} \tag{6.5.5a}$$

$$\begin{aligned} -\hat{\mathbf{n}} \times [\mathbf{E}^{\mathrm{in}} + \mathbf{E}^{\mathrm{r}}] &= \frac{\mathbf{M}}{2} - \mathrm{P.V.} \oint_S \hat{\mathbf{n}} \times [\nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{M}] \, \mathrm{d}S' \\ &\quad + \mathrm{i}k_0 Z_0 \oint_S \hat{\mathbf{n}} \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}_s(\mathbf{r}') \, \mathrm{d}S' \end{aligned} \tag{6.5.5b}$$

where P.V. is the Cauchy principal value. Equation (6.5.5a) is referred to as the Magnetic Field Integral Equation (MFIE), and (6.5.5b) the Electric Field Integral Equation (EFIE). Here, the first terms on the right hand side are due to the identity [186, pp 230]

$$\oint_S \hat{\mathbf{n}} \times \left[\nabla \left(\frac{\mathrm{e}^{-\mathrm{i}k_0 R}}{4\pi R} \right) \times \mathbf{J}_s \right] ds = \frac{1}{2}\mathbf{J}_s + \mathrm{P.V.} \oint_S \hat{\mathbf{n}} \times \left[\nabla \left(\frac{\mathrm{e}^{-\mathrm{i}k_0 R}}{4\pi R} \right) \times \mathbf{J}_s \right] ds. \tag{6.5.6}$$

The principle value is not necessary since the evaluation of this integral with $\mathbf{r}$ on S does not produce a singularity, and the 1/2 value can be obtained without invoking the principal value theorem by placing $\mathbf{r}$ slightly off of S and taking the limit as $\mathbf{r}$ approaches S [186, pp. 230-231]. Note that we included the reflected terms as knowns in the integral equations above. If we do not know them, they can be neglected and will be included within the scattered fields on solution of the integral equations [186, pp. 230].

It is well known that both the MFIE and EFIE can suffer spurious internal resonance problems at resonant frequencies of the enclosed volume [131, 190], as well as having different characteristics such as accuracy of solution, convergence rate, low-frequency breakdown or inaccuracy etc. This problem can be alleviated by taking some linear combination of the two, referred to as the combined field integral equation (CFIE), such as

$$\text{TETH} = (1-\alpha)\text{MFIE} + \frac{\alpha}{Z_0}\text{EFIE}, \tag{6.5.7}$$

where α is a parameter between 0 and 1. (6.5.7) is referred to as the TETH (tangential E tangential H) formulation of the CFIE equation, which Sheng et al [153] find does not completely remove the problem of interior resonances but reduces the problem to occur over a very narrow frequency range. Other possible linear combinations include

$$\begin{aligned}\text{TENH} =& \alpha_1\text{EFIE} + \alpha_2\hat{\mathbf{n}} \times \text{MFIE},\\ \text{NETH} =& \alpha_1\hat{\mathbf{n}} \times \text{EFIE} + \alpha_2\text{MFIE},\\ \text{NENH} =& \alpha_1\hat{\mathbf{n}} \times \text{EFIE} + \alpha_2\hat{\mathbf{n}} \times \text{MFIE},\end{aligned} \tag{6.5.8}$$

where α_1 and α_2 are scaling parameters, as well further additions of NE ($\hat{\mathbf{n}} \times \mathbf{E}$) or NH ($\hat{\mathbf{n}} \times \mathbf{H}$). Sheng et al [153] find that the TENH formulation yields the best conditioned finite element matrices, so is most suitable for iterative solvers, and NETH the worst. Further, they find that TENH and NETH overcome the problems of interior resonance, which TETH and NENH do not although only suffer the problem in a very narrow bandwidth. Finally, they that the accuracy of (6.5.7) and (6.5.8) can all be improved by including both TE and NE formulations, and found that a TENENH formulation (with $\alpha_1 = \alpha_2 = 0.45$, $\alpha_3 = 0.1$) was both accurate and efficient when using a multi-level fast multipole algorithm [161]. However in later work, Liu and Jin [100], see also Jin [80, pp 465], have found that the NETH formulation is useful in that an effective and efficient preconditioner exists based on the absorbing boundary conditions (see Section 6.8.1).

6.5.3 Variational formulation

The variational form of the MFIE is derived by by multiplying a test function $\mathbf{F} \in H^1_{\text{div}}(\partial\Omega)$, where $H^1_{\text{div}}(\partial\Omega)$ is defined in (A.1.10), and integrating over the boundary,

$$\begin{aligned} -\frac{1}{2}\oint_S \mathbf{F}\cdot\mathbf{J}_s\,\mathrm{d}S - \oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}_s(\mathbf{r}')\,\mathrm{d}S'\right\}\mathrm{d}S \\ +\mathrm{i}k_0Y_0\oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{M}_s(\mathbf{r}')\,\mathrm{d}S'\right\}\mathrm{d}S = f_i^{\text{ext}}, \end{aligned} \tag{6.5.9}$$

where the source term f_i^{ext} is given by

$$f_i^{\text{ext}} = \oint_S \mathbf{F}\cdot\hat{\mathbf{n}}\times[\mathbf{H}^{\text{in}}+\mathbf{H}^{\text{r}}]\,\mathrm{d}S. \tag{6.5.10}$$

Making use of (3.6.21) and (3.6.24) to write the MFIE in terms of the scalar Greens function, we have

$$\begin{aligned} -\frac{1}{2}\oint_S \mathbf{F}\cdot\mathbf{J}_s\,\mathrm{d}S - \oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\nabla' G_0(\mathbf{r},\mathbf{r}')\times\mathbf{J}_s(\mathbf{r}')\,\mathrm{d}S'\right\}\mathrm{d}S \\ +\mathrm{i}k_0Y_0\oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}G_0(\mathbf{r},\mathbf{r}')\mathbf{M}_s\,\mathrm{d}S'\right\}\mathrm{d}S \\ +\mathrm{i}\frac{Y_0}{k_0}\oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\nabla'\cdot\mathbf{M}_s\nabla G_0\,\mathrm{d}S'\right\}\mathrm{d}S = f_i^{\text{ext}}. \end{aligned} \tag{6.5.11}$$

In order to combine the MFIE with the FEM system, we substitute for $\mathbf{M}_s = -\hat{\mathbf{n}}\times\mathbf{E}^{\text{ext}}$, and enforce the continuity conditions (3.3.2),

$$\begin{aligned} \hat{\mathbf{n}}\times\mathbf{H}^{\text{int}} &= \hat{\mathbf{n}}\times\mathbf{H}^{\text{ext}} \qquad \text{on the surface } S \\ \hat{\mathbf{n}}\times\mathbf{E}^{\text{int}} &= \hat{\mathbf{n}}\times\mathbf{E}^{\text{ext}} \qquad \text{on the surface } S. \end{aligned} \tag{6.5.12}$$

The magnetic continuity condition is ‘natural’, and we may simply set $\mathbf{H}^{\text{ext}} = \mathbf{H}^{\text{int}}$ in (6.5.5a). The electric continuity condition must be explicitly forced, either by enforcing the relation

$$\oint_S \mathbf{F}\cdot\hat{\mathbf{n}}\times(\mathbf{E}^{\text{int}}-\mathbf{E}^{\text{ext}})\,ds = 0, \tag{6.5.13}$$

or by simply setting $\mathbf{E}^{\text{ext}} = \mathbf{E}^{\text{int}}$ in (6.5.11). The former allows us to use different test functions to represent $\mathbf{E}^{\text{ext}}$ and $\mathbf{E}^{\text{int}}$, and so may have disjoint surface and volume meshes (i.e. ones who’s vertices do not necessarily coincide), as well as different order basis functions, whereas the latter is simpler to implement. Since we wish to use linear basis functions in both cases, and the surface of our domain will not be a particularly complex shape (in general, a cuboid with the top surface somewhat distorted), we choose the latter. This gives the weak form of the MFIE, which is to find $\mathbf{E}^{\text{in}}, \mathbf{J}_s \in H^1_{\text{div}}(\partial\Omega)$ such that

$$\begin{aligned} -\frac{1}{2}\oint_S \mathbf{F}\cdot\mathbf{J}_s\,\mathrm{d}S - \oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\nabla' G_0(\mathbf{r},\mathbf{r}')\times\mathbf{J}_s(\mathbf{r}')\,\mathrm{d}S'\right\}\,\mathrm{d}S \\ -\mathrm{i}k_0Y_0\oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'} G_0(\mathbf{r},\mathbf{r}')(\hat{\mathbf{n}}\times\mathbf{E}^{\text{int}})\,\mathrm{d}S'\right\}\,\mathrm{d}S \\ +\mathrm{i}\frac{Y_0}{k_0}\oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\nabla'\cdot(\hat{\mathbf{n}}\times\mathbf{E}^{\text{int}})\nabla' G_0\,ds'\right\}\,\mathrm{d}S = f_i^{\text{ext}}, \end{aligned} \tag{6.5.14}$$

for all $\mathbf{F} \in H^1_{\text{div}}(\partial\Omega)$. The weak formulation of the EFIE, similarly, is to find $\mathbf{E}^{\text{in}}, \mathbf{J}_s \in H^1_{\text{div}}(\partial\Omega)$ such that

$$\begin{aligned} -\frac{1}{2}\oint_S \mathbf{F}\cdot(\hat{\mathbf{n}}\times\mathbf{E}^{\text{int}})\,\mathrm{d}S + \oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\nabla' G_0(\mathbf{r},\mathbf{r}')\times(\hat{\mathbf{n}}\times\mathbf{E}^{\text{int}})\,\mathrm{d}S'\right\}\,\mathrm{d}S \\ +\mathrm{i}k_0Z_0\oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'} G_0(\mathbf{r},\mathbf{r}')\mathbf{J}_s\,\mathrm{d}S'\right\}\,\mathrm{d}S \\ -\mathrm{i}\frac{Z_0}{k_0}\oint_S \mathbf{F}\cdot\left\{\hat{\mathbf{n}}\times\oint_{S'}\nabla'\cdot\mathbf{J}_s\nabla' G_0\,ds'\right\}\,\mathrm{d}S = f_i^{\text{ext}}, \end{aligned} \tag{6.5.15}$$

for all $\mathbf{F} \in H^1_{\text{div}}(\partial\Omega)$, where the forcing term in (6.5.15) is given by

$$f_i^{\text{ext}} = \oint_S \mathbf{F}\cdot\hat{\mathbf{n}}\times[\mathbf{E}^{\text{in}} + \mathbf{E}^{\text{r}}]\,\mathrm{d}S. \tag{6.5.16}$$

The weak forms of the NE and NH formulations of the E and CFIE equations (6.5.8) are given simply by taking $\hat{\mathbf{n}}\times$ equations (6.5.14) and (6.5.15). By the scalar triple product rule $\mathbf{a}\cdot\mathbf{b}\times\mathbf{c} = -\mathbf{b}\times\mathbf{a}\cdot\mathbf{c}$, this is equivalent to replacing test functions $\mathbf{F}$ with $\hat{\mathbf{n}}\times\mathbf{F}$.

6.5.4 FE-BI formulation

As with the FEM formulation of the vector wave equation, for the boundary element formulation of (6.5.14) and (6.5.15) we discretise $\partial\Omega$ into a set of triangular elements, and approximate the infinite dimensional space $H^1_{\text{div}}(\partial\Omega)$ with a finite dimensional subspace $H^1_{\text{div}}(\partial\Omega_h)$. In this model, the numerical solution to the surface currents $\hat{\mathbf{n}}\times\mathbf{H} = \mathbf{J}_s$ and $\hat{\mathbf{n}}\times\mathbf{E}^{\text{ext}}|_{\partial\Omega} = \mathbf{M}_s$ are approximated by a linear combination of vector basis functions $\mathbf{f}_n$, such that

$$\mathbf{J}_{sh}(\mathbf{r}) = \sum_{n=1}^{n_s} J_n\mathbf{f}_n(\mathbf{r}) \qquad \text{and} \qquad \mathbf{M}_{sh}(\mathbf{r}) = \sum_{n=1}^{n_s} M_n\mathbf{f}_n(\mathbf{r}). \tag{6.5.17}$$

For the $\mathbf{f}_n$, we choose the RWG basis functions (named after its creators Rao, Wilton and Glisson [138]),

$$\mathbf{f}_n(\mathbf{r}) = \begin{cases} \frac{l_n \boldsymbol{\rho}_n^+(\mathbf{r})}{2A_n^+} & \mathbf{r} \in T_n^+ \\ \frac{l_n \boldsymbol{\rho}_n^-(\mathbf{r})}{2A_n^-} & \mathbf{r} \in T_n^- \\ 0 & \text{elsewhere} \end{cases} . \tag{6.5.18}$$

Here, l_n is the length of edge n, $A_n^\pm$ is the area of triangles $T_n^\pm$ sharing edge n, and

$$\boldsymbol{\rho}_n^+ = (\mathbf{r} - n), \qquad \boldsymbol{\rho}_n^- = (n - \mathbf{r}), \tag{6.5.19}$$

where we have abused the notation of n to mean the position of the free vertex of the triangle. We see that these basis functions have support only on the two triangles sharing edge n, and are zero elsewhere. Further, they are numerically equivalent to $\hat{\mathbf{n}} \times \mathbf{L}_n$, where the $\mathbf{L}_n$ are the linear basis functions used to represent the internal $\mathbf{E}$ field in the FEM approximation.

Substituting the approximate tangential components of the $\mathbf{E}$ and $\mathbf{H}$ fields on the surface $\partial\Omega$, (6.5.17), into the weak formulation of the MFIE (6.5.14) and EFIE (6.5.15), we arrive at the Boundary Element formulation of the CFIE integrals, which are linear combinations of

$$\begin{aligned} C_{m,n}^{(1)} &= \frac{1}{2} \oint_S \mathbf{f}_m \cdot \mathbf{f}_n \, \mathrm{d}S \\ C_{m,n}^{(2)} &= \oint_S \mathbf{f}_m \cdot \left(\hat{\mathbf{n}} \times \oint_{S'} \nabla' G_0(\mathbf{r}, \mathbf{r}') \times \mathbf{f}_n(\mathbf{r}') ds' \right) \mathrm{d}S \end{aligned} \tag{6.5.20}$$

and

$$\begin{aligned} D_{m,n}^{(1)} &= \mathrm{i}k_0 \oint_S \mathbf{f}_m(\mathbf{r}) \cdot \left(\hat{\mathbf{n}} \times \oint_{S'} G_0(\mathbf{r}, \mathbf{r}') \mathbf{f}_n(\mathbf{r}') \, \mathrm{d}S' \right) \mathrm{d}S \\ D_{m,n}^{(2)} &= \frac{\mathrm{i}}{k_0} \oint_S \mathbf{f}_m(\mathbf{r}) \cdot \left(\hat{\mathbf{n}} \times \oint_{S'} \nabla' G_0(\mathbf{r}, \mathbf{r}') \nabla' \cdot \mathbf{f}_n(\mathbf{r}') \, \mathrm{d}S' \right) \mathrm{d}S, \end{aligned} \tag{6.5.21}$$

Upon assembly and coupling with the Finite Element system, we have

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{E}_h \\ \mathbf{J}_h \end{bmatrix} = \begin{bmatrix} \phi^{\mathrm{E}} \\ \phi^{\mathrm{M}} \end{bmatrix} \tag{6.5.22}$$

Here, $\mathbf{E}_h$ and $\mathbf{J}_h$ are the coefficients of the basis functions for the $\mathbf{E}$ field and surface currents, respectively, A is the original FEM system matrix, and B represents the integral of $\hat{\mathbf{n}} \times \mathbf{H} = \mathbf{J}_s$ over $\partial\Omega$ in (6.2.1). For the TETH CFIE, the matrices C and D are given by

$$\begin{aligned} C =& \frac{\alpha}{Z_0}(C^{(1)} + C^{(2)}) + Y_0(1 - \alpha)(D^{(1)} + D^{(2)}) \\ D =& Z_0 \frac{\alpha}{Z_0}(D^{(1)} + D^{(2)}) + (1 - \alpha)(C^{(1)} + C^{(2)}), \end{aligned} \tag{6.5.23}$$

The matrices A and B are sparse, while C and D are dense (with C containing non-zeros in columns corresponding to surface elements $\mathbf{L}_i$, and zero columns elsewhere). Figure 6.2 shows the sparsity pattern of the FE-BI system matrix.

The boundary integral method is coupled with the finite element formulation over a finite aperture of the ground surface $\mathcal{S} \subset \partial\Omega_h$, as shown in Figure 6.3. The boundary of S, $\partial\mathcal{S}$, is taken to be the inside boundary of the PML region, such that the PML (and discretised volume Ω_h) extends beyond the BI region, and the PML region and S do not overlap. This is a similar approach to that used by Jin and Volakis [79, 186] for simulating wave fields emitted by an antenna in a cavity recessed in an infinite metallic plane: the boundary integrals are not taken over the entire (infinite) metallic surface, where the electric field (and Green's function) is zero valued. The difference is our cavity is backed with a PML region, or (physically) an anisotropic absorbing material, which is also backed by a metallic surface for simplicity.

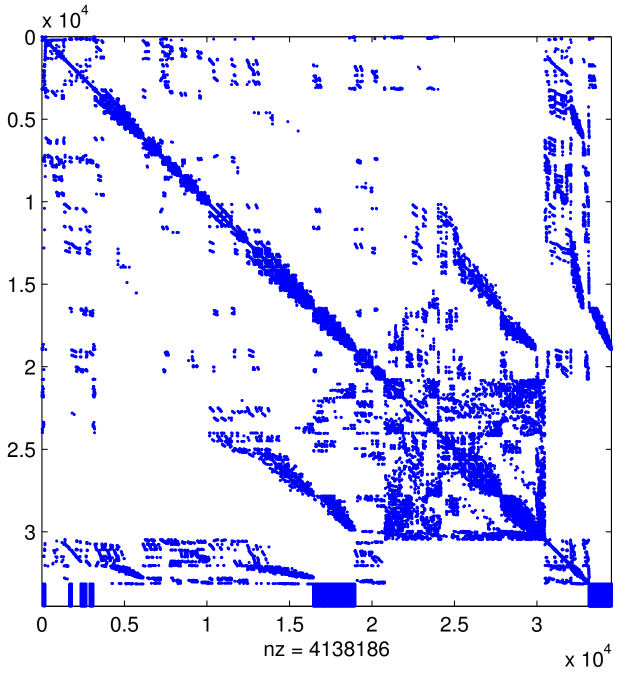


Figure 6.2: Sparsity plot of the FE-BI system matrix.

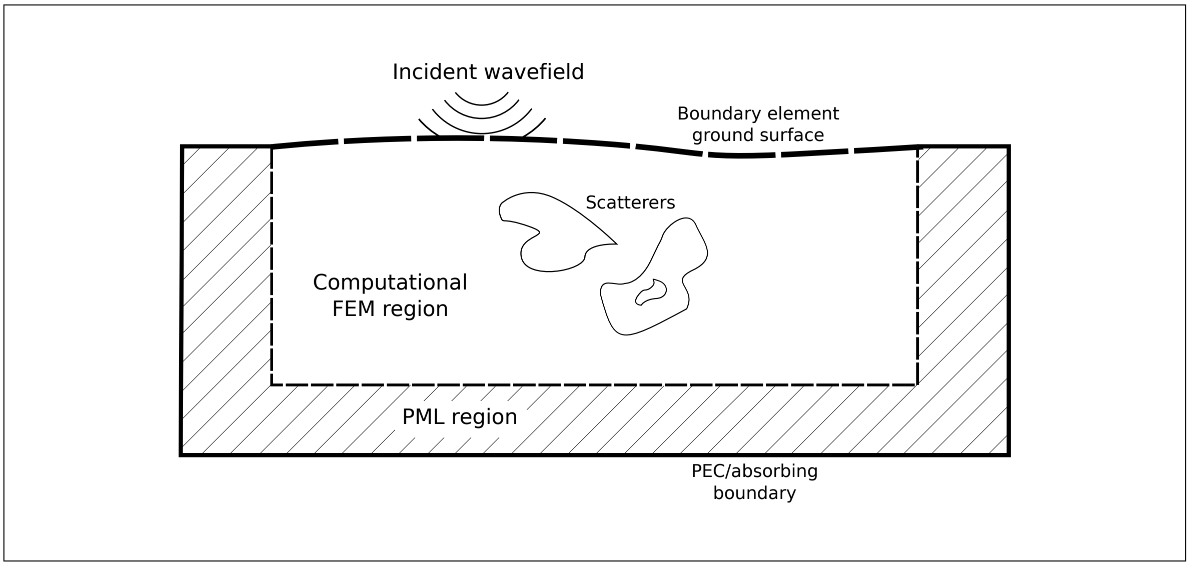


Figure 6.3: Combining the FEM formulation with a BI region on the ground surface

6.6 Integration of singular integrands

6.6.1 Introduction

Notice that the boundary element formulation of the CFIE equations involves singular integrands when $\mathbf{r}$ and $\mathbf{r}'$ are close. Here, we will define close as being when the two points lie within two surface elements which either touch or overlap, though more generally we could say when $|\mathbf{r} - \mathbf{r}'| < R$, where R is a given radius. When the integrand is singular low order numerical methods are unsuitable. One alternative method is to separate the integrand into a sum of two parts, where one is smooth enough for numerical integration, and the second is singular but has an analytic result for its integration over the surface triangle [63, 136, 200]. When $\mathbf{r}$ is not close to $\mathbf{r}'$, we may proceed to integrate numerically using Gaussian quadrature.

In this section we present the analytic results in the literature required to evaluate the integrals of singular integrands in this way, and apply the results to the D^1, D^2 and C^2 matrices.

6.6.2 Analytic results for integrals of singular integrands

Let us define a local labeling of the vertices and edges of the surface triangle S', in which the nodes are ordered clockwise when viewed from the side in which the outward normal $\hat{\mathbf{n}}$ points; $\hat{\mathbf{s}}_i$ is the counter-clockwise unit tangential vector to edge i, the edge opposite node i; and $\hat{\mathbf{m}}_i = \hat{\mathbf{s}}_n \times \hat{\mathbf{n}}$ is the outward unit normal to edge i in the plane of S'. With this notation, we define a local co-ordinate system $\mathcal{L} := (u, v, w)$ on the surface triangle S', where the origin O' is the first node of the triangle, $\hat{\mathbf{u}}$ is the unit vector $\hat{\mathbf{s}}_3$, $\hat{\mathbf{v}} = \hat{\mathbf{n}} \times \hat{\mathbf{u}}$, and $\hat{\mathbf{w}} = \hat{\mathbf{n}}$. See Figure 6.4 and Table 6.1 for variable disambiguation.

With this notation, the analytic results we require for the integration of singular integrands over surface triangle S' are [63, 136, 200]

$$\begin{aligned} \oint_{S'} R\,\mathrm{d}S' &= \frac{1}{3}\sum_{i=1}^{3} t_i^0 \oint_{\partial_i S'} R\,\mathrm{d}l' && \text{for } w_0 = 0 \\ &= \frac{1}{3}\left(w_0^2 \oint_{S'} \frac{1}{R}\,\mathrm{d}S'\right) + \frac{1}{3}\sum_{i=1}^{3} t_i^0 \oint_{\partial_i S'} R\,\mathrm{d}l' && \text{for } w_0 \neq 0, \end{aligned} \tag{6.6.1}$$

and

$$\begin{aligned} \int_{S'} \frac{1}{R}\,\mathrm{d}S' &= \sum_{i=1}^{3} t_i^0 \int_{\partial_i S'} \frac{1}{R}\,\mathrm{d}l' && \text{for } w_0 = 0 \\ &= -w_0\left(w_0 \int_{S'} \frac{1}{R^3}\,\mathrm{d}S'\right) + \sum_{i=1}^{3} t_i^0 \int_{\partial_i S'} \frac{1}{R}\,\mathrm{d}l' && \text{for } w_0 \neq 0 \end{aligned} \tag{6.6.2}$$

where the integrals over $\partial_i S'$ indicates the contour integral along edge i, t_i is the distance from O' to the projection of $\mathbf{r}$ into the plane of S',

$$w_0 \int_{S'} \frac{1}{R^3}\,\mathrm{d}S' = \operatorname{sgn}(w_0)\beta, \tag{6.6.3}$$

and

$$\beta = \sum_{i=1}^{3}\left(\arctan\left(\frac{t_i^0 s_i^-}{(R_i^0)^2 + |w_0| R_i^-}\right) + \arctan\left(\frac{t_i^0 s_i^+}{(R_i^0)^2 + |w_0| R_i^+}\right)\right). \tag{6.6.4}$$

Further, we require the integrals of the gradients of powers of R, given by

$$\int_{S'} \nabla' R\,\mathrm{d}S' = \sum_{i=1}^{3} \hat{\mathbf{m}}_i \oint_{\partial_i S'} R\,\mathrm{d}l' - \hat{\mathbf{n}}(\mathbf{r}') w_0 \int_{S'} \frac{1}{R}\,\mathrm{d}S', \tag{6.6.5}$$

$$\int_{S'} \nabla' \frac{1}{R}\,\mathrm{d}S' = \sum_{i=1}^{3} \hat{\mathbf{m}}_i \int_{\partial_i S'} \frac{1}{R}\,\mathrm{d}l' + \hat{\mathbf{n}}(\mathbf{r}') w_0 \int_{S'} \frac{1}{R^3}\,\mathrm{d}S', \tag{6.6.6}$$

$$\int_{S'} \nabla' R^3\,\mathrm{d}S' = -\sum_{i=1}^{3} \hat{\mathbf{m}}_i \int_{\partial_i S'} R^3\,\mathrm{d}l' + 3\omega_0 \hat{\mathbf{n}}(\mathbf{r}') \int_{S'} R\,\mathrm{d}S'. \tag{6.6.7}$$

Variable	Description
$\hat{\mathbf{n}}$	Outward unit normal on S
$\mathbf{r}$	Observation point on S
$\mathbf{r}'$	Integration point in S'
O	Origin
O'	Origin of local coordinates $\mathcal{L}$ on S' - i.e. node 1
w_0	Distance from $\mathbf{r}$ to integration surface S', $w_0 = (\mathbf{r} - \hat{\mathbf{n}})v$
R	Distance from $\mathbf{r}$ to $\mathbf{r}'$
$\boldsymbol{\rho}$	Vector from O' to the projection of $\mathbf{r}$ onto S', $\boldsymbol{\rho} = \mathbf{r} - w_0\hat{\mathbf{n}}$
$\boldsymbol{\rho}'$	Vector from O' to $\mathbf{r}'$
$\boldsymbol{\rho}_n$	Vector from O' to vertex n
$\boldsymbol{\rho}'_n$	Vector from vertex n to $\mathbf{r}'$
$\mathbf{R}_n$	Vector from vertex n to $\mathbf{r}'$
$\hat{\mathbf{s}}_n$	Unit vector on edge n in the direction of test function $\mathbf{f}_n$
$\hat{\mathbf{m}}_n$	Unit normal on edge n in the plane of S'
t_n^0	Distance from $\boldsymbol{\rho}$ to the edge n
$s_n^\pm$	Distance from t_n^0 to the beginning/end of edge n
$t_n^\pm$	Distance $\sqrt{(t_n^0)^2 + (s_n^\pm)^2}$
R_n^0	Distance $\sqrt{(t_n^0)^2 + w_0^2}$
$R_n^\pm$	Distance $\sqrt{(t_n^\pm)^2 + w_0^2}$

Table 6.1: Variable definitions for analytic integration of boundary integrals

Finally, the integrals along the boundaries of the surface triangle S' (i.e. edges) required to evaluate equations (6.6.1)–(6.6.6) are given by

$$\int_{\partial_i S'} \frac{1}{R}\,\mathrm{d}l' = \ln\left(\frac{R_i^+ + s_i^+}{R_i^- + s_i^-}\right) \tag{6.6.8}$$

$$\int_{\partial_i S'} R\,\mathrm{d}l' = \frac{1}{2}\left(s_i^+ R_i^- s_i^- R_i^- + (R_i^0)^2 \int_{\partial_i S'} \frac{1}{R}\,\mathrm{d}l'\right) \tag{6.6.9}$$

$$\int_{\partial_i S'} R^3\,\mathrm{d}l' = \frac{1}{4}\left(s_i^+ (R_i^+)^3 - s_i^- (R_i^-)^3 + 3(R_i^0)^2 \int_{\partial_i S'} R\,\mathrm{d}l'\right). \tag{6.6.10}$$

The definitions of the distances $R_i^{\pm/0}$, $t_i^{\pm/0}$ and $s_i^{\pm/0}$ are given in Table 6.1. In the proceeding subsections, we we use these results to express the integrals for C^2, D^1 and D^2.

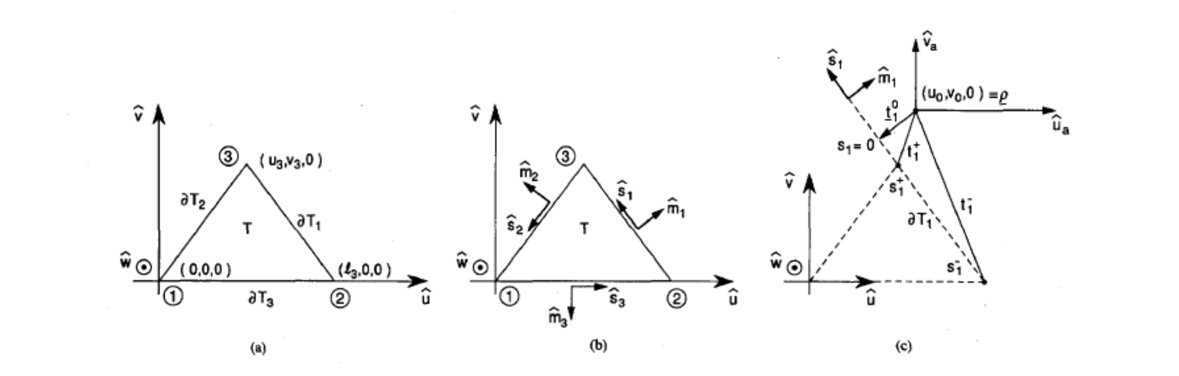


Figure 6.4: Variable definitions for analytic integration of boundary integrals. Image via Graglia [63].

6.6.3 The integrals for $D^{(1)}$

We begin by separating the integral $D^1_{m,}$ into a sufficiently smooth and a singular part,

$$\begin{aligned} D^{(1)}_{m,n} = \mathrm{i}k_0 \oint_S \mathbf{f}_m \cdot \hat{\mathbf{n}} \times \Bigg[& \oint_{S'} \left(\frac{\mathrm{e}^{-\mathrm{i}k_0 R} - 1}{4\pi R} + \frac{k_0^2 R}{8\pi} \right) \mathbf{f}_n(\mathbf{r}')\,\mathrm{d}S' \\ & + \frac{1}{4\pi} \oint_{S'} R^{-1} \mathbf{f}_n(\mathbf{r}')\,\mathrm{d}S' \\ & - \frac{k_0^2}{8\pi} \oint_{S'} R \mathbf{f}_n(\mathbf{r}')\,\mathrm{d}S' \Bigg] \,\mathrm{d}S \end{aligned} \tag{6.6.11}$$

The integrand of the first part has the limit

$$\lim_{R\to 0} \left\{ \frac{\mathrm{e}^{-\mathrm{i}k_0 R} - 1}{4\pi R} + \frac{k_0^2 R}{8\pi} \right\} = -\mathrm{i}\frac{k_0}{4\pi}, \tag{6.6.12}$$

and is sufficiently smooth for low-order numerical integration. By equation (6.6.5) and the definition of $\mathbf{f}_n$, with $\boldsymbol{\rho} = \mathbf{r} - w_0\hat{\mathbf{n}}$, the integral of the first singular part in (6.6.11) is given by

$$\begin{aligned} \oint_{S'} (\mathbf{r}' - n)\frac{1}{R}\,\mathrm{d}S' &= \oint_{S'} \frac{\mathbf{r}' - \mathbf{r}}{R}\,\mathrm{d}S' + (\mathbf{r} - n) \oint_{S'} \frac{1}{R}\,\mathrm{d}S' \\ &= \oint_{S'} \nabla' R\,\mathrm{d}S' + (\mathbf{r} - n) \oint_{S'} \frac{1}{R}\,\mathrm{d}S' \\ &= \sum_{i=1}^{3} \hat{\mathbf{m}}_i \oint_{\partial_i S'} R\,dl' + (\boldsymbol{\rho} - n) \oint_{S'} \frac{1}{R}\,\mathrm{d}S'. \end{aligned} \tag{6.6.13}$$

Similarly, the second part is given by

$$\begin{aligned} \int_{S'} (\mathbf{r}' - n) R\,\mathrm{d}S' =& \frac{1}{3} \int_{S'} \nabla'(R^3)\,\mathrm{d}S' + (\mathbf{r} - n) \int_{S'} R\,\mathrm{d}S' \\ =& \frac{1}{3} \sum_{i=1}^{3} \hat{\mathbf{m}}_i \int_{\partial_i S'} R^3\,dl' + (\boldsymbol{\rho} - n) \int_{S'} R\,ds'. \end{aligned} \tag{6.6.14}$$

by (6.6.7).

6.6.4 The integrals for $D^{(2)}$

Since we are using linear edge shape functions on the boundary, their divergence $\nabla' \cdot \mathbf{f}_n(\mathbf{r}') = \nabla' \cdot (\hat{\mathbf{n}}(\mathbf{r}') \times \mathbf{L}_n)$ is constant and so can be taken outside of the integral over S'. This gives us

$$D^{(2)}_{m,n} = \frac{\mathrm{i}}{k_0} \oint_S \mathbf{f}_m(\mathbf{r}) \cdot \hat{\mathbf{n}} \times \left(\nabla' \cdot \mathbf{f}_n(\mathbf{r}') \oint_{S'} \nabla' G_0(\mathbf{r}, \mathbf{r}') ds' \right) \, ds. \tag{6.6.15}$$

If we write the scalar Green's function as

$$G_0(\mathbf{r}, \mathbf{r}') = \left(G_0(\mathbf{r}, \mathbf{r}') - \frac{1}{4\pi R} + \frac{k_0^2}{8\pi} R \right) + \frac{1}{4\pi R} - \frac{k_0^2}{8\pi} R, \tag{6.6.16}$$

then on taking the gradient we have

$$\oint_{S'} \nabla' G(\mathbf{r}, \mathbf{r}') \, ds' = \oint_{S'} \nabla' q(R) \, ds' + \frac{1}{4\pi} \oint_{S'} \nabla' \frac{1}{R} \, ds' - \frac{k_0^2}{8\pi} \oint_{S'} \nabla' R \, ds', \tag{6.6.17}$$

where the term $\nabla' q$ is given by

$$\nabla' q(R) = \left(\frac{2\mathrm{e}^{-\mathrm{i}k_0 R}(\mathrm{i}k_0 R + 1) - k_0^2 R^2 - 2}{8\pi R^3} \right) \mathbf{R}, \tag{6.6.18}$$

and has the limit

$$\lim_{R \to 0} \nabla' q(R) = -\mathrm{i} \frac{k_0^3}{12\pi} \mathbf{R}. \tag{6.6.19}$$

The integrals of the two singular integrands are given by equations (6.6.5) and (6.6.6).

6.6.5 The integrals for $C^{(2)}$

Consider the integrals for matrix $C^{(2)}$,

$$C^{(2)}_{m,n} = \oint_S \mathbf{f}_m \cdot \left(\hat{\mathbf{n}} \times \oint_{S'} \nabla' G_0(\mathbf{r}, \mathbf{r}') \times \mathbf{f}_n(\mathbf{r}') ds' \right) \, ds, \tag{6.6.20}$$

which contains a $1/R^2$ singularity. However, if the two triangles S and S' are on the same plane the value of the integral vanishes. This is because both $\nabla' G$ and $\mathbf{f}_n = \hat{\mathbf{n}} \times \mathbf{L}_n$ are tangential to the plane of S and S', and so their cross product is normal to the plane. Then, as $\hat{\mathbf{n}}(\mathbf{r}) = \hat{\mathbf{n}}(\mathbf{r}')$, the cross product of $\hat{\mathbf{n}}(\mathbf{r})$ with the inner integral is zero.

Assuming S and S' lie on different planes, let us write

$$\begin{aligned} H^n(\mathbf{r}) &= \oint_{S'} \nabla' G_0(\mathbf{r}, \mathbf{r}') \times \mathbf{f}_n \, ds'. \\ &= \frac{1}{4\pi} \oint_{S'} \left(\mathrm{i}k_0 + \frac{1}{R} \right) \hat{\mathbf{R}} \times \mathbf{f}_n(\mathbf{r}') \frac{\mathrm{e}^{-\mathrm{i}k_0 R}}{R} \, ds' \end{aligned} \tag{6.6.21}$$

If $\boldsymbol{\rho}'_n = \boldsymbol{\rho}_n(\mathbf{r}') = (\mathbf{r}' - n)$, then

$$H^n(\mathbf{r}) = \frac{l_n}{8\pi A_{S'}} \oint \left(\mathrm{i}k_0 + \frac{1}{R}\right) \hat{\mathbf{R}} \times \boldsymbol{\rho}'_n \frac{\mathrm{e}^{-\mathrm{i}k_0 R}}{R}\, ds'. \tag{6.6.22}$$

Expressing $\mathbf{R}$ in terms of $\boldsymbol{\rho}'_n$ and the observation related vector $\mathbf{R}_n$,

$$\hat{\mathbf{R}} \times \boldsymbol{\rho}'_n = \frac{\mathbf{R}}{R} \times \boldsymbol{\rho}'_n = \frac{\mathbf{R}_n - \boldsymbol{\rho}'_n}{R} \times \boldsymbol{\rho}'_n = \frac{1}{R}(\mathbf{R}_n \times \boldsymbol{\rho}'_n), \tag{6.6.23}$$

we have

$$H^n(\mathbf{r}) = \frac{l_n}{8\pi A_{S'}} \mathbf{R}_n \times \oint \boldsymbol{\rho}'_n \frac{(\mathrm{i}k_0 R + 1)\mathrm{e}^{-\mathrm{i}k_0 R}}{R^3}\, ds', \tag{6.6.24}$$

since $\mathbf{R}_n$ is constant with respect to $\mathbf{r}'$. Following the same approach of extracting the singularity as in D^1 and D^2, we have

$$\begin{aligned} H^n(\mathbf{r}) =& \frac{l_n}{8\pi A_{S'}} \mathbf{R}_n \times \Bigg[\oint_{S'} \boldsymbol{\rho}'_n \frac{(\mathrm{i}k_0 R + 1)\mathrm{e}^{-\mathrm{i}k_0 R} - (1 + 0.5k_0^2 R^2)}{R^3}\, ds' \\ &+ \oint_{S'} \frac{\boldsymbol{\rho}'_n}{R^3}\, ds' + \frac{k_0^2}{2} \oint_{S'} \frac{\boldsymbol{\rho}'_n}{R}\, ds'\Bigg]. \end{aligned} \tag{6.6.25}$$

Here, we have the limit

$$\lim_{R\to 0} \frac{(\mathrm{i}k_0 R + 1)\mathrm{e}^{-\mathrm{i}k_i R} - (1 + 0.5k_0^2 R^2)}{R^3} = -\frac{1}{3}\mathrm{i}k_0^3, \tag{6.6.26}$$

as well as the analytic expressions for singular parts

$$\oint_{S'} \frac{\boldsymbol{\rho}'_n}{R^3}\, ds' = -\sum_{i=1}^{3} \hat{\mathbf{m}}_i \ln\left(\frac{R_i^+ + s_i^+}{R_i^- + s_i^-}\right) + (\boldsymbol{\rho} - n) \oint_{S'} \frac{1}{R^3}\, ds' \tag{6.6.27}$$

and

$$\oint_{S'} (\mathbf{r}' - n)\frac{1}{R}\, ds' = \sum_{i=1}^{3} \hat{\mathbf{m}} \oint_{\partial_i S'} R\, dl' + (\boldsymbol{\rho} - n) \oint_{S'} \frac{1}{R}\, ds', \tag{6.6.28}$$

by equations (6.6.5), (6.6.6) and (6.6.8).

6.7 Modelling sources and receivers

The excitation vectors for the MFIE and EFIE are given by

$$\phi_i^M = \oint_{S_1} \mathbf{f}_i \cdot \hat{\mathbf{n}} \times (\mathbf{H}^{\mathrm{in}} + \mathbf{H}^{\mathrm{r}})\, ds \tag{6.7.1}$$

and

$$\phi_i^E = \oint_{S_1} \mathbf{f}_i \cdot \hat{\mathbf{n}} \times (\mathbf{E}^{\mathrm{in}} + \mathbf{E}^{\mathrm{r}})\, ds \tag{6.7.2}$$

respectively, where S_1 is the ground surface . The simplest realistic source we can consider is that of a dipole antenna, which we can approximate as

$$\mathbf{E}^{\text{in}}(\mathbf{r}) = A(\theta)\overline{\overline{\mathrm{G}}}(\mathbf{r}_s, \mathbf{r})\hat{\mathbf{p}}^{\text{in}} \tag{6.7.3}$$

where $\mathbf{r}_s$ is the source location, $\hat{\mathbf{p}}^{\text{in}}$ the source polarisation, and $A(\theta)$ an amplitude term which varies with the angle between $\mathbf{r} - \mathbf{r}_s$ and direction the antenna is pointed $\hat{\mathbf{d}}^{\text{in}} \perp \hat{\mathbf{p}}^{\text{in}}$. Choice of a smoothly decaying amplitude allows us to model approximately a wide range of possible antenna patterns, which can be fitted to true antenna radiation patterns.

Given the surface values of $\mathbf{J}_s$ and $\mathbf{M}_s$ on S_1, the response at the receiving antenna is calculated by the surface equivalence theorem,

$$\begin{aligned}
\mathbf{H}^{\text{ext}} &= \oint_{S_1} \Big\{ -\mathbf{J}_s(\mathbf{r}') \times \nabla G_0(\mathbf{r}, \mathbf{r}') - \mathrm{i}k_0 Y_0 \mathbf{M}_s(\mathbf{r}') G_0 \\
&\qquad - \mathrm{i}\frac{Y_0}{k_0} \cdot \mathbf{M}(\mathbf{r}') \cdot \nabla\nabla G_0(\mathbf{r}, \mathbf{r}') \Big\}\, ds' \\
\mathbf{E}^{\text{ext}} &= \oint_{S_1} \Big\{ \mathbf{M}_s(\mathbf{r}') \times \nabla G_0(\mathbf{r}, \mathbf{r}') - \mathrm{i}k_0 Z_0 \mathbf{J}_s(\mathbf{r}') G_0 \\
&\qquad - \mathrm{i}\frac{Z_0}{k_0} \mathbf{J}(\mathbf{r}') \cdot \nabla\nabla G_0(\mathbf{r}, \mathbf{r}') \Big\}\, ds'.
\end{aligned} \tag{6.7.4}$$

As with the source term, if we have a directional receive antenna then we can multiply the Green's function with some (smooth) angle-dependent cut-off function $A(\theta)$, physically accounting for waves not being picked up (or unable to enter) a directional antenna. These integrals are then evaluated at the position(s) of the receiving antenna, and the scalar product taken with the polarisation direction of receiving antenna $\hat{\mathbf{p}}^{\text{rec}}$. Note that, unlike in the CFIE equations, the integrals in equations (6.7.4) are over surfaces away from the singularity of G_0, since we assume the receiving antenna is off of the surface.

6.8 Iterative solution of the FE-BI system

6.8.1 Preconditioning methods for the FE-BI system

The coupled finite-element boundary-integral method results in a large partially sparse, partially dense, square system matrix S, which is complex, non-symmetric and fairly ill-conditioned. The matrix equation is difficult to solve efficiently, especially as the scale of the problem increases. As discussed in Section 4.2.2, for large systems the most efficient approach is to use an iterative solver, while direct methods are only suitable for smaller systems. Indeed, at some point it becomes prohibitive to use direct LU based methods to solve the system, since a full LU decomposition of S will produce large amounts of fill-in and one only ever has limited availability of memory. The rate of convergence of iterative solvers though will likely depend drastically on the choice of preconditioning used: with a poor choice (or no preconditioning), Krylov subspace based methods may fail to converge to the required tolerance or simply stagnate, similar to the discretised Helmholtz problem.

Recall that the point of preconditioning is to replace the system

$$S\mathbf{x} = \mathbf{b} \tag{6.8.1}$$

by one hopefully easier to solve,

$$M^{-1}S\mathbf{x} = M^{-1}\mathbf{b}. \tag{6.8.2}$$

The preconditioner matrix M is required to be easy to calculate and invert (e.g. using a direct method), but $M^{-1}S$ should either be close to the identity or simply be better suited to our iterative method (e.g. through having a preferable eigenvalue spectrum). If $M^{-1}S = I$ then we have already solved the problem, but have made no performance gain since we have solved (6.8.1) directly.

In Section 4.2.2 we mentioned some general algebraic preconditioners applicable to many linear algebra problems, such as the Jacobi preconditioner and incomplete factorisations, but it is possible too to derive methods based on physical approximations of the forward problem. Liu and Jin [80, 100] give a method in which the BI portion of S is approximated by the absorbing boundary condition (3.3.4). Writing FE-BI system as

$$S\mathbf{x} = (A + B)\mathbf{x} = \mathbf{b}, \tag{6.8.3}$$

where A is the sparse symmetric FE matrix, B is the dense, non-symmetric BI matrix, the preconditioner suggested by Liu and Jin is to use

$$\begin{aligned} \hat{\mathbf{n}} \times (\nabla \times \mathbf{E}) + \mathrm{i}k_0\hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{E}) =& \hat{\mathbf{n}} \times (\nabla \times \mathbf{E}^{\mathrm{in}}) \\ & + \mathrm{i}k_0\hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{E}^{\mathrm{in}}) \qquad \mathbf{r} \in \partial\Omega, \end{aligned} \tag{6.8.4}$$

or

$$\begin{aligned}\hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{E}) - \hat{\mathbf{n}} \times \mathbf{H} =& \hat{\mathbf{n}} \times (\hat{\mathbf{n}} \times \mathbf{E}^{\text{in}}) \\ & - \hat{\mathbf{n}} \times \mathbf{H}^{\text{in}} \qquad \mathbf{r} \in \partial\Omega,\end{aligned} \tag{6.8.5}$$

to replace the CFIE equations. Unlike the BI CFIE equations, the discrete approximation of the ABC results in a sparse matrix, denoted L. L is a local approximation to B, and the large eigenvalues of $(A + L)$ are similar to those of $(A + B)$ [100] which makes it suitable for GMRES.

Multi-grid methods [202], [64, pp 183-204], could also be used to precondition the FE-BI system, such as the Multilevel Fast Multi-pole Algorithm (MLFMA) [153, 161]. In MLFMA, the domain is successively subdivided into smaller cubes. Electromagnetic interactions within the same or neighbouring cubes are calculated directly, but those apart by the addition theorem (i.e. multiple scattering at this stage are ignored). The process is then carried out iteratively on the higher level grids, until the coarsest subdivision is reached. MLFMA reduces the computational complexity of the BI portion of the matrix from $\mathcal{O}(N_s^2)$ to $\mathcal{O}(N_s \log N_s)$, becoming a very practical method for large problems

6.8.2 An ILUC preconditioner

Through some preliminary testing, we have found that for our scale of problem iterative methods will often stagnate with a simple Jacobi preconditioner, and while using a sparse approximation to S (such as the ABC preconditioner described in (6.8.5)) is robust, the time taken to solve the linear system with such a preconditioner is generally greater than when using MATLAB's inbuilt direct method `mldivide` (backslash). This is likely due to the dense BI matrices taking up only a very small proportion S, so the cost of inverting M is almost on par with inverting S. Similarly, we find that constructing an ILU factorisation of S either results in a non-robust scheme if the drop tolerance is too high (i.e. LU is a poor approximation of S), or the time taken to construct the decomposition is on par with using a direct method.

Table 6.2 shows some function timings for the built in MATLAB functions `mldivide` and `ILU` of a system matrix for a 50 cm^3 domain with 110123 edges, $\epsilon_r = 4$ background and a 5 cm radius sphere target with $\epsilon_r = 6$, for which we used the efficient Crout

Operation	Computation time (s)			
	Pass 1	Pass 2	Pass 3	Average
`mldivide`$(S, \mathbf{b})$	207.94	206.76	206.94	206.88
`mldivide`(S, B)	363.36	365.18	363.54	364.03
`ILU`$(S, 10^{-2})$	190.05	190.14	189.81	190.00
`ILU`$(S, 10^{-3})$	292.36	290.80	288.96	290.71
`ILU`$(S, 10^{-4})$	1217.63	1233.08	1216.85	1222.52

Table 6.2: Function timings in MATLAB for `mldivide` with a single and 90 right-hand-side vectors, and the Crout variant of `ILU`(S, tol) for given drop tolerances, for a 110123 square FE-BI system matrix. Timings performed in single thread mode on an Intel Xeon E3-1280 v3 at 3.60 GHz based computer in MATLAB R2014a under Scientific Linux 6.4. S is a 110123^2 matrix, of which the two dense blocks are 990^2.

variant of ILU [99]. We see that `mldivide` scales well when given multiple right-hand-side vectors, as it only performs the LU decomposition once. The `ILU` with a drop tolerance of 10^{-2} would be an obvious choice of preconditioner here,as if an iterative method such as GMRES would converge in 2s per right-hand-side vector it would be faster than `mldivide` for this problem. Unfortunately, the high drop tolerance makes it a poor preconditioner and GMRES fails to converge. If we use an ILU preconditioner with a drop tolerance of 10^{-3}, we would need GMRES to converge in under 0.8s per right-hand-side in order to be more efficient than `mldivide` when including the time to construct the preconditioner. While this seems to be a stable preconditioner, `mldivide` still outperforms the built in MATLAB implementation `gmres`, when it is possible to use the direct method within memory constraints.

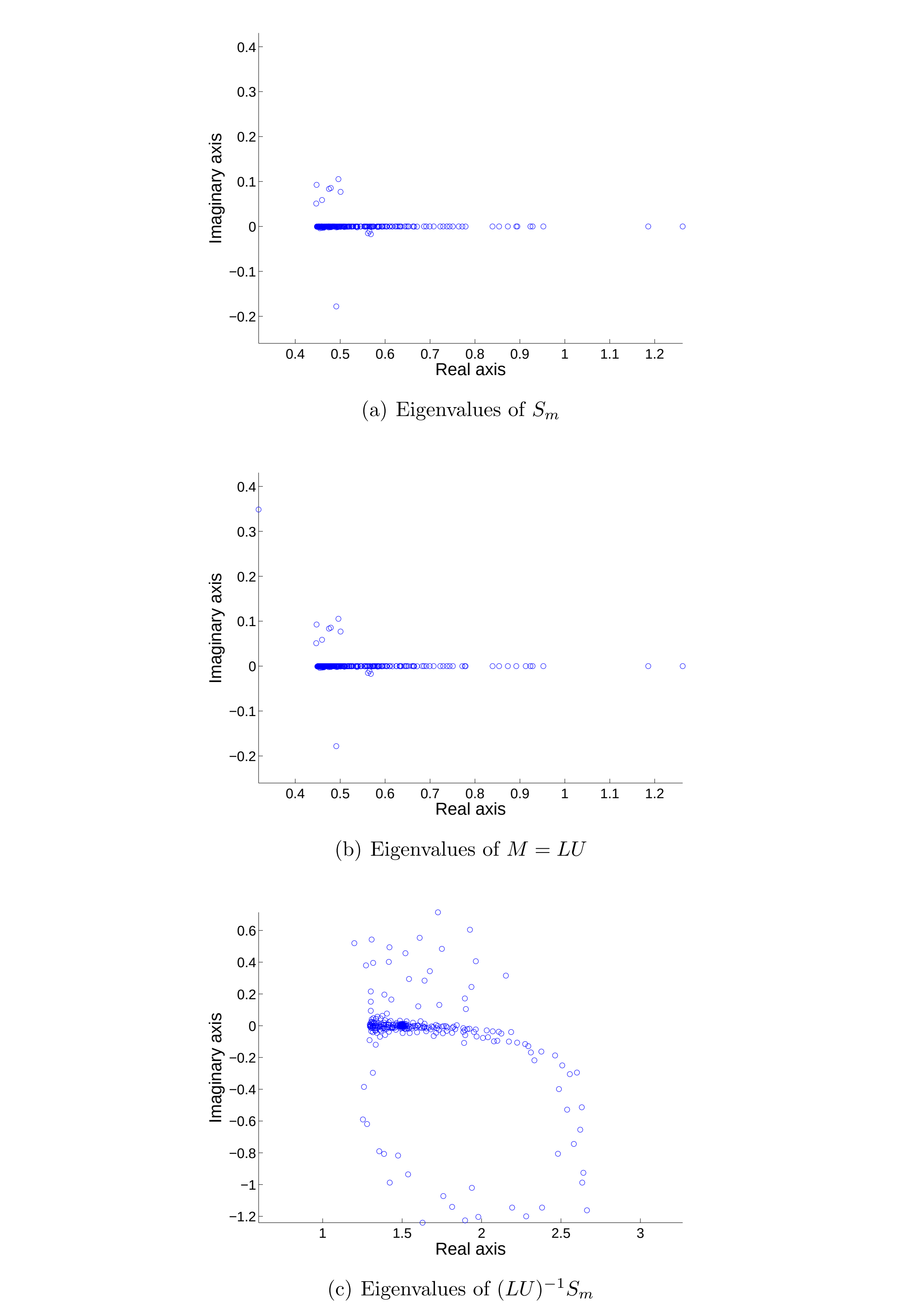
We assume that our scale of problem is on the verge of where direct methods cease to become effective (or possible) and iterative methods are necessary, but the inherent efficiencies built in to `mldivide` often allow it to outperform iterative schemes when we include the cost of constructing M since `mldivide` scales well when given multiple right-hand-sides simultaneously. We have not attempted to implement a multi-grid method, on the assumption that it would only be of benefit for (at least slightly) larger scaled problems.

However, while the time taken to create an ILU decomposition of S is on par with solving directly the linear system, we have found that once we have an ILU of $S(\mathbf{m}_0)$ this can continue to be an effective preconditioner for $S(\mathbf{m}_k)$, where $\mathbf{m}_0$ is some (possibly homogeneous) initial model and $\mathbf{m}_k$ some perturbed model. We assume this is due to the large similarities between the two: the boundary integral portion of S always stays the same, and the parts corresponding to the PML should not be able to change much during any inversion. ILU($S(\mathbf{m}_0)$) as a preconditioner mimics the properties of the outgoing wave only.

The GMRES algorithm often performs well when the complex eigenvalues of $M^{-1}S$ are tightly clustered about a single point away from the origin [64, pp55], which from Figure 6.5 we see is the case for $(LU)^{-1}S$. In Figure 6.6, we see we gain exponential convergence using the preconditioned GMRES for the same model problem used above, for ILU preconditioners calculated on homogeneous domains of permittivity $\epsilon_r = 3.8, 4, 4.2$ and for various drop tolerances. Reduction in residual is greater per iteration for a lower drop tolerance, although A drop tolerance of 10^{-2} gave too poor an approximation to the LU decomposition and GMRES failed. With a lower drop tolerance, each iteration of GMRES becomes more expensive to compute due to the increased density of L and U. Computation times of MATLAB's inbuilt gmres function are given in Table 6.3, and we see there is a 'sweet spot' in computation time with a drop tolerance $\mathcal{O}(10^{-3})$.

From the convergence plots and execution times that the ILU preconditioner calculated on a homogeneous domain can remain effective even where the background permittivity is incorrect (at least for this simple test problem). In this case, if one had fewer than around 30 source terms (which is perhaps not representative of a GPR data set), then preconditioned GMRES out-performs mldivide. For larger systems we expect this number of source terms to grow, and of course iterative methods become a necessity due to memory limitations. To test the implementation of the FE-BI system, several simple scattering problems containing dielectric spheres were set up, though the results are omitted here.

One of the reasons iterative methods can perform slowly for the FE-BI system is that matrix-multiplication algorithms for sparse systems are often not suited to the partially dense, partially sparse structure if it is stored all together in a sparse format. The time taken to perform the dense part of the multiplication $S\mathbf{x}$ will be disproportionately large, since it will not have the structure assumed by the sparse format used and lead to repeated access of the same elements of $\mathbf{x}$, with elements of the dense part of S likely stored in memory in a less than optimal order. Conversely, a dense matrix



(a) Eigenvalues of S_m

(b) Eigenvalues of $M = LU$

(c) Eigenvalues of $(LU)^{-1}S_m$

Figure 6.5: Largest 250 eigenvalues of (a) S_m, (b) $M = LU$, (c) $(LU)^{-1}S_m$. S_m is the system matrix for the model problem described in section 6.8.2, and LU is an ILUC decomposition with drop tolerance 10^{-3} of a system matrix S_0 for homogeneous permittivity $\epsilon_r = 4$.

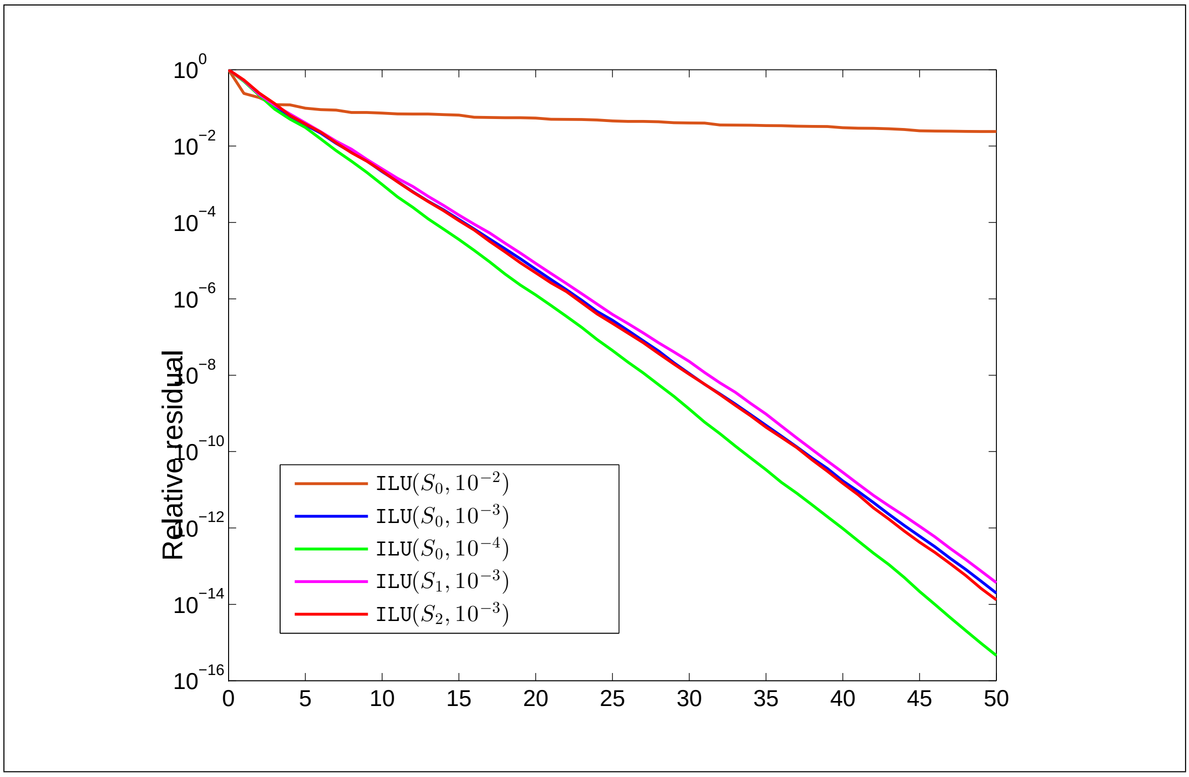


Figure 6.6: Relative residual of GMRES solution against iteration for various ILU based preconditioners and drop tolerances. S_0, S_1 and S_2 are the system matrix for homogeneous permittivities $\epsilon_r = 4, 3.8$ and 4.2 respectively

Preconditioner	System matrix	`gmres`$(S, \mathbf{b}, 10^{-6})$ computation time (s) Pass 1	 Pass 2	 Pass 3	 Average	Iterations
`ILU`$(S_0, 10^{-2})$	S_m	–	–	–	–	200
`ILU`$(S_0, 10^{-3})$	S_0	5.91	5.95	5.83	5.90	19
	S_m	7.03	7.04	7.15	7.07	23
`ILU`$(S_0, 10^{-4})$	S_m	17.14	16.90	16.99	17.01	21
`ILU`$(S_1, 10^{-3})$	S_m	7.38	7.35	7.40	7.38	23
`ILU`$(S_2, 10^{-3})$	S_m	7.46	7.30	7.59	7.45	23

Table 6.3: Computation times for preconditioned `gmres` to reach a relative residual of 10^{-6}. `ILU`(S, tol) is the Crout variant of the `ILU` preconditioner with a drop tolerance tol. S_m is the system matrix for the sample scattering problem with a single spherical target of $\epsilon_r = 6$ in a background of $\epsilon_r = 4$, S_0, S_1 and S_2 are created for homogeneous $\epsilon_r = 4, 3.8$ and 4.2, respectively. `gmres` failed to converge in 200 iterations with the `ILU`$(S_0, 10^{-2})$ preconditioner. `gmres` was run without restarts. Timings performed in single thread mode on a Intel Xeon E3-1280 v3 at 3.60 GHz based computer in MATLAB R2014a under Scientific Linux 6.4. The S are 110123^2 matrices, of which the two dense blocks are 990^2.

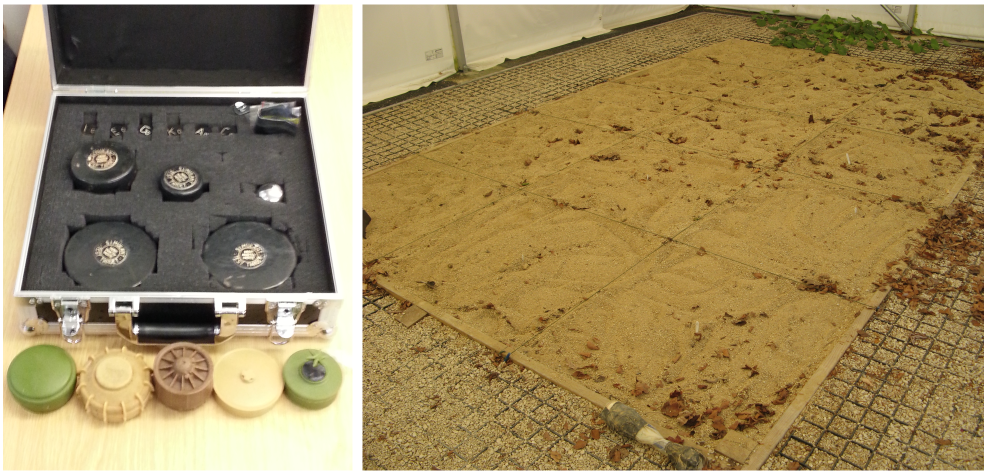
multiplication algorithm will perform well in the dense portion, accessing elements of $\mathbf{x}$ in memory only once, but will also compute all the zero multiplications in the sparse portion (moreover, S will be too large to store in dense format). The 'Fast Multi-pole' portion of MLFMA [153, 161] described above overcomes this difficulty. More simply though, one can gain decent computation speed gains and modest reductions in memory footprint by storing the sparse and dense portions of the FE-BI system separately in the appropriate format, and performing the multiplications of these parts against $\mathbf{x}$ individually before summing the contributions.

6.9 Comparison to field data

GPR data was collected at a test bed at Cranfield University, Shrivenham, supported by Utsi Electronics and EPSRC grant IAA074 [196]. Figure 6.7(b) shows the test bed, which contains a very sandy soil (imported from Afghanistan), and so is not representative of many of the places in which landmines are found or are a humanitarian concern, but is a useful starting point to verify simulations. A range of targets were buried in the site, including various landmine surrogates kindly lent by the School of Electrical and Electronic Engineering, University of Manchester, shown in Figure 6.7(a). There were also a range of IED and UXO surrogates and various clutter objects already in place. Utsi Electronics kindly loaned two GPR systems to collect data, one on-ground and one off-ground (technical details are omitted here to protect Utsi Electonics' IP). Figure 6.8 shows a B-scan through the test site using the on-ground system which had a 1 GHz centre frequency.

To begin verifying our FEBI solver, we simulate data for an AISI 420 stainless steel sphere with 44.45mm diameter, buried 12cm deep (to the bottom of the sphere) in a homogeneous sand background, and compare to the data recorded for this target. We assume the sand has permittivity $\epsilon_r = 4$, and the steel sphere $\epsilon_1 = 1$, $\mu_r = 1.005$ and $\sigma = 1.25 \times 10^6$ S/m. These values may not be accurate, as we could not find specific values of AISI 420 stainless steel at microwave frequencies, nor do we have an accurate permittivity from the Afghanistan sand. The source wavelet was estimated by an average fit of the simulated wave amplitude to the recorded at each frequency,

$$A(\omega_i) = (\mathbf{d}_i^{\text{sim}})^{\dagger}(\mathbf{d}_i^{\text{obs}})^T, \tag{6.9.1}$$



(a) Landmine surrogates (b) Test bed at Cranfield University, Shrivenham

Figure 6.7: (a) Landmine surrogates and (b) test bed used for data collection. The larger surrogates in the case are designed to give an accurate GPR response of a landmine, while the small targets in the top row (which can be inserted into the larger bodies) are metal detection surrogates.

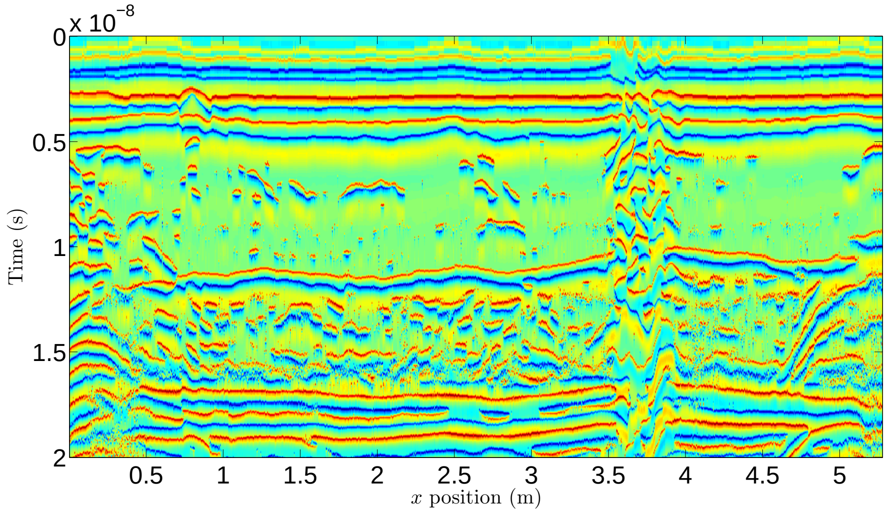


Figure 6.8: Time domain B-scan or surrogate landmine targets. The target at 0.8 m is a stainless steel ball, and the target at 3.75 m is a bundle of wires on the ground surface. Landmine surrogate targets were out of the survey line shown, but a slight response to one can be seen at 2.5 m.

where $\mathbf{d}_i^{\text{sim}}$ and $\mathbf{d}_i^{\text{obs}}$ are the simulated and recorded data for the i^{th} frequency ω_i, respectively, and † the Moore Penrose pseudo-inverse. If we only simulated data in a single location this would force our simulation to fit exactly the recorded data, so we simulate in a sufficiently large area such that we have parts of the data that are and parts that are not affected by the target. It is easier to understand the data in the time domain, so we IFFT the simulated data to compare A/B-scans. We compare to the recorded data at just the frequencies simulated, so that

$$\tilde{\mathbf{d}}^{\text{obs}} = \text{IFFT}(FFT(\mathbf{d}^{\text{obs}}, 512), n_\omega), \tag{6.9.2}$$

where n_ω is the number of simulated frequencies. This way, the comparison is not obscured by frequency features in the data we have not simulated. Figures 6.9 and 6.10 compare simulated and recorded A and B scans, respectively.

Qualitatively, from Figure 6.9 we see we have the correct number of peaks and troughs, though the amplitudes do not match. The ground reflection (up to 2 ns) has a greater amplitude, most likely due to an incorrect estimation of ground permittivity. We also do not simulate the fairly rough ground surface. The first reflection from the steel sphere (3–5 ns) also has a greater amplitude, but the multiple after is less pronounced. Again, this is likely to be due to an incorrect estimation of the contrast to the background medium. Finally, we see no obvious affect from the PML region, which would begin at approximately 10 ns in the time domain signal.

Figure 6.10 displays similar simulated hyperbolas, both in the primary reflection and multiples after, to the field data. It is clear that much more complex interactions are occurring in the real data, due to layering of the sand from regular digging as well as the medium being quite granular with an uneven surface.

This qualitative comparison is promising, but more would need to be done to accurately verify the data simulation. Particularly, we should compare a multi-static response to a well characterised object in a well known background medium, in a controlled environment such that the background is (approximately) homogeneous. Antenna position would also need to be more accurately controlled or measured.

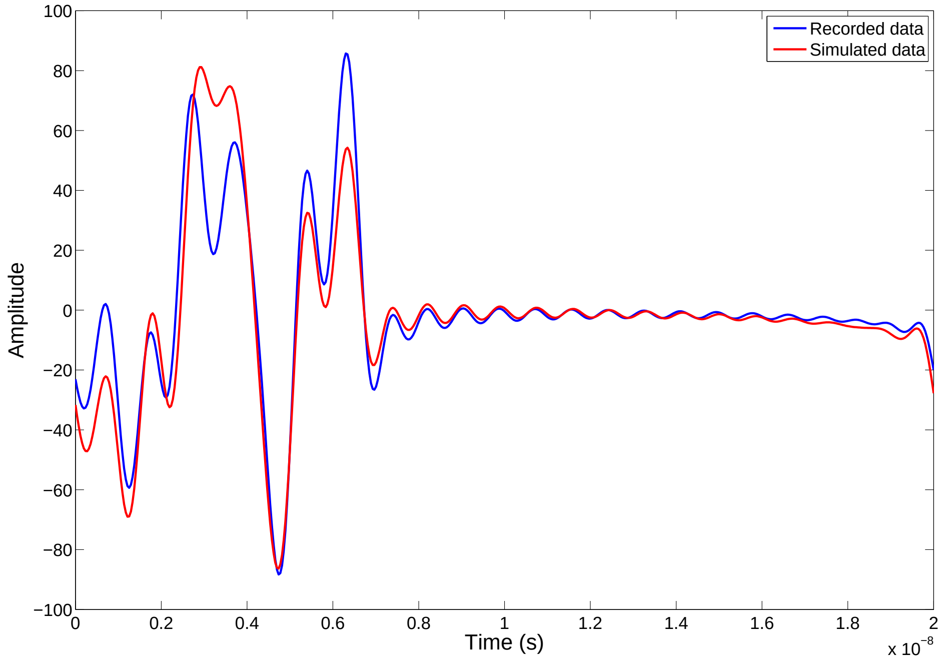


Figure 6.9: Comparison of simulated and recorded A-scan directly above a 44.45mm diameter stainless steel ball in a sand background.

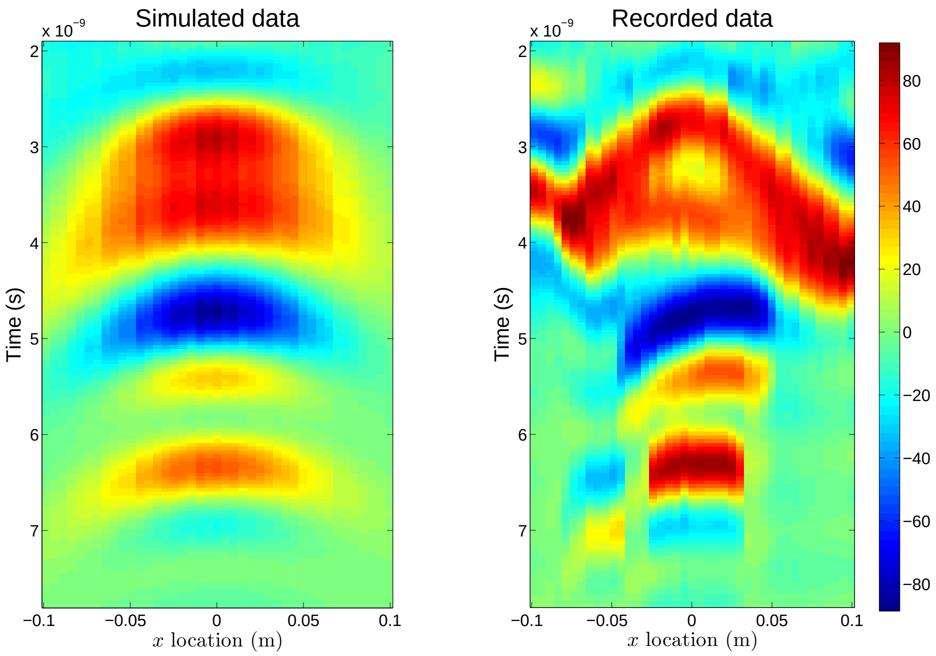


Figure 6.10: Comparison of simulated and recorded B-scan of a 44.45mm diameter stainless steel ball in a sand background.

6.10 Summary

This chapter has described the numerical simulation of GPR data in 3D using a finite-element boundary-integral method (FE-BI). We have first given the finite-element (FE) formulation of Maxwell's equations, allowing numerical solution of scattering problems in a given domain. The FE system was verified against the analytic solution of a plane wave.

Using FEM to simulate GPR data requires boundary conditions on the ground surface representing the incident wave. These transmission conditions are non-local due to reflection of the waves off of the ground as well as scattering in the ground surface, and are solved via a boundary integral (BI) method. This is then coupled with the FEM system, and the resulting system matrix is partially dense and partially sparse. The infinite subsurface also needs to be truncated for numerical computation, which was achieved via a perfectly matched layer.

We then discussed solution methods for the resulting FE-BI linear system. We showed that an ILU preconditioned iterative scheme can be more effective than a direct method for the average size of system we have, though up to this size MATLAB's built in `mldivide` performs comparably.

Finally, we compared the numerical solution of the GPR response of a stainless steel sphere in a sandy background to field data. The simulated data was promising, containing all the qualitative features we expected to see. However, more needs to be done to fully verify the solver is able to accurately

Chapter 7

Polarization Tensors

7.1 Introduction

The perturbation, or scattering, of electric and magnetic field due to the inclusion of an object of small size, either at high frequencies or in the far field, has received much attention. See for example the review of work prior to 1965 by Kleinman [84], or a more recent monograph by Dassios and Kleinman [46], and references therein. Particularly, this perturbation in field can be described in the form of an asymptotic expansion in the object's size, in which each term involves the product of a polarization tensor (sometimes referred to as a *polarizability tensor*) and the incident field at the object's location [83]. Particularly, for the scattering by a perfectly conducting object, the polarization tensor is related to that of Pólya, Schiffer and Szegö [135, 148]. Expansions involving a polarization tensor have been carried not out for various regimes of electromagnetics, including low-frequency (or objects small compared with the wavelength) [7, 14, 95, 185], the Electric Impedance Tomography problem [8, 10, 13, 34, 59], and the Eddy-current/metal detection problem [94, 107, 203], but also for non electromagnetic problems such as acoustics and linear elasticity [9, 12]. In the case of EIT the full asymptotic series is known, in which higher order terms involve higher order generalised polarisation tensors [9, Chapter 5]. These examples all consider characterising a finite number of isolated objects in a homogeneous background, which differs from GPR where the ground is inhomogeneous, with many small-scale perturbations from a granular material or layers. The polarization tensors depend on the shape of the object, as well as the contrast of its material parameters to that of the surrounding (homogeneous) medium, and so are invariant to the objects location (though they rotate with the object). They therefore give a useful description of the effect an object placed in an arbitrary electro-magnetic field.

Before proceeding to the 3D inverse problem, we consider in this chapter such scattering approximations using polarization tensors. We first define the tensor, before presenting a scattering approximation for the full Maxwell equations using the polarization tensor available in the literature. We show that this approximation can be appropriate for GPR with minor modifications if one had a homogeneous background and sufficiently isolated scatterers, as well as showing that the proof of the literature result still holds. While such approximations can result in efficient reconstruction algorithms in cases where one is looking for sufficiently isolated objects [7], this is unlikely to be suitable for GPR where both multiple scattering and an inhomogeneous background give large contributions to the data.

Rather than use the approximations to form an imaging method, we use them to inform a better understand sensitivity of our measurements to given perturbations or targets in the subsurface. We compare this alternative and novel view of sensitivity to the more traditional understanding involving the Fréchet derivative, contrasting what information each understanding can give and where each is applicable. This gives us a novel approach to sensitivity in FWI, extending analysis usually carried out with the Born approximation to include polarization effects. Finally, we use this novel understanding of sensitivity to analyse the type of linear array considered for the 2D inverse problem in Chapter 5.

7.2 Polarization tensors

7.2.1 The scattering problem

Let $\mathbf{E}$ and $\mathbf{H}$ be the electric and magnetic fields, respectively, satisfying Maxwell's equations (3.2.6) in a domain $\Omega \subseteq \mathbb{R}^3$, in which the permittivity $\epsilon(\mathbf{r})$ and permeability

$\mu(\mathbf{r})$ can be described as

$$\epsilon(\mathbf{r}) = \begin{cases} \epsilon^j & \mathbf{r} \in D_j, \quad j = 1, \ldots, m, \\ \epsilon^0 & \mathbf{r} \in \Omega \setminus \overline{\mathcal{I}}, \end{cases} \tag{7.2.1a}$$

$$\mu(\mathbf{r}) = \begin{cases} \mu^j & \mathbf{r} \in D_j, \quad j = 1, \ldots, m, \\ \mu^0 & \mathbf{r} \in \Omega \setminus \overline{\mathcal{I}}, \end{cases} \tag{7.2.1b}$$

$$\tag{7.2.1c}$$

where the $D_j \subset \mathbb{R}^3$ are sufficiently separated C^∞ smooth closed bounded domains (inclusions), with boundary ∂D_j and equipped with outward unit normal ν_j, and $\mathcal{I} = \bigcup_{j=1}^m D_j$.

The total fields can be described as the sum of incident fields and scattered fields, $\mathbf{E} = \mathbf{E}^{\text{in}} + \mathbf{E}^{\text{sc}}$, $\mathbf{H} = \mathbf{H}^{\text{in}} + \mathbf{H}^{\text{sc}}$. Here, the incident fields $\mathbf{E}^{\text{in}}$ and $\mathbf{H}^{\text{in}}$ are the solution to Maxwell's equations in Ω with homogeneous permittivity $\epsilon(\mathbf{r}) = \epsilon^0$ and permeability $\mu(\mathbf{r}) = \mu^0$ (in which we have used the superscript notation to emphasise that they need not be the free space permittivity and permeability), and $\mathbf{E}^{\text{sc}}$ and $\mathbf{H}^{\text{sc}}$ are the perturbations in fields due to the inclusions D_j. The incident and scattered fields both satisfy the transmission conditions (3.3.2),

We will consider inclusions of the form $D_j = \mathbf{z} + \alpha B_j$, where B_j are C^∞ smooth closed bounded domains of unit size and containing the origin, with boundary ∂B_j, so that the D_j are a translation and scaling (by a small α) of inclusions B_j. We consider also only cases in which the points $\mathbf{z}_j$ are sufficiently far apart (and the α sufficiently small) such that the principle of linear superposition holds – i.e. the total scattered field is the sum of scattered fields caused by each single inclusion alone. It is therefore sufficient for us to restrict our attention to a single inclusion $D = \mathbf{z} + \alpha B$ with permittivity and permeability ϵ^* and μ^*. The cases for multiple inclusions follow simply by summing the scattered fields.

7.2.2 Definition of a generalised polarisation tensor

Following Ammari et al [14], the polarization tensor $\mathscr{M}$ for an inhomogeneity B containing the origin, with parameter contrast $c = \gamma^0/\gamma^*$, is defined as the rank 2 3×3 tensor

$$\mathscr{M}_{ij}(c) = \frac{1}{c} \int_B \frac{\partial}{\partial x_i} \phi_j \, \mathrm{d}V, \tag{7.2.2}$$

where φ_j is the solution to the transmission problem

$$\begin{cases} \nabla^2\phi_j = 0 & \mathbf{x} \in B \text{ and } \mathbb{R}^3 \setminus \overline{B} \\ \phi_j & \text{is continuous across } \partial B \\ \frac{\gamma^0}{\gamma^*}\left(\frac{\partial\phi_j}{\partial\nu}\right)^+ - \left(\frac{\partial\phi_j}{\partial\nu}\right)^- = 0 & \text{on } \partial B \\ \phi_i(\mathbf{x}) - x_j \to 0 & \text{as } |\mathbf{x}| \to \infty. \end{cases} \tag{7.2.3}$$

It is clear that $\mathscr{M}$ is a function of the shape B of the inclusions D (in equation (7.2.1)), but not of their position $\mathbf{z}$. It can be shown that the tensor is symmetric, positive definite for $c > 1$ and negative definite for $0 < c < 1$, the elements are monotonic with c, and that (7.2.3) defines components of a tensor in Cartesian coordinates [11]. The expression (7.2.3) can alternatively be written as

$$\begin{cases} \nabla \cdot \gamma(\mathbf{x})\nabla\phi_j = 0 & \mathbf{x} \in \mathbb{R}^3 \\ \phi_j(\mathbf{x}) - x_j \to 0 & \text{as } |\mathbf{x}| \to \infty \end{cases} \tag{7.2.4}$$

where

$$\gamma(\mathbf{r}) = \begin{cases} \gamma^* & \mathbf{x} \in B \\ \gamma^0 & \mathbf{x} \in \mathbb{R}^3 \setminus \overline{B}. \end{cases} \tag{7.2.5}$$

We see that the ϕ_j can be considered as scalar potential to Maxwell's equations, $\mathbf{E} = \nabla\phi_j$ for an electrostatic field, with zero electric charge density and (non-physical) radiation conditions. Other definitions of the polarization tensor exist, which are equivalent (up to a constant) and found by an alternative choice of scalar potential [93]. It is also the same rank 2 tensor as used in asymptotic expansions for electrical impedance tomography [9] (the Pólya Szegö tensor of [135, 148]), where the higher order terms of the expansion are also known (and involve higher ranked tensors) [8].

7.2.3 Analytic values of tensor components

For the polarization tensor defined by (7.2.3) and (7.2.4), it can be shown that for B a ellipsoid oriented in the coordinate axes, $\mathscr{M}$ is the diagonal tensor [83, pp 45], [7],

$$\mathscr{M}_{ij} = \delta_{ij}|B|\frac{\gamma^0}{\gamma^0 + (\gamma^* - \gamma^0)L_i}, \tag{7.2.6}$$

where L_i is the depolarization factor of the ellipsoid given by [176, pp. 71]

$$L_i = \frac{1}{2}a_1a_2a_3\int_0^\infty \frac{\mathrm{d}s}{(s^2 - a_i^2)(s^2 - a_1^2)^{\frac{1}{2}}(s^2 - a_2^2)^{\frac{1}{2}}(s^2 - a_3^2)^{\frac{1}{2}}}, \tag{7.2.7}$$

for a_i the principle axes of the ellipsoid, and $L_1 + L_2 + L_3 = 1$. In the case of a sphere these are simply $L_i = \frac{1}{3}$. For prolate spheroids, $a_1 > a_2 = a_3$, we have

$$L_1 = \frac{1-e^2}{e^2}\left(-1 + \frac{1}{2e}\ln\frac{1+e}{1-e}\right), \qquad e^2 = 1 - \left(\frac{a_2}{a_1}\right)^2, \tag{7.2.8}$$

and for oblate spheroids, $a_1 > a_2 = a_3$,

$$L_1 = \frac{1+f^2}{f^2}\left(1 - \frac{1}{f}\arctan f\right), \qquad f^2 = \left(\frac{a_2}{a_1}\right)^2 - 1. \tag{7.2.9}$$

For either a prolate or oblate spheroid, for a_1 close to $a_2 = a_3$ then

$$L_1 \approx \frac{1}{3} + \frac{4}{15}\frac{a_2 - a_1}{a}. \tag{7.2.10}$$

Finally, for a flat elliptic disc, $a_1 \ll a_2, a_3$,

$$L_1 = 1, \quad L_2 = L_3 = 0, \tag{7.2.11}$$

and for a long elliptical cylinder, $a_2, a_3 \ll a_1$,

$$L_1 = 0, \quad L_2 = \frac{a_3}{a_2 + a_3}, \quad L_3 = \frac{a_2}{a_2 + a_3}. \tag{7.2.12}$$

If B is not oriented with the coordinate axes, the result is that the polarization tensor is pre-and post-multiplied by the rotation matrix R which maps the coordinate axes to the principle axes of B,

$$\mathscr{M} = R^T \mathscr{M}' R. \tag{7.2.13}$$

Note that (7.2.13) is an eigendecomposition of $\mathscr{M}$, since for rotation matrices $R^T = R^{-1}$.

We remark (but present no formal proof) that for any given (non-ellipsoidal) object B with tensor $\mathscr{M}$, there is an ellipsoid B' which would give the same tensor. To see this, it is sufficient to take an eigendecomposition of $\mathscr{M} = Q^{-1}\mathscr{M}'Q$ to find the diagonal tensor $\mathscr{M}'$, and show that there exists an ellipsoid oriented with coordinate axis with tensor $\mathscr{M}'$ (which can then be rotated by $R = Q$). Since L_i decreases monotonically with any a_i but increases monotonically with $a_j, j \neq i$, and $L_1 + L_2 + L_3 = 1$ with $0 \leq L_i \leq 1$, we can find any set of depolarization factors $0 \leq L_1 \leq L_2 \leq L_3 \leq 1$ which sum to 1. By (7.2.6), $\mathscr{M}'_{ii}$ decreases monotonically with L_i and γ^*, and increases with γ^0 and $|B|$. The depolarization factors, contrast and size of perturbation allow any diagonal $\mathscr{M}'$ (or equivalently any 3 eigenvalues) to be found.

In light of this remark, the most the polarization tensor $\mathscr{M}$ can say about the scattering an object B causes, is how its *closest fitting ellipsoid* would scatter the same wave field. The term 'closest' is not defined geometrically but through (7.2.6) and (7.2.7). In some applications, for example electric impedance tomography [9, chapter 5], a whole asymptotic series is known with each term involving a higher order tensor. In this case, higher order tensors give information about how the object differs from an ellipsoid (e.g. anti-symmetries), as the whole series describes exactly the scattering due to the object. As far as we are aware, the exact information conveyed by each order tensor about a scatterer is an open problem.

Expressions for the effect of a small ellipsoidal particle on an electric (or magnetic) field involving the depolarization factors were derived as early as Chasles [35] (1840), Thomson and Tait [170, pp. 525] (1879), Maxwell [109, pp. 60–70] (1873) and Lord Rayleigh [140] (1897). It is interesting that, while these are not formal asymptotic expressions nor in tensor form, some more recent results reduce to similar expressions as those in classical electrodynamics in the case of small ellipsoids.

7.3 An asymptotic expansion for GPR

7.3.1 Introduction

In this section, we present an asymptotic result from the literature which is given in terms of the tangential traces of the electromagnetic field on the surface of a closed and bounded domain. By application of a surface equivalence principle, we show the boundary integral of tangential traces in the result is equivalently the scattered field, and so this particular result can be applicable to GPR. Finally, we show that proof of the result still holds given the modifications required to apply the surface equivalence principle.

7.3.2 Tangential traces of fields on the surface of a closed domain

Ammari et al [14], also see Vogelius and Volkov [185], give the leading order terms in an asymptotic expansion as $\alpha \to 0$ for a source (charge) free closed and bounded domain $\Omega \subset \mathbb{R}$, with Neumann boundary conditions $(\nabla \times \mathbf{H}) \times \nu = g$ on $\partial\Omega$. They obtain that the tangential trace of the magnetic field on the boundary $\partial\Omega$ can be written as

$$\begin{aligned}\mathbf{H}^{\mathrm{sc}}(\mathbf{r}') \times \nu(\mathbf{r}') - 2\oint_{\partial\Omega} \nabla' \times (G_0(\mathbf{r},\mathbf{r}')(\mathbf{H}^{\mathrm{sc}}(\mathbf{r}) \times \nu(\mathbf{r})) \times \nu(\mathbf{r}')\,\mathrm{d}S \\ =2\alpha^3\omega^2\frac{\mu^0}{\mu^*}(\mu^0-\mu^*)\overline{\overline{\mathrm{G}}}(\mathbf{z},\mathbf{r}') \times \nu(\mathbf{r}')\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z}) \\ + 2\alpha^3\left(\frac{1}{\epsilon^*}-\frac{1}{\epsilon^0}\right)(\nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{z},\mathbf{r}'))^T\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\nabla \times \mathbf{H}^{\mathrm{in}}(\mathbf{z}) + \mathcal{O}(\alpha^4),\end{aligned} \tag{7.3.1}$$

as $\alpha \to 0$, for any $\mathbf{r}' \in \partial\Omega$ and ν the outward unit normal to $\partial\Omega$. The result holds for $\mathbf{z}$ bounded away from $\partial\Omega$ by some distance d_0, and holds for $0 < \alpha < \alpha_{\max}$ where $\alpha_{\max}$ depends on d_0 and ω but otherwise independent of $\mathbf{z}$.

Ammari et al also give the corollary that, for a smooth vector valued function $\mathbf{w}$ which satisfies

$$\nabla \times \nabla \times \mathbf{w} - \omega^2\epsilon^0\mu^0\mathbf{w} = 0 \quad \text{in an open neighbourhood of } \Omega, \tag{7.3.2}$$

then

$$\begin{aligned}\mathrm{i}\omega\mu^0\oint_{\partial\Omega} \mathbf{H}^{\mathrm{sc}} \times \nu \cdot \mathbf{w}\,\mathrm{d}S - \oint_{\partial\Omega} (\nabla \times \mathbf{w}) \times \nu \cdot \nu \times (\mathbf{E}^{\mathrm{sc}} \times \nu)\,\mathrm{d}S \\ =\alpha^3\omega^2\epsilon^0\mu^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot \mathbf{w}(\mathbf{z}) \\ + \alpha^3\mathrm{i}\omega\mu^0\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot \nabla \times \mathbf{w}(\mathbf{z}) + \mathcal{O}(\alpha^4)\end{aligned} \tag{7.3.3}$$

as $\alpha \to 0$.

7.3.3 Application of a surface equivalence principle

In GPR, we are interested in the scattering due to dielectric objects in the subsurface of incident waves created above the domain. In this section, we demonstrate Ammari et al's asymptotic expansion for the tangential components of $\mathbf{H}$ on a surface $\partial\Omega$, equation (7.3.3), can also be written in terms of the fields inside the domain Ω. In order to make this clarification we use the surface equivalence principle given in Section 3.7, and a further application of the theory allows us to describe the fields outside of Ω.

First, noting the quadruple product rule

$$\begin{aligned}((\nabla\times\mathbf{w})\times\nu)\cdot\nu\times(\mathbf{E}\times\nu)=&((\nabla\times\mathbf{w})\cdot\nu)\,(\nu\cdot\mathbf{E}\times\nu)\\&-((\nabla\times\mathbf{w})\cdot(\mathbf{E}\times\nu))\,(\nu\cdot\nu)\\=&-(\nabla\times\mathbf{w})\cdot(\mathbf{E}\times\nu),\end{aligned}$$

we have for (7.3.3)

$$\begin{aligned}&\mathrm{i}\omega\mu^0\oint_{\partial\Omega}\mathbf{H}\times\nu\cdot\mathbf{w}\,\mathrm{d}S+\oint_{\partial\Omega}(\nabla\times\mathbf{w})\cdot(\mathbf{E}\times\nu)\,\mathrm{d}S\\&=\alpha^3\omega^2\epsilon^0\mu^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot\mathbf{w}(\mathbf{z})\\&\quad+\alpha^3\mathrm{i}\omega\mu^0\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla\times\mathbf{w}(\mathbf{z})+\mathcal{O}(\alpha^4).\end{aligned}\tag{7.3.4}$$

We now raise $\mathbf{w}(\mathbf{z})$ to be a dyad satisfying

$$\nabla\times\nabla\times\overline{\overline{\mathrm{w}}}-\omega^2\epsilon^0\mu^0\overline{\overline{\mathrm{w}}}=\overline{\overline{0}}\quad\text{in an open neighbourhood of }\Omega,$$

where $\overline{\overline{\mathrm{w}}}=\mathbf{w}_1\mathbf{e}_x+\mathbf{w}_2\mathbf{e}_y+\mathbf{w}_3\mathbf{e}_z$ represents three different fields each satisfying the vector wave equation. Such a tensor is given by the dyadic Greens function $\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')$ for wavenumber $k^2=\epsilon^0\mu^0$, for $\mathbf{r}$ within Ω. Substituting into (7.3.4), we have

$$\begin{aligned}&\mathrm{i}\omega\mu^0\oint_{\partial\Omega}\mathbf{H}\times\nu\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\,\mathrm{d}S'+\oint_{\partial\Omega}(\nabla_{\mathbf{r}'}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}'))\cdot(\mathbf{E}\times\nu)\,\mathrm{d}S'\\&=\alpha^3\omega^2\epsilon^0\mu^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\\&\quad+\alpha^3\mathrm{i}\omega\mu^0\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla_{\mathbf{r}}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})+\mathcal{O}(\alpha^4).\end{aligned}\tag{7.3.5}$$

We notice that the integrals on the left-hand side of (7.3.5) are, by surface equivalence theorem (3.7.2), the scattered electric field[1] $\mathbf{E}^{\mathrm{sc}}(\mathbf{r})$ radiated from $\partial\Omega$ into a medium of dielectric properties ϵ^0 and μ^0,

$$\begin{aligned}\mathbf{E}^{\mathrm{sc}}(\mathbf{r})=&\alpha^3\omega^2\epsilon^0\mu^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\\&+\mathrm{i}\alpha^3\omega\mu^0\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla_{\mathbf{r}}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})+\mathcal{O}(\alpha^4).\end{aligned}\tag{7.3.6}$$

[1] In the next section we show that the contribution of the Dirac delta term in the Green's function results in $\mathbf{E}^{\mathrm{sc}}$ rather than the total field $\mathbf{E}$.

The magnetic field is then given by Maxwell's equations as

$$\begin{aligned}\mathbf{H}^{\mathrm{sc}}(\mathbf{r}) =&\frac{\mathrm{i}}{\omega\mu}\nabla_{\mathbf{r}}\times\mathbf{E}(\mathbf{r})\\ =&\mathrm{i}\alpha^3\omega\epsilon^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla_{\mathbf{r}}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\\ &+\alpha^3\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla_{\mathbf{r}}\times\nabla_{\mathbf{r}}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})+\mathcal{O}(\alpha^4),\end{aligned}\tag{7.3.7}$$

as $\alpha\to 0$, for$\mathbf{r}\in\Omega$. Since $\overline{\overline{\mathrm{G}}}$ satisfies the vector wave equation, for $\mathbf{r}\neq\mathbf{z}$ we have

$$\begin{aligned}\mathbf{H}^{\mathrm{sc}}(\mathbf{r}) =&\mathrm{i}\alpha^3\omega\epsilon^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla_{\mathbf{r}}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\\ &+\alpha^3\omega^2\epsilon^0\mu^0\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})+\mathcal{O}(\alpha^4),\end{aligned}\tag{7.3.8}$$

as $\alpha\to 0$, for$\mathbf{r}\in\Omega$. This result is similar to that of Ledger and Lionheart [95], though their result is not written explicitly in terms of a Green's function. If we choose $\mathbf{r}$ to lie on $\partial\Omega$, then it is straightforward to see that the field radiated from Ω is given by a further application of the surface equivalence theorem (3.7.2),

$$\begin{aligned}\mathbf{E}^{\mathrm{ext}}(\mathbf{r}) =&\oint_{\partial\Omega}\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\mathbf{E}(\mathbf{r}')\times\nu\,\mathrm{d}S'\\ &+\mathrm{i}k_0Z_0\oint_{\partial\Omega}\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\cdot\nu\times\mathbf{H}(\mathbf{r}')\,\mathrm{d}S'.\end{aligned}\tag{7.3.9}$$

By the duality theorem, one can fine equivalent expressions for $\mathbf{E}^{\mathrm{sc}}$ and $\mathbf{H}^{\mathrm{sc}}$ for the case where Neumann boundary conditions are applied to the Electric field $\mathbf{H}\times\nu=\frac{1}{i\omega\mu}(\nabla\times\mathbf{E})\times\nu=g$ on $\partial\Omega$. That is, by interchanging $\mathbf{E}\to\mathbf{H}$, $\mathbf{H}\to-\mathbf{E}$, $\mu\to\epsilon$ and $\epsilon\to\mu$, we have

$$\begin{aligned}\mathbf{E}^{\mathrm{sc}}(\mathbf{r}) =&\alpha^3\omega^2\epsilon^0\mu^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\\ &-\mathrm{i}\alpha^3\omega\mu^0\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla_{\mathbf{r}}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})+\mathcal{O}(\alpha^4)\end{aligned}\tag{7.3.10a}$$

$$\begin{aligned}\mathbf{H}^{\mathrm{sc}}(\mathbf{r}) =&\alpha^3\omega^2\epsilon^0\mu^0\left(\frac{\mu^0}{\mu^*}-1\right)\left[\mathscr{M}_B\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{z})\right]\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\\ &-\mathrm{i}\alpha^3\omega\epsilon^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\left[\mathscr{M}_B\left(\frac{\epsilon^0}{\epsilon^*}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right]\cdot\nabla_{\mathbf{r}}\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})+\mathcal{O}(\alpha^4)\end{aligned}\tag{7.3.10b}$$

as $\alpha\to 0$, for$\mathbf{r}\in\Omega$. Equations (7.3.10) are directly applicable where we have solved the weak form of the vector wave equation for the E-field, and therefore applied Neumann conditions to $\mathbf{E}$. We notice the expression for $\mathbf{H}$ is the same, but that the expression for $\mathbf{E}$ has a negative $\mathbf{H}^{\mathrm{in}}$ component.

7.3.4 Proof of the asymptotic result

In the previous section, we used the corollary of Ammari et al [14] for the case when $\mathbf{w}$ is raised to the rank-2 tensor $\overline{\overline{\mathrm{G}}}$. Note though that our choice of $\overline{\overline{\mathrm{G}}}$ does not in fact satisfy (7.3.2) at $\mathbf{r} = \mathbf{z}$ – rather it satisfies (7.3.2) only in a *punctured* neighbourhood of Ω. We also note that when we applied the surface equivalence theorem in Section 7.3.3, the full fields $\mathbf{E}$ and $\mathbf{H}$ in the integrands resulted in $\mathbf{E}^{\mathrm{sc}}$ and $\mathbf{H}^{\mathrm{sc}}$ after application of the theorem. In this section we follow closely the proof of corollary given by Ammari et al [14, Section 7, 8] for the case where $\mathbf{w}$ has been raised to the rank 2 tensor $\overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}')$. We will find that the contribution given by the in $\overline{\overline{\mathrm{G}}}$, owing to the Dirac delta term in the vector wave equation, results precisely in a $-\mathbf{E}^{\mathrm{in}}$ term. Thus, on application of the surface equivalence principle for $\mathbf{E}$, one is left precisely with $\mathbf{E}^{\mathrm{sc}}$ as was presented in the previous section.

As with Ammari et al, for simplicity let the position of the inclusion be $\mathbf{z} = \mathbf{0} \in \Omega$. Let $\overline{\overline{\mathrm{G}}}_{m2}$ be the magnetic Green's function of the second kind [166, pp 68] which satisfies

$$\begin{aligned} \nabla \times \frac{1}{\epsilon^0} \nabla \times \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}') - \omega^2 \mu^0 \overline{\overline{\mathrm{G}}}_{m2} &= -\delta(\mathbf{r} - \mathbf{r}') I \quad \text{in } \Omega \\ \frac{1}{\epsilon^0} \left(\nabla \times \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}') \right) \times \hat{\mathbf{n}}(\mathbf{r}) &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{7.3.11}$$

where ∇ is with respect to $\mathbf{r}$. For any $\mathbf{z} \in \Omega \setminus \overline{\alpha B}$ we have

$$\begin{aligned} (\mathbf{H} - \mathbf{H}^{\mathrm{in}})(\mathbf{r}) = &- \int_{\Omega} \left(\nabla \times \frac{1}{\epsilon^0} \nabla \times -\omega^2 \mu^0 \right) \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}') (\mathbf{H} - \mathbf{H}^{\mathrm{in}})(\mathbf{r}') \, \mathrm{d}V' \\ = &\int_{\partial\Omega} \frac{1}{\epsilon^0} \left((\nabla \times \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}')) \times \hat{\mathbf{n}} \right)^T (\mathbf{H} - \mathbf{H}^{\mathrm{in}})(\mathbf{r}') \, \mathrm{d}S' \\ &- \int_{\partial\Omega} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}') \frac{1}{\epsilon^0} (\nabla \times ((\mathbf{H} - \mathbf{H}^{\mathrm{in}})(\mathbf{r})) \times \hat{\mathbf{n}}) \, \mathrm{d}S' \\ &- \int_{(\Omega \setminus \overline{\alpha B}) \cup \alpha B} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}') \left(\nabla \times \frac{1}{\epsilon^0} \nabla \times -\omega^2 \mu^0 \right) (\mathbf{H} - \mathbf{H}^{\mathrm{in}})(\mathbf{r}') \, \mathrm{d}V' \\ &+ \int_{\partial(\alpha B)} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}') \frac{1}{\epsilon^0} \left((\nabla \times (\mathbf{H} - \mathbf{H}^{\mathrm{in}})(\mathbf{r}')^+ \times \hat{\mathbf{n}} \right) \, \mathrm{d}S' \\ &- \int_{\partial(\alpha B)} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}, \mathbf{r}') \frac{1}{\epsilon^0} \left((\nabla \times (\mathbf{H} - \mathbf{H}^{\mathrm{in}})(\mathbf{r}')^+ \times \hat{\mathbf{n}} \right) \, \mathrm{d}S'. \end{aligned} \tag{7.3.12}$$

In the above, we have added and removed integrals $\int \nabla \times \nabla \times (\mathbf{H} - \mathbf{H}^{\mathrm{in}}) \, \mathrm{d}V'$ using the divergence theorem.

Due to the common boundary conditions, the second term in (7.3.12) is zero. The first term is also zero, due to the boundary conditions for $\overline{\overline{\mathrm{G}}}_{m2}$ on $\partial\Omega$. Our choice of Green's function does not affect the other integrals compared with Ammari et al, so we may follow their manipulations directly. These involve the introduction of two auxiliary fields, which are left out here for the sake of brevity, and bring us to the asymptotic formula

$$\begin{aligned}(\mathbf{H}-\mathbf{H}^{\mathrm{in}})(\mathbf{r}') =&\alpha^3\omega^2(\mu^0-\mu^*)\frac{\mu^0}{\mu^*}\overline{\overline{\mathrm{G}}}_{m2}(\mathbf{0},\mathbf{r}')^T\mathscr{M}\left(\frac{\mu^0}{\mu^*}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{0})\\ &+\alpha^3\left(\frac{1}{\epsilon^*}-\frac{1}{\epsilon^0}\right)\left(\nabla\times\overline{\overline{\mathrm{G}}}_{m2}(\mathbf{0},\mathbf{r}')\right)^T\mathscr{M}\left(\frac{\epsilon^0}{\epsilon^*}\right)\nabla\times\mathbf{H}^{\mathrm{in}}(\mathbf{0})+\mathcal{O}(\alpha^4).\end{aligned}\tag{7.3.13}$$

Now, we wish to use (7.3.13) to find an asymptotic formula for

$$\frac{\mathrm{i}}{\omega\epsilon^0}\left(\int_{\partial\Omega}\left((\nabla\times\mathbf{H})(\mathbf{r}')\times\hat{\mathbf{n}}\right)\cdot\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\,\mathrm{d}S'-\omega^2\mu^0\epsilon^0\int_{\partial\Omega}\mathbf{H}(\mathbf{r}')\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}\,\mathrm{d}S'\right).$$

Here we first differ from Ammari et al, in that we have

$$\begin{aligned}&\int_{\partial\Omega}\left((\nabla\times\mathbf{H}^{\mathrm{in}})(\mathbf{r}')\times\hat{\mathbf{n}}\right)\cdot\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\,\mathrm{d}S'\\ &-\omega^2\mu^0\epsilon^0\int_{\partial\Omega}\mathbf{H}^{\mathrm{in}}(\mathbf{r}')\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}\,\mathrm{d}S'=\mathbf{E}^{\mathrm{in}},\end{aligned}\tag{7.3.14}$$

by the surface equivalence theorem (3.7.2), where Ammari et al. have a right hand side of $\mathbf{0}$. Using (7.3.13) and (7.3.14), as well as the shared boundary condition for $\mathbf{H}$ and $\mathbf{H}^{\mathrm{in}}$, we have that

$$\begin{aligned}&\int_{\partial\Omega}\left((\nabla\times\mathbf{H})(\mathbf{r}')\times\hat{\mathbf{n}}(\mathbf{r}')\right)\cdot\nabla'\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\,\mathrm{d}S'\\ &-\omega^2\mu^0\epsilon^0\int_{\partial\Omega}\mathbf{H}(\mathbf{r}')\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}(\mathbf{r}')\,\mathrm{d}S'-\mathbf{E}^{\mathrm{in}}\\ &\quad=\int_{\partial\Omega}\left(\nabla'\times\mathbf{H}(\mathbf{r}')-\nabla'\times\mathbf{H}^{\mathrm{in}}(\mathbf{r}')\right)\times\hat{\mathbf{n}}(\mathbf{r}')\cdot\nabla'\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\,\mathrm{d}S'\\ &\qquad-\omega^2\mu^0\epsilon^0\int_{\partial\Omega}\left(\mathbf{H}(\mathbf{r}')-\mathbf{H}^{\mathrm{in}}(\mathbf{r}')\right)\cdot\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}(\mathbf{r}')\,\mathrm{d}S'\\ &\quad=-\alpha^3\omega^2\mu^0\epsilon^0(\mu^0-\mu^*)\frac{\mu^0}{\mu^1}\int_{\partial\Omega}\overline{\overline{\mathrm{G}}}_{m2}(\mathbf{0},\mathbf{r}')\left(\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}(\mathbf{r}')\right)\,\mathrm{d}S'\cdot\mathscr{M}\left(\frac{\mu^0}{\mu^1}\right)\mathbf{H}^{\mathrm{in}}(\mathbf{0})\\ &\qquad-\alpha^3\omega^2\mu^0\left(\frac{\epsilon^0}{\epsilon^*}-1\right)\int_{\partial\Omega}\left(\nabla'\times\overline{\overline{\mathrm{G}}}_{m2}(\mathbf{0},\mathbf{r}')\right)\left(\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}(\mathbf{r}')\right)\,\mathrm{d}S'\cdot\mathscr{M}\left(\frac{\epsilon^0}{\epsilon^*}\right)\nabla\times\mathbf{H}^{\mathrm{in}}(\mathbf{0})\\ &\qquad+\mathcal{O}(\alpha^4).\end{aligned}\tag{7.3.15}$$

To rewrite (7.3.15) in the form of (7.3.6), we first note that for $\mathbf{r}\neq\mathbf{r}'$,

$$\int_{\partial\Omega}\overline{\overline{\mathrm{G}}}_{m2}(\mathbf{0},\mathbf{r})\left(\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}(\mathbf{r}')\right)\,\mathrm{d}S'=-\frac{1}{\omega^2\mu^0}\nabla\times\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{0})\tag{7.3.16}$$

and

$$\int_{\partial\Omega}\left(\nabla'\times\overline{\overline{\mathrm{G}}}_{m2}(\mathbf{0},\mathbf{r})\right)\left(\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}')\times\hat{\mathbf{n}}(\mathbf{r}')\right)\,\mathrm{d}S'=-\epsilon^0\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{0}).\tag{7.3.17}$$

The first of these follows from application the divergence theorem,

$$\begin{aligned}
&\int_{\partial\Omega} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{z},\mathbf{r}') \left(\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \times \hat{\mathbf{n}}(\mathbf{r}')\right) \mathrm{d}S' \\
&= - \int_{\partial\Omega} \left(\overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}',\mathbf{z}) \times \hat{\mathbf{n}}(\mathbf{r}')\right)^T \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}S' \\
&= \int_{\Omega} \left(\nabla' \times \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}',\mathbf{z})\right)^T \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}V' - \int_{\Omega} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}',\mathbf{z})^T \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}V' \\
&= \frac{1}{\omega^2\mu^0\epsilon^0} \int_{\Omega} \left(\nabla' \times \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}',\mathbf{z})\right)^T \nabla' \times \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}V' \\
&\quad - \int_{\Omega} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}',\mathbf{z})^T \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}V' \\
&= \frac{1}{\omega^2\mu^0\epsilon^0} \int_{\Omega} \left(\nabla' \times \nabla' \times \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}',\mathbf{z})\right)^T \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}V' \\
&\quad - \int_{\Omega} \overline{\overline{\mathrm{G}}}_{m2}(\mathbf{r}',\mathbf{z})^T \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}V' \\
&= - \frac{1}{\omega^2\mu^0} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z}).
\end{aligned} \tag{7.3.18}$$

In the above, we have used that both $\overline{\overline{\mathrm{G}}}_{m2}$ and $\overline{\overline{\mathrm{G}}}$ satisfy vector wave equations and that $\mathbf{r} \neq \mathbf{z} \notin \partial\Omega$. Setting $\mathbf{z} = \mathbf{0}$, (7.3.16) follows directly. (7.3.17) follows simply by taking the curl of (7.3.16) and substituting for $\nabla \times \nabla \times \overline{\overline{\mathrm{G}}}$.

Inserting formulas (7.3.16) and (7.3.17) into (7.3.15), one obtains

$$\begin{aligned}
&\frac{\mathrm{i}}{\omega\epsilon^0} \Bigg(\int_{\partial\Omega} ((\nabla \times \mathbf{H})(\mathbf{r}') \times \hat{\mathbf{n}}(\mathbf{r}')) \cdot \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \,\mathrm{d}S' \\
&\quad -\omega^2\mu^0\epsilon^0 \int_{\partial\Omega} \mathbf{H}(\mathbf{r}') \cdot \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{r}') \times \hat{\mathbf{n}}(\mathbf{r}') \,\mathrm{d}S' - \mathbf{E}^{\mathrm{in}} \Bigg) \\
&= \alpha^3 \mathrm{i}\omega\mu^0 \left(\frac{\mu^0}{\mu^*} - 1\right) \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{0}) \mathscr{M}\left(\frac{\mu^0}{\mu^1}\right) \mathbf{H}^{\mathrm{in}}(\mathbf{0}) \\
&\quad + \alpha^3 \mathrm{i}\omega\mu^0 \left(\frac{\epsilon^0}{\epsilon^1} - 1\right) \overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{0}) \mathscr{M}\left(\frac{\epsilon^0}{\epsilon^*}\right) \nabla \times \mathbf{H}^{\mathrm{in}}(\mathbf{0}) + \mathcal{O}(\alpha^4).
\end{aligned} \tag{7.3.19}$$

Application of the surface equivalence theorem for $\mathbf{E}(\mathbf{r})$ and multiplication through by $-\mathrm{i}\omega\epsilon^0$ yields the left hand side of $\mathbf{E}^{\mathrm{sc}} = \mathbf{E} - \mathbf{E}^{\mathrm{in}}$, and substitution for $\nabla \times \mathbf{H}^{\mathrm{in}}$ from Maxwell's equations yields the required result.

7.3.5 Expressing the asymptotic expansion as a linear operator on $\mathscr{M}$

It is useful and somewhat informative to write the approximation of a GPR dataset using asymptotic formula (7.3.10) as a linear operator D on the tensor $\mathscr{M}$. We consider the case of a single inclusion at $\mathbf{z}$ with a contrast only in ϵ (i.e. $\mu^* = \mu^0$), but the result is easily extended to the more general case. $\mathscr{M}$ has 6 different values which we store in the vector $\mathbf{m} = [\mathscr{M}_{11}, \mathscr{M}_{12}, \mathscr{M}_{13}, \mathscr{M}_{22}, \mathscr{M}_{23}, \mathscr{M}_{33}]^T = [m_1, \ldots, m_6]^T$. With this notation, we first write the product $\mathscr{M}\mathbf{E}^{\text{in}}$ as

$$\mathscr{M}\mathbf{E}^{\text{in}} = \begin{bmatrix} E_{0x} & E_{0y} & E_{0z} & 0 & 0 & 0 \\ 0 & E_{0x} & 0 & E_{0y} & E_{0z} & 0 \\ 0 & 0 & E_{0x} & 0 & E_{0y} & E_{0z} \end{bmatrix} \mathbf{m} \tag{7.3.20}$$

where $\mathbf{E}^{\text{in}} = [E_{0x}, E_{0y}, E_{0z}]^T$. In this form, the expression (7.3.10) can be written as

$$\begin{aligned} \mathbf{E}^{\text{sc}}(\mathbf{r}, \omega) =& \alpha^3 \omega^2 \epsilon^0 \mu^0 \left(\frac{\epsilon^0}{\epsilon^*} - 1 \right) \begin{bmatrix} G_{11}E_{0x} & \substack{G_{11}E_{0y} \\ +G_{12}E_{0x}} & \substack{G_{11}E_{0z} \\ +G_{13}E_{0x}} & G_{12}E_y & \substack{G_{12}E_{0z} \\ +G_{13}E_{0y}} & G_{13}E_{0z} \\ G_{21}E_{0x} & \substack{G_{21}E_{0y} \\ +G_{22}E_{0x}} & \substack{G_{21}E_{0z} \\ +G_{23}E_{0x}} & G_{22}E_{0y} & \substack{G_{22}E_{0z} \\ +G_{23}E_{0y}} & G_{23}E_{0z} \\ G_{31}E_{0x} & \substack{G_{31}E_{0y} \\ +G_{32}E_{0x}} & \substack{G_{31}E_{0z} \\ +G_{33}E_{0x}} & G_{32}E_{0y} & \substack{G_{32}E_{0z} \\ +G_{33}E_{0y}} & G_{33}E_{0z} \end{bmatrix} \mathbf{m} \\ =& A_e^j(\mathbf{r}, \mathbf{z}, \omega)\mathbf{m}, \end{aligned} \tag{7.3.21}$$

where G_{ij} are the elements of $\overline{\overline{\mathrm{G}}}$, and we have absorbed the constants into $A_e^j(\mathbf{r}, \mathbf{z}, \omega)$. We have used the superscript j to denote that A_e^j is associated with the j^{th} source field. Similarly, the scattered magnetic field can be written as

$$\mathbf{H}^{\text{sc}}(\mathbf{r}, \omega) = A_m^j(\mathbf{r}, \mathbf{z}, \omega)\mathbf{m}, \tag{7.3.22}$$

where A_m^j differs to A_e^j by a constant as well as use of the magnetic Greens function $\overline{\overline{\mathrm{G}}}_m = \nabla \times \overline{\overline{\mathrm{G}}}$ rather than $\overline{\overline{\mathrm{G}}}$. The fields external to Ω are found by integrals of the tangential components of $\mathbf{E}^{\text{sc}}$ and $\mathbf{H}^{\text{sc}}$ on $\partial\Omega$, which are given by

$$(\hat{\mathbf{n}} \times \mathbf{E}^{\text{sc}})(\mathbf{r}) = \hat{\mathbf{n}} \times (A_e^j \mathbf{m}) = N A_e^j(\mathbf{r}, \mathbf{z}, \omega)\mathbf{m}, \tag{7.3.23a}$$

$$(\hat{\mathbf{n}} \times \mathbf{H}^{\text{sc}})(\mathbf{r}) = \hat{\mathbf{n}} \times (A_m^j \mathbf{m}) = N A_m^j(\mathbf{r}, \mathbf{z}, \omega)\mathbf{m}, \tag{7.3.23b}$$

for $\mathbf{r} \in \partial\Omega$, where N is the skew-symmetric matrix equivalent to $\hat{\mathbf{n}}\times$ at surface location $\mathbf{r}$, given by

$$N = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}.$$

Carrying out the surface equivalence integration over $\partial\Omega$ numerically involves the summation of these surface tangential components at integration points $\mathbf{r}_i$ as

$$\begin{aligned} \mathbf{E}^{\text{sc}}(\mathbf{r}) =& \sum_i w_i \left[-\nabla \times \overline{\overline{\mathrm{G}}}_0(\mathbf{r}, \mathbf{r}_i) N^i A_e^j(\mathbf{r}_i, \mathbf{z}, \omega) + ik_0 Z_0 \overline{\overline{\mathrm{G}}}(\mathbf{r}, \mathbf{r}_i) N^i A_m^j(\mathbf{r}_i, \mathbf{z}) \right] \mathbf{m} \\ =& A^j(\mathbf{r}, \mathbf{z})\mathbf{m}, \qquad \mathbf{r} \in \mathbb{R}^3 \setminus \Omega \end{aligned} \tag{7.3.24}$$

where N^i represents the operation $\hat{\mathbf{n}}\times$ at location $\mathbf{r}_i$, and the matrix A^j is the sum of integration weights w_i times the matrices within the square brackets. Accordingly, the data element d_{jk}, the k^{th} recorded voltage from the j^{th} source field, is given by

$$d_{jk} = \mathbf{p}_k \cdot \mathbf{E}^{\text{sc}}(\mathbf{r}_k) = \mathbf{p}_k^T \mathbf{E}^{\text{sc}} = \mathbf{p}^T A^j(\mathbf{r}_k, \mathbf{z}, \omega)\mathbf{m}, \tag{7.3.25}$$

and so the part of the data $\mathbf{d}_j$ due to the j^{th} source field may be written

$$\mathbf{d}_j = \begin{bmatrix} \mathbf{p}_1^T A^j(\mathbf{r}_{j1}, \mathbf{z}) \\ \vdots \\ \mathbf{p}_k^T A^j(\mathbf{r}_{jK}, \mathbf{z}) \end{bmatrix} \mathbf{m} = D_j \mathbf{m}. \tag{7.3.26}$$

The total data set can then be written as the matrix operation

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_J \end{bmatrix} = \begin{bmatrix} D_1 \\ \vdots \\ D_J \end{bmatrix} \mathbf{m} = D\mathbf{m}. \tag{7.3.27}$$

This step by step formulation of D makes clear what is happening in equations (7.3.10). First, the product $\mathscr{M}\mathbf{E}^{\text{in}}$ mixes the components of $\mathbf{E}^{\text{in}}$, and acts as a dipole source, with the field propagated to $\mathbf{r}_i$ by the Greens function. Equivalently, the operation $A_e^j\mathbf{m}$ mixes the components of $\mathbf{m}$, but only those which have an affect on the polarization of $\mathbf{E}^{\text{in}}$. For example, with an x-polarised incident field, A_e^j will only act on $\mathscr{M}_{11}$, $\mathscr{M}_{12}$ and $\mathscr{M}_{13}$. As $\mathbf{E}^{\text{in}}$ and $\mathbf{H}^{\text{in}}$ are perpendicular to one another, if they are propagating in the z-direction then the product $A_m^j\mathbf{m}$ will involve only $\mathscr{M}_{12}$, $\mathscr{M}_{22}$ and $\mathscr{M}_{23}$. In both cases, the electric or magnetic Green's functions will further scale the contributions of the terms in each of the three components of $\mathscr{M}\mathbf{E}^{\text{in}}$, being null in the direction of propagation. Finally, the operators A^j account for the change in polarization and direction of propagation at the ground surface, and the data terms are a linear combination of the elements of $A^j\mathbf{m}$ (given by the polarization of receivers $\mathbf{p}$). The polarization of the incident field at $\mathbf{z}$ determines which components of $\mathscr{M}$ have an affect on the scattered wave, and the position and polarization of receiving antennas affects what linear combination of these elements $\mathscr{M}_{ij}$ appear in the data.

7.4 Understanding the polarization tensor as sensitivity

7.4.1 Comparison to the Taylor series

When we carry out non-linear least-squares imaging such as FWI, one often refers to the Fréchet derivative, or Jacobian matrix J, of the objective functional $\mathcal{J}$,

$$\nabla_{\delta m}\mathcal{J}(m) = J(m)\delta m, \tag{7.4.1}$$

as the sensitivity. The idea that the Jacobian matrix is the the sensitivity comes naturally: J_{ij} tells us the rate of change of a simulated datum with respect to a single parameter. The greater J_{ij}, the more sensitive the datum is to the parameter. This idea is only valid when the first order approximation,

$$\mathcal{J}(m+\delta m) = \mathcal{J}(m) + J(m)\delta m + \mathcal{O}(|\delta m|^2) \tag{7.4.2}$$

holds, and so δm must be small in the L_∞ norm. That is, it is valid for any *shape* perturbation δm, but the supremum of δm is small. Note that (7.4.2) is simply the first-order Taylor series approximation, or the Born approximation, and so is a single scattering approximation. Higher order terms account for higher order multiple scattering, and with the complete series (which is then exact) we can account for the saturation effect (that if an object is sufficiently conductive, say, making it more conductive will have little effect). If we are considering a large (L_∞) change in parameter, the Jacobian matrix may offer far less insight into how sensitive our data is to such an object.

Asymptotic expansions using the polarisation tensor give us another natural view of sensitivity, since they give us an expression of the change in field due to the inclusion of a small object, or a collection of small objects sufficiently separated, in a homogeneous domain. These can be of any contrast in permittivity, conductivity and permeability, and any shape. I.e. they may have a large L_∞ norm, but the small volume results in a small L_2 norm. Unlike J_{ij} this is not a rate of change (though trivially we can differentiate it), but it does include the non-linear saturation effect which the Taylor series would require the entire series to accurately represent. If we are interested in the sensitivity of our measurements to individual high contrast objects then it may be more applicable. Also in contrast, the approximation (7.4.2) is valid at any m for which $\mathcal{J}$ is Fréchet differentiable, unlike the polarization tensor approximation which is valid only for a homogeneous background.

Recall also that by the adjoint-state method (2.5.20), columns of the Jacobian (in discretised form) are given by

$$J_{\cdot i} = A^{-1}\left(-\frac{\partial A}{\partial m_i}\mathbf{u}\right),$$

where A is the finite difference or finite element system matrix and $\mathbf{u}$ is the numerical solution. The operator $\frac{\partial A}{\partial m_i}$ plays a similar role here to polarisation tensor $\mathscr{M}$: reflecting the incident wave which is then propagated to receivers. The two approximations then are indeed of a similar form, though one of the niceties of the asymptotic expansion in this regard is that $\mathscr{M}$ is invariant to the position of the parameter.

We then have two ideas of sensitivity. The first, and more traditional, uses the first order Taylor series expansion, and is valid for any parameterisation (or any size and shape object) but only for a small (supremum) change in material parameters. It is also applicable for an inhomogeneous domain. The second, using the polarisation tensors, is valid only when we have small objects, but for any material contrast. Higher order terms add information about how the object differs from an ellipsoid, but can never account for multiple scattering.

7.4.2 Resolution in 3D FWI

To analyse sensitivity in full-wave inversion to experimental setup (e.g. frequency, source-receiver offset), Sirgue and Pratt [159] (see also [181]) use the scalar Green's functions for the Helmholtz equation to approximate the wave field terms in the gradient. Consider a homogeneous background model with wavespeed c_0, an incident monochromatic plane wave of angular frequency ω in direction $\hat{\mathbf{s}}$, and a scattered plane wave in direction $\hat{\mathbf{r}}$ (in the far-field approximation). Ignoring amplitude effects, the Green's functions are

$$\begin{aligned} G_0(\mathbf{x},\mathbf{s}) &= \exp(\mathrm{i}k_0\hat{\mathbf{s}}\cdot\mathbf{x}), \\ G_0(\mathbf{x},\mathbf{r}) &= \exp(\mathrm{i}k_0\hat{\mathbf{r}}\cdot\mathbf{x}), \end{aligned} \tag{7.4.3}$$

where $\mathbf{s}$ and $\mathbf{r}$ are the source location and receiver locations, respectively, and $k_0 = \omega/c_0$. Using these expressions for the incident and scattered fields in the gradient of the misfit function (2.5.22) we have (in the continuous setting) [159]

$$\begin{aligned}\nabla\mathcal{J}(\mathbf{x}) &= -\sum_{\omega}\sum_{s}\sum_{r}\omega^2\Re\left\{\exp(-\mathrm{i}k_0\hat{\mathbf{s}}\cdot\mathbf{x})\exp(-\mathrm{i}k_0\hat{\mathbf{r}}\cdot\mathbf{x})\delta d(\mathbf{r},\mathbf{s})\right\}\\ &= -\sum_{\omega}\sum_{s}\sum_{r}\omega^2\Re\left\{\exp(-\mathrm{i}k_0(\hat{\mathbf{s}}+\hat{\mathbf{r}})\cdot\mathbf{x})\delta d(\mathbf{r},\mathbf{s})\right\}.\end{aligned} \tag{7.4.4}$$

They remark that (7.4.4) has the form of a truncated Fourier series, in which the coefficients are the data residuals and integration variable is the scattering wavenumber vector $\mathbf{k} = k_0(\hat{\mathbf{s}}+\hat{\mathbf{r}})$. This can be expressed as

$$\mathbf{k} = \frac{2f}{c_0}\cos\left(\frac{\theta}{2}\right)\hat{\mathbf{n}}, \tag{7.4.5}$$

where $\hat{\mathbf{n}}$ is a unit vector in direction $\hat{\mathbf{s}}+\hat{\mathbf{r}}$, and θ the angle between $\hat{\mathbf{s}}$ and $\hat{\mathbf{r}}$. From (7.4.5) we see that one frequency and one offset (aperture) in the data space map to one wavenumber in the model space, and so give redundant control of wavenumber coverage. We also see that low frequency and wide offsets help resolve large length-scale features of the medium, and that highest -frequency small-offset ($\theta = 0$) leads to a maximum resolution of half a wavelength. Finally, larger offsets will be helpful for resolving small-scale horizontal features. This analysis relies on the Born approximation, which neglects features of vector wave scattering in 3D such as polarization.

To extend the above analysis to electromagnetic wave scattering in 3D, we consider the case where the forward problem can be written in the form

$$\mathbf{E}^{\mathrm{sc}}(\mathbf{r}) = \omega^2\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\mathscr{M}_B(m)\mathbf{E}^{\mathrm{in}}(\mathbf{z}) + \mathcal{O}(\alpha^4) \tag{7.4.6}$$

as $\alpha \to 0$. This is the equivalent expression to (7.3.10) with the additional constants absorbed into $\mathscr{M}$, which is valid either for an infinite domain, or a finite domain Ω with measurement positions $\mathbf{r}$ within or on the boundary $\partial\Omega$. Without loss of generality, we consider a single scattering object B at $\mathbf{z}$.

The 2-norm data misfit cost function is given by

$$\mathcal{J}(m) = h\big(\mathbf{E}^{\mathrm{sc}}(m), m\big) = \sum_{\omega}\sum_{s}\sum_{r}\frac{1}{2}\|\mathcal{R}_{s,r}\mathbf{E}^{\mathrm{sc}}(\omega,m) - d_{s,r}(\omega)\|_2^2, \tag{7.4.7}$$

where $\mathcal{R}_{s,r} : \mathbf{E} \mapsto d$ is a restriction matrix onto receiver r for source s (in the relevant polarization). We use the adjoint state method to calculate the gradient of (7.4.7) with respect to m, which requires the augmented functional (Lagrangian) associated with the minimisation problem

$$\begin{aligned}\mathcal{L}(\tilde{\mathbf{E}},\tilde{\lambda},m) =&\Re\Bigg\{\frac{1}{2}\sum_{\omega}\sum_{s}\sum_{r}\|\mathcal{R}_{s,r}\tilde{\mathbf{E}}(\omega)-d_{s,r}(\omega)\|_2^2-\\ &\sum_{\omega}\sum_{s}\left\langle\tilde{\lambda}(\omega),\frac{\partial\mathcal{F}}{\partial m}\right\rangle\Bigg\}\\ =&\Re\Bigg\{\frac{1}{2}\sum_{\omega}\sum_{s}\sum_{r}\|\mathcal{R}_{s,r}\tilde{\mathbf{E}}(\omega)-d_{s,r}(\omega)\|_2^2-\\ &\sum_{\omega}\sum_{s}\left\langle\tilde{\lambda}(\omega),\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\frac{\mathscr{M}_B}{\partial m}\mathbf{E}^{\mathrm{in}}(\mathbf{z})\right\rangle\Bigg\},\end{aligned} \tag{7.4.8}$$

where $\mathcal{F}$ is the forward operator

$$\mathcal{F}=\omega^2\overline{\overline{\mathrm{G}}}(\mathbf{r},\mathbf{z})\mathscr{M}_B(m)\mathbf{E}^{\mathrm{in}}(\mathbf{z})-\mathbf{E}^{\mathrm{sc}}(\mathbf{r})=0. \tag{7.4.9}$$

The adjoint state is defined by $\partial\mathcal{L}(\mathbf{E}^{\mathrm{sc}},\lambda,m)/\partial\tilde{\mathbf{E}}=0$, which gives

$$\lambda_s(\omega)=\sum_r\mathcal{R}^*_{s,r}(\mathcal{R}_{s,r}\mathbf{E}^{\mathrm{sc}}_s(\omega)-d_{s,r}(\omega). \tag{7.4.10}$$

The gradient is then given by

$$\frac{\partial\mathcal{J}}{\partial m}=\frac{h(\mathbf{E}^{\mathrm{sc}},m)}{\partial m}-\left\langle\lambda,\frac{\partial\mathcal{F}(\mathbf{E}^{\mathrm{sc}},m)}{\partial m})\right\rangle, \tag{7.4.11}$$

and since $\partial h/\partial m=0$ (h depends on m only through $\mathbf{E}^{\mathrm{sc}}$)

$$\frac{\partial\mathcal{J}}{\partial m}=-\Re\left\{\sum_{\omega}\sum_{s}\left\langle\mathcal{R}^*_{s,r}\delta d_{s,r}(\omega),\omega^2\overline{\overline{\mathrm{G}}}\frac{\partial\mathscr{M}}{\partial m}\mathbf{E}^{\mathrm{in}}_s\right\rangle\right\}. \tag{7.4.12}$$

Assuming a point source $\mathbf{E}^{\mathrm{in}}_s(\mathbf{z})=\overline{\overline{\mathrm{G}}}(\mathbf{s},\mathbf{z})\mathbf{p}_s$, and $\mathcal{R}_{s,r}$ takes polarization $\mathbf{p}_r$ at $\mathbf{r}$, the gradient is given by

$$\nabla\mathcal{J}=-\sum_{\omega}\sum_{s}\sum_{r}\omega^2\Re\left\{\mathbf{p}_r\cdot\overline{\overline{\mathrm{G}}}^*(\mathbf{r},\mathbf{z})\frac{\partial\mathscr{M}}{\partial m}\overline{\overline{\mathrm{G}}}^*(\mathbf{z},\mathbf{s})\mathbf{p}_s\delta d_{s,r}(\omega)\right\}. \tag{7.4.13}$$

The similarity to (7.4.4) is clear, though the result is valid in the near field and for vector-valued waves.

The term $\partial\mathscr{M}/\partial m$ accounts for the shape of the scatterer B (as well as the saturation effect). If we are using a discretisation of the subsurface with elements of approximately of equal length in each coordinate direction this term could be neglected (being a scale multiple of the identity). In some applications other shaped discretisations are used though, for example if we have a layer-like structure one might use flatter elements. In this case, we have

$$\frac{\partial \mathscr{M}}{\partial m} = c \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad c > 0, \tag{7.4.14}$$

and so the $\hat{\mathbf{z}}$ component of $\overline{\overline{\mathrm{G}}}(\mathbf{z}, \mathbf{s})\mathbf{p}_s$ would largely give little contribution to the gradient. Moreover, we can see that an acquisition array on the ground surface we would gain most from having $\mathbf{p}_s = \mathbf{p}_r$.

Using the far-field approximation and neglecting amplitude as was done in (7.4.4), we have

$$\begin{aligned} \nabla \mathcal{J} &= -\sum_\omega \sum_s \sum_r \omega^2 \Re \left\{ {\mathbf{p}'_r}^T \exp(-\mathrm{i}k_0 \hat{\mathbf{r}} \cdot \mathbf{z}) \frac{\partial \mathscr{M}}{\partial m} \exp(-\mathrm{i}k_0 \hat{\mathbf{s}} \cdot \mathbf{z}) \mathbf{p}'_s \delta d_{s,r}(\omega) \right\}, \\ &= -\sum_\omega \sum_s \sum_r \omega^2 \Re \left\{ {\mathbf{p}'_r}^T \frac{\partial \mathscr{M}}{\partial m} \mathbf{p}'_s \exp\big(-\mathrm{i}k_0 (\hat{\mathbf{s}} + \hat{\mathbf{r}}) \cdot \mathbf{z}\big) \delta d_{s,r}(\omega) \right\} \end{aligned} \tag{7.4.15}$$

where $\mathbf{p}'_s \perp \hat{\mathbf{s}}$ and $\mathbf{p}'_r \perp \hat{\mathbf{r}}$ are the components of source and receiver polarizations orthogonal to $\hat{\mathbf{s}}$ and $\hat{\mathbf{r}}$, respectively.

We can draw similar conclusions from (7.4.15) as were made by Sirgue and Pratt [159] and Virieux [181] with regards to equation (7.4.4). For vector valued waves, the (complex) scattering wavenumber vector (7.4.5) can be expressed as

$$\begin{aligned} \mathbf{k} =& \frac{2f}{c_0} \cos\left(\frac{\theta}{2}\right) \hat{\mathbf{n}} + \mathrm{i} \log\left({\mathbf{p}'_r}^T \frac{\partial \mathscr{M}}{\partial m} \mathbf{p}'_s \right) \frac{\mathbf{z}}{\mathbf{z} \cdot \mathbf{z}} \\ =& \frac{2f}{c_0} \cos\left(\frac{\theta}{2}\right) \hat{\mathbf{n}} + \mathrm{i} \log\left(\big(\mathbf{p}_r - \cos\varphi_r \hat{\mathbf{r}}\big)^T \frac{\partial \mathscr{M}}{\partial m} \big(\mathbf{p}_s - \cos\varphi_s \hat{\mathbf{s}}\big) \right) \frac{\mathbf{z}}{\mathbf{z} \cdot \mathbf{z}}, \end{aligned} \tag{7.4.16}$$

where φ_s is the angle between source polarization $\mathbf{p}_s$ and source wave direction $\hat{\mathbf{s}}$, φ_r the angle between receiver polarization $\mathbf{p}_r$ and scattered wave direction $\hat{\mathbf{r}}$, and we have assumed that $|\mathbf{p}_r| = |\mathbf{p}_s| = 1$. In the above, we have used

$$a \exp\big(-\mathrm{i}\mathbf{k} \cdot \mathbf{z}\big) = \exp\big(\log(a) - \mathrm{i}\mathbf{k} \cdot \mathbf{z}\big) = \exp\left\{ -\mathrm{i} \left(\mathbf{k} + \mathrm{i} \log(a) \frac{\mathbf{z}}{\mathbf{z} \cdot \mathbf{z}} \right) \cdot \mathbf{z} \right\}.$$

Without loss of generality, we assume ${\mathbf{p}'_r}^T (\partial \mathscr{M} / \partial m) \mathbf{p}'_s \geq 0$. Equivalently, we assume ${\mathbf{p}_r}^T \mathbf{p}_s \geq 0$, since $\mathscr{M}$ (and it's derivative) is positive definite, and $\mathbf{p}'_r$ and $\mathbf{p}'_s$ are projections of $\mathbf{p}_r$ and $\mathbf{p}_s$. Negative values can be absorbed into the data $\delta d_{s,r}$ by redefining polarizations in the negative direction (and having a negative source term), and it is usual to refer to polarizations consistently in (positive) $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ directions. Since we have neglected amplitude affects, we also assume $\mathbf{p}_r'^T(\partial\mathscr{M}/\partial m)\mathbf{p}_s' \leq 1$. These assumptions makes it clear that the complex component of $\mathbf{k}$ is a lossy term which varies with the two angles φ_s and φ_r. We can consider the resolution at $\mathbf{z}$ to be contributed to by only the measurements which have

$$\big(\mathbf{p}_r - \cos\varphi_r\hat{\mathbf{r}}\big)^T \frac{\partial\mathscr{M}}{\partial m}\big(\mathbf{p}_s - \cos\varphi_s\hat{\mathbf{s}}\big) \geq a > 0, \tag{7.4.17}$$

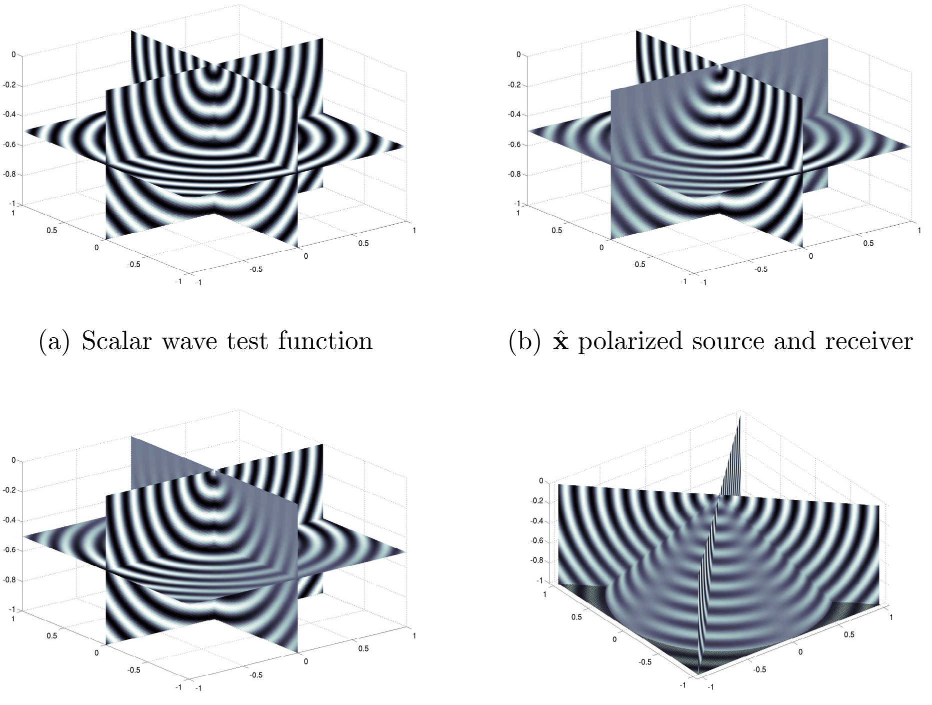
where a is some threshold determined by measurement accuracy.

For linear waves, (7.4.5) shows that frequency and aperture have redundant control of wavenumber coverage. Considering the complex wavenumber (7.4.16) for vector valued waves, this coverage is no longer completely redundant since $\mathbf{k}$ lives in the complex space $\mathbf{k} \in \mathbb{C}^3, \Im\{\mathbf{k}\} \leq 0$, which has twice the dimension as that of the scalar valued wave.

The terms multiplying δd in (7.4.4) and (7.4.15) are test functions which map data residuals to a perturbation in the model. This is similar to the SVD analysis carried out in Chapter 5, in which the singular vectors (above the noise level) are the test functions available to map the image space. In both cases, we need a set of measurements corresponding to test functions which well represent (resolve) the perturbations $\delta m(\mathbf{x})$ we wish to be able to image.

Figure 7.1 shows the test functions for co-located source and receiver at the origin, at 1 GHz in a subsurface with $\epsilon_r = 4$. Figures 7.1(a), 7.1(b), 7.1(c) and 7.1(d) are for the scalar wave case, $\hat{\mathbf{x}}\hat{\mathbf{x}}$, $\hat{\mathbf{y}}\hat{\mathbf{y}}$ and $\hat{\mathbf{x}}\hat{\mathbf{y}}$ polarized source and receiver, respectively. Figure 7.2 shows the same but for a source and receiver offset 50 cm in the $\hat{\mathbf{x}}$ direction. We see that the same-polarized measurements are of the form of concentric ellipsoids, with amplitude zero in the direction of polarization and 1 in the plane tangential to polarization. The cross-polar test functions similarly take nulls in the directions of both source and receiver polarization, and have a maximal amplitude of 0.5. The concentric ellipsoids for the cross-polar test functions however change phase over each plane of nulls: they are in the form of a quadrupole, whereas the single polarized test functions are in the form of a dipole.

Clearly, polarization plays a role in how well we are able to resolve targets. With a single polarization $\mathbf{p}$ for sources and receivers, we will see similar nulls in each test



(a) Scalar wave test function

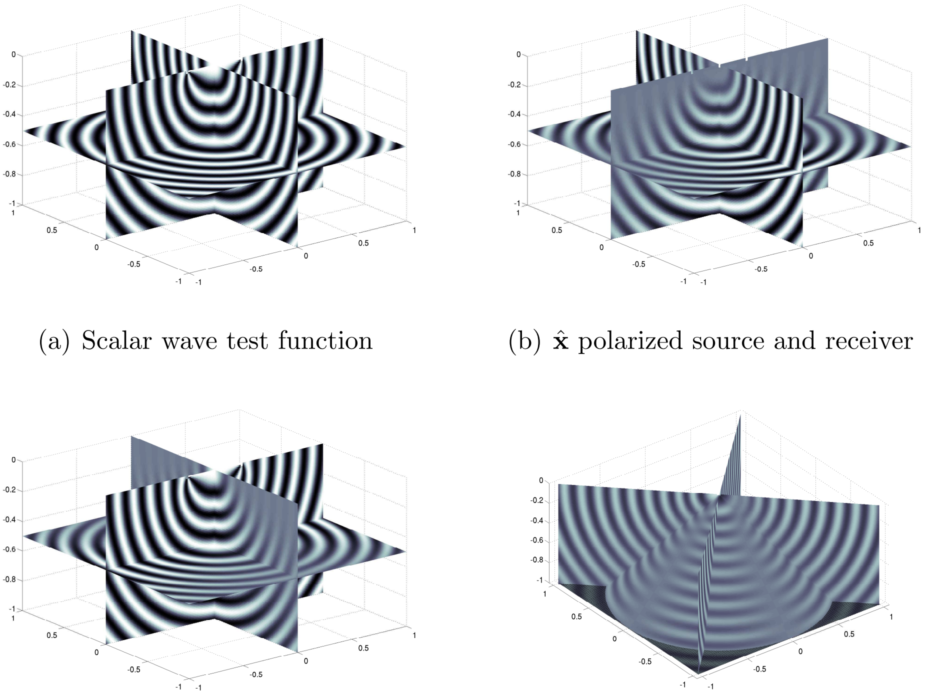
(b) $\hat{\mathbf{x}}$ polarized source and receiver

(c) $\hat{\mathbf{y}}$ polarized source and receiver

(d) $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ polarized source receiver

Figure 7.1: Test functions in the imaging space for co-located source and receiver at the origin, at 1 GHz in a background of permittivity $\epsilon_r = 4$

function and may be unable to resolve targets oriented in the plane tangential to **p**. Measuring both sets of polarizations mitigates this, and may allow targets to be pinpointed more accurately. Cross polarized measurements add an entirely additional class of test functions (quadrupoles), and so we may expect a far greater ability to resolve targets where all 3 types are used. This both gives some theoretical basis to the common experimental result that cross polar measurements give 'much more' information (for some informal definition of more), for example [103, 142, 201]. It also goes some way to explaining why sub-wavelength resolution can be obtained with cross-polar measurements in the (more general) inverse problem of electromagnetic wave scattering, for example Godavarthi et al [61].



(a) Scalar wave test function

(b) $\hat{\mathbf{x}}$ polarized source and receiver

(c) $\hat{\mathbf{y}}$ polarized source and receiver

(d) $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ polarized source receiver

Figure 7.2: Test functions in the imaging space for a 50 cm source-receiver offset in the $\hat{\mathbf{x}}$ direction, at 1 GHz in a background of permittivity $\epsilon_r = 4$

7.5 Maximising Sentitivity to a Mine-like Object

7.5.1 Introduction

In Chapter 5, we presented an argument for the use of hand-held (i.e. small-scale) multi-static arrays for landmine detection – both as a necessity for full-wave inversion, but also to improve information even for a linear imaging method. This was based on an SVD analysis of the Jacobian matrix, and the method could be extended to optimise the offsets of GPR antennas for the response of a target(s) of interest. Calculating the Jacobian matrix is computationally expensive, and an optimisation scheme would necessarily involve calculation of many Jacobians – it would therefore be prohibitive to optimise an array in this way.

Since we have an understanding of the polarization tensor being related to sensitivity in the data to a small object, we can use it instead of the Jacobian in numerical experiments to optimise GPR equipment for suitability for landmine detection. The computational cost is far less computationally expensive, and produces results which are applicable in a similar single-scattering regime.

In this section, we consider optimising antenna position of a linear array to determine the optimum offsets (within some sensible bounds) to maximise the response of a landmine. These are the same form of small-scale multi-static array as we advocated were necessary for inversion (and distinguishing objects) in 2D. With these results, we can consider critically if the optimal found antenna layout for a 2D array is sufficient.

7.5.2 Optimising arrays to maximise landmine distinguishability

We are concerned that a GPR antenna array should always be able to detect a landmine, down to the required depth in the soil, regardless of its orientation. We also require an array to gain as much information as possible about the object for deminers to know that they are dealing with a landmine.

As an object rotates, so too does its polarization tensor (7.2.13). By requiring that a GPR array should always be able to detect a mine-like object at a given depth z, we may wish to maximise the semi-norm objective function

$$\mathcal{J} = \min_{\phi_i} \|\mathbf{d}_{\phi_i}\|_\infty, \tag{7.5.1}$$

where $\phi_i = [\varphi_i, \theta_i]$ is polar and azimuthal rotation from flat of a mine-like object (and its polarization tensor) in spherical polar coordinates, $\mathbf{d}_{\phi_i}$ is the approximation of the data recorded for the isolated mine-like object in a homogeneous ground, oriented at $[\varphi_i, \theta_i]$. We have chosen the l_∞ norm as if one data-point is sufficiently strong we can say that the object has been detected, regardless of whether other data-points are much smaller. The minimum value over all angles is then taken as we are concerned with the worst case scenario – landmines continue to function and present a threat at all orientations, and still need to be removed.

We wish to perform full-wave inversion to resolve the subsurface in the l_2 norm. Requiring that an array gives as much information about a mine-like object as possible then, we may wish to maximise the semi-norm objective function

$$\mathcal{J} = \min_{\phi_i} \|\mathbf{d}_{\phi_i}\|_2^2. \tag{7.5.2}$$

Here, we are maximising the worst case (over orientation) 2-norm of the dataset, which we assume results in more information suitable for 2-norm inversion of the dataset.

Both semi-norms (7.5.1) and (7.5.2) are non-smooth at points where the orientation ϕ_i for the worst data set would change. They also may not be convex, and so all that can be concluded when optimising with such an objective function is that a locally optimal case has been found. This is still useful when considering the design of a small-scale GPR array, which will necessarily have to fit within given (possibly very restrictive) physical bounds.

7.5.3 Numerical experiment

We consider here optimisation of similar arrays as those we considered in Chapter 5: a multi-static array with antennas arranged in a line, with the 1 source to the side of 3 receivers. The optimisation was bounded by a minimum antenna offset of 2 cm, and a maximum source-receiver offset of 30 cm, and carried out to maximise the response of a mine-like target. This was taken to be the polarization tensor of an 8 × 4 cm oblate spheroid with $\epsilon_r = 3.25 - 0.018i$ at 2GHz, which is a similar dimension to a Type-72 landmine (a TNT filled mine) and approximately the permittivity and conductivity of TNT [51].

The optimisation was carried out using MATLAB's interior point method in `fmincon` function, which approximates gradients via finite differences. Table 7.1 shows the optimised array positions, the objective function values at the start initial and optimised configuration, and the first order optimality value. To ensure the results are consistent, the optimisation was restarted both from a different configuration as well as the optimised result.

	$\mathbf{x}_0$	$\mathbf{x}_s$	$f_p(\mathbf{x}_0)$	$f_p(\mathbf{x}_s)$	$\mathcal{O}(1)$ Optimality
$p = 2$	$[5.00, 10.0, 15.0]$	$[2.11, 4.48, 7.04]$	0.110	0.150	9.89×10^{-2}
	$[10.0, 20.0, 24.0]$	$[2.66, 4.86, 7.67]$	0.066	0.148	3.45×10^{-2}
	$[2.11, 4.48, 7.04]$	$[2.04, 4.26, 7.05]$	0.150	0.150	1.12×10^{-3}
$p = \infty$	$[5.00, 10.0, 15.0]$	$[3.28, 13.7, 23.9]$	0.0146	0.0174	6.56×10^{-4}
	$[10.0, 20.0, 24.0]$	$[2.70, 15.0, 25.4]$	0.0096	0.0178	1.04×10^{-1}
	$[3.28, 13.7, 23.9]$	$[3.11, 13.7, 24.4]$	0.0174	0.0174	1.12×10^{-3}

Table 7.1: Array optimisation results to maximise distinguishability of a mine-like object with a linear array. $\mathbf{x}_0$ is the starting configuration and $\mathbf{x}_s$ the optimised result. f_p is the $p = 2$ or ∞ semi-norm (7.5.1),(7.5.2).

Throughout the optimisation procedure (at all trial configurations), the least distinguishable orientation was always $[\varphi, \pi/2]$, i.e. the 'mine' oriented vertically, since this presents the smallest cross-section facing the antennas. Since this worst orientation never changed, the cost function was smooth within the range of values tested. The results also appear fairly robust to starting configurations, though there is some change in solution – likely due to the gradient being small in a wide region around the solution.

The $p = 2$ optimisation moves the receivers as close to $x = 0$ as possible, since this is where the maximum response of the worst-oriented target occurs. With the worst orientation being the same for all possible antenna arrangements, optimising with f_2 does not yield any illuminating results. Optimising for f_∞ though separates the receivers, presumably to each detect different peaks in response. I.e. the first receiver moves close to $x = 0$ which is where the maximum response for the most difficult to detect orientation is found, allowing the other two receivers to improve the response of the next worse orientations. This is aligned with the result of Sirgue and Pratt [159] that the wavenumber coverage in the domain is related to the source-receiver offset, with a larger maximal offset giving better coverage, as discussed in Sections 4.3.7 and 7.4.2.

This is a simple test example in which the results are not particularly illuminating with regards to how to arrange antennas for landmine detection. The principle could be useful though with more realistic problems or more complex design constraints, since it is relatively cheap to calculate the cost function as it does not require simulating more data. More importantly, we find only a small difference in distinguishability between the starting and optimal found array layouts: the limiting factor is not antenna position, but a lack of 2D array and cross-polar measurements. The mine-like object was always least distinguishable when rotated closer to $\pi/2$ about $\hat{\mathbf{x}}$, at which point the single-polarised antennas will receive the smallest reflection no matter where they are placed along the $\hat{\mathbf{x}}$ axis.

7.5.4 An SVD analysis

We can conduct an SVD analysis of a linear array using the polarisation tensor expansion, in a similar manner to the that in Chapter 5. To do so, we form the linear operators (7.3.27) for objects at a set of locations $x = -10, -5, 0, 5, 10,$, $y = -10, -5, 0, 5, 10$ and $z = 2.5, 5, 7.5, 10$ cm, concatenate these matrices and calculate a singular value decomposition of the map from tensors to data,

$$D\begin{bmatrix} D_1 & D_2 & \cdots & D_n \end{bmatrix} = U\Sigma V^T \tag{7.5.3}$$

where D_i is the linear operator for a polarization tensor at the i^{th} location. The linear array considered had $\hat{\mathbf{y}}$ polarised receivers at $x = 10, 20$ and $y = 0$ cm, 24 frequencies between 1 and 2 GHz, and 100 source locations over $[-10, 10] \times [-10, 10]$ cm, 10cm above the ground surface. The resulting linear operator D is an overdetermined system, with 4800 data points mapped to from 600 or 1200 tensor elements, for mapping permittivity alone or both permittivity and conductivity, respectively.

The singular values of D (for both permittivity and conductivity) are shown in Figure 7.3. The singular values decay as $\sigma_i \propto a^{-\gamma i}$, for some $a, \gamma > 0$, fitting the rapid decay of singular values generally seen inverse problems as discussed in Section 5.3. The rate of decay is greater than in the 2D Helmholtz case analysed in Chapter 5 (exponential vs polynomial), which we suppose is due to the vector valued nature of the problem.

We see a clear drop in singular value around $i = 1000$, corresponding to an effective drop in rank, and so have a definite nullspace. If we assume an accuracy threshold of $10^{-3}\sigma_1$, corresponding three significant figures measurement accuracy, then we find the singular vectors in the effective nullspace involve the tensor elements we expect: largely $\mathscr{M}_{11}$ and $\mathscr{M}_{13}$, as shown in Figure 7.4. This is because neither the source nor the measurements are polarised in $\hat{\mathbf{x}}$ or $\hat{\mathbf{z}}$, so these tensor components only come in to play when refraction at the ground surface changes polarization to have some $\hat{\mathbf{x}}$ or $\hat{\mathbf{z}}$ component, and then back to $\hat{\mathbf{y}}$ after reflection from a scatterer.

The nullspace not only gives us information about what a given acquisition system would not be able to detect, but also what it would be unable to tell apart – which is given by tensors for with their differences in the nullspace. For the acquisition system

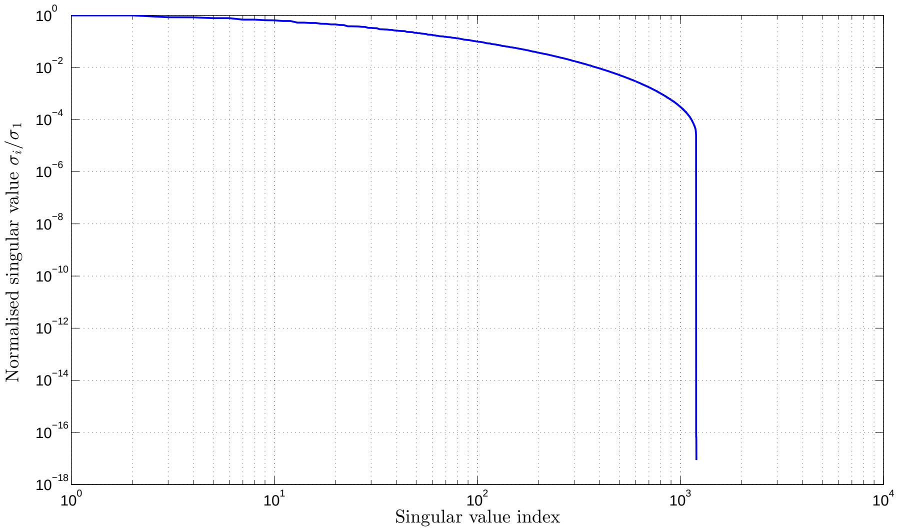


Figure 7.3: Singular values of the map from polarization tensor elements to data for a hand-held linear array.

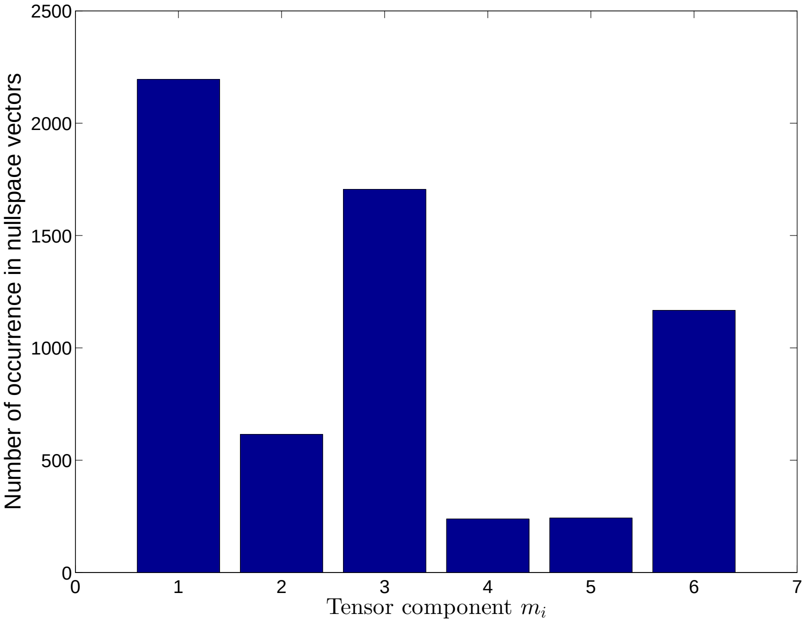


Figure 7.4: Occurrences of tensor elements in nullspace vectors. $m_1 = \mathscr{M}_{11}$, $m_2 = \mathscr{M}_{12}$, $m_3 = \mathscr{M}_{13}$, $m_4 = \mathscr{M}_{22}$, $m_5 = \mathscr{M}_{23}$ and $m_6 = \mathscr{M}_{66}$.

analysed, we find the difference between the pairs of tensors and locations

$$\begin{aligned}\mathscr{M}^{(1)}(B_1) &= \begin{bmatrix} 2.0000 & 0.1744 & -0.6700 \\ 0.1744 & 0.0284 & 0.0386 \\ -0.6700 & 0.0386 & 0.2932 \end{bmatrix} \quad \text{at } \mathbf{r}_1 = [0,0,5] \\ \mathscr{M}^{(2)}(B_2) &= \begin{bmatrix} 2.0000 & 0.0823 & -0.8818 \\ 0.0823 & -0.0317 & -0.0906 \\ -0.8818 & -0.0906 & 2.0418 \end{bmatrix} \quad \text{at } \mathbf{r}_2 = [0,2.5,5]\end{aligned} \tag{7.5.4}$$

lies in the nullspace, where $\mathscr{M}^{(1)}$ is corresponds to a spheroidal inclusion B_1 at $\mathbf{r}_1 = [0,0,5]$ cm and $\mathscr{M}^{(2)}$ corresponds to a spheroidal inclusion B_2 at $\mathbf{r}_2 = [0,2.5,5]$ cm. The acquisition system would not be able to distinguish between an object at $[0,0,5]$ with tensor $\mathscr{M}^{(1)}$ and an object at $[0,2.5,5]$ with tensor $\mathscr{M}^{(2)}$, or any two systems of scatterers which only differ by any scalar multiple of $\mathscr{M}^{(1)} - \mathscr{M}^{(2)}$.

One can easily understand from this example how such a linear acquisition system may be unable to accurately locate (and resolve) an object which is greater in $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$ dimension than in $\hat{\mathbf{y}}$, for any rotation about the $\hat{\mathbf{z}}$ axis in the subsurface. This is clear since the polarization tensors $\mathscr{M}^{(1)}$ and $\mathscr{M}^{(2)}$ have $\mathscr{M}_{11}$, $\mathscr{M}_{13}$ and $\mathscr{M}_{33}$ as the greatest magnitude components. This coincides with remarks by D.J. Daniels that a hand-held system may struggle to accurately locate mine-like targets at an angle, for example D.J. Daniels [44].

From this section, we ought to conclude that while a multi-static linear array is necessary for 2D FWI and target distinguishability, something more complex again is needed for the 3D FWI problem. We need at least some form of 2D array to be able to accurately locate and resolve targets of arbitrary orientation in 3D, and to improve resolution we ought also to have cross-polar measurements. Otherwise, we find a large nullspace containing objects we wish to resolve.

7.6 Summary and conclusions

In this chapter, we have shown that an asymptotic expansion for the scattering of electromagnetic waves by a small object can be re-written, by use of a surface equivalence theorem, in a way which may be applicable to ground-penetrating radar. While we have noted that such expansions have been used for efficient target location schemes, the lack of multiple scattering means that for our application (in which we may have a very inhomogeneous background) it is more useful to use the approximation to better understand sensitivity. This gave us a novel viewpoint of sensitivity which is comparable to the more traditional understanding involving the Fréchet derivative.

The two ideas of sensitivity are applicable under different criteria, one with a small perturbation in parameter contrast (e.g. permittivity) but of any dimension and shape, and the other for any sized perturbation in parameter but of small dimension. Both are applicable when considering a small change in parameter of small dimension, and so the two can coincide. This idea of sensitivity was used to extend the resolution analysis in FWI of scalar valued waves based on the Born approximation, so that the polarization of vector-valued waves could be included in the inverse problem with full the Maxwell equations.

Finally, we have used the polarization tensor expansion to analyse the performance of a linear array of single polarised antennas – the same form of system as shown to be necessary for 2D inversion in Chapter 5. We have seen that even an optimal linear array may behave poorly if we wish to accurately locate and resolve a mine-like object which is not sitting flat. For the full 3D inversion problem, it will not be sufficient to use a single polarised linear array.

Chapter 8

Full-Wave Inversion in 3D

8.1 Introduction

There is now a substantial amount of research on full-wave inversion of GPR data in 2D (see Chapter 2 for an overview), and we have also had positive results with the 2D inversions carried out in Chapter 4. Unfortunately, 2D imaging assumes a particularly inaccurate property of the subsurface: that it is invariant in one coordinate axis. This may be acceptable for certain applications, such as the detection of large layers or objects such as pipes running in the direction assumed to be invariant. For objects which vary in 3 dimensions, the 2D assumption necessarily misinterprets their shapes, and misplaces out-of-plane reflections. Unfortunately, landmines are not 2D objects, so we must attempt to resolve in 3D. Particularly, what we wish to gain from FWI in 3D is to safely rule out some detected metal targets as not being landmines.

The computational cost of full 3D FWI of GPR data has previously been prohibitive for any significant results, primarily due to the cost of simulating data. For many applications of GPR a large domain is required for useful results, since often a picture of the subsurface as a whole is desired. Depending on the scale of feature one wishes to resolve for, a domain may be 10s or 100s of wavelengths long in each of the 3 co-ordinate directions. Numerical simulation of electromagnetic waves requires many grid points per wavelength, and so the resulting linear system will involve hundreds of thousands, if not millions, of unknowns. Such a system may either be unsolvable within given memory limitations, or may take a prohibitively large amount of time (especially when we consider we will need to simulate data many 1000s of times to solve the FWI problem).

Solving the inverse problem itself is also more complex, since 3D scattering of transverse waves is inherently more complex than the scattering of 2D longitudinal waves. We also have many more subsurface parameters in our model to fit to our data, for which simulated data varies non-linearly with each. Reconstruction in 3D also requires a third independent dimension of measurements (a second spacial dimension $\hat{\mathbf{y}}$ on the ground surface, to go with the first $\hat{\mathbf{x}}$ and time/frequency used in 2D imaging). Not only does this make the dataset larger (increasing storage cost), but there is more data to simulate, increasing the computational cost further. Solving the 3D inverse problem then has a much greater storage cost, is more complex due to the larger number of parameters to be fitted which all cause a non-linear variation, and involves a greater number of solutions to the forward problem which is also far more costly to compute than a 2D simulation. For the landmine detection problem however we only need to invert in a small area around a detected target, rather than the entire region currently being explored. In other words, we do not need to use FWI to generate an image of the entire lane being explored in order to detect landmines, only as a verification for targets already detected. We can then invert on a domain just small enough that 3D FWI becomes plausible.

In this chapter, we explore the use of 3D FWI to gain quantitative information about detected targets, to help reduce the rate of false positives in landmine detection. We present the first full 3D FWI results for simulated GPR data, and use a combined FE-BI forward solver which is novel to GPR FWI. We begin by covering some implementation aspects which are different to the 2D reconstruction problem using a finite-difference grid, before presenting initial numerical results. We then introduce the concept of nuisance parameters, one familiar to inverse problems literature in general but not yet applied to the GPR problem. Based on this concept, we present a new algorithm which separates subsurface parameters of interest (those around a detected target) to those which are not (those away from the target). The solution to the inverse problem varies non-linearly with the nuisance parameters in this case, so they cannot be defined implicitly as functions of the target parameters in a straightforward way. We do though treat the two sets of parameters separately in a novel 'line search' algorithm, enabling us to fit well the target parameters to the data, but only loosely fit those away from the target for which we have worse data coverage.

8.2 Implementation aspects

8.2.1 Regularisation

In 2D the construction of regularisation operators was straightforward on the evenly spaced finite-difference grid, but we must reconsider their construction on an unstructured tetrahedral mesh in 3D. Since the volumes of the elements can vary, we modify Tikhonov regularisation o

$$R(\mathbf{m}) = \|V(\mathbf{m} - \mathbf{m}_{\text{prior}})\|_2^2 := \|\mathbf{m} - \mathbf{m}_{\text{prior}}\|_{V,2}^2, \tag{8.2.1}$$

where V is a diagonal matrix of the volumes of each tetrahedron. This way, a small element does not have a disproportionate affect on the regularisation cost.

The discrete Laplace operator L defined on tetrahedral volumes is given by

$$L_{ij} = \begin{cases} \dfrac{s_i}{\tilde{s}} & \text{if } i = j \\ -\dfrac{s_{ij}}{\tilde{s}} & \text{if } \triangle_j \text{ adjacent to } \triangle_i \\ 0 & \text{otherwise} \end{cases} \tag{8.2.2}$$

where s_{ij} is the surface area of the face between elements $\triangle_i$ and $\triangle_j$, s_i is the total surface area of element $\triangle_i$, and $\tilde{s}$ the mean surface area of all the triangular faces. Note that if all the elements are equilateral this reduces to the usual finite difference operator (for tetrahedral meshes), which has 4 on the diagonal and -1 where $\triangle_i$ is adjacent to $\triangle_j$. The definition (8.2.2) in terms of surface areas also holds for arbitrary shaped polygonal meshes (e.g. cubic).

Similarly, the discrete grad/difference operator (for Total Variation) is given by

$$G_{ij} = \begin{cases} +\dfrac{s_i}{\tilde{s}} & \text{if } \triangle_j \text{ contains face } i \text{ and is before the other element containing } i \\ -\dfrac{s_i}{\tilde{s}} & \text{if } \triangle_j \text{ contains face } i \text{ and is after the other element containing } i \\ 0 & \text{otherwise} \end{cases}, \tag{8.2.3}$$

where s_i is the area if face i.

8.2.2 Gradient calculation

The calculation of the gradient via the adjoint-state method largely follows that of the finite-difference approximation of the Helmholtz equation, but differs slightly since our measurements are no longer taken to be the values of the wave field at the grid points on the surface. Using a coupled FE-BI forward model, our measurements are calculated by a linear combination of the $\mathbf{M}_s$ and $\mathbf{J}_s$ coefficients. The single-frequency single-source measurements $\mathbf{d}^{\text{sim}} \in \mathbb{C}^d$ are given by

$$\mathbf{d}^{\text{sim}} = \mathcal{F}[\mathbf{m}] = RS(\mathbf{m})^{-1}\phi, \tag{8.2.4}$$

where $\mathcal{F}$ is the forward operator, $\mathbf{m} \in \mathbb{R}^p$ is the discretisation of permittivity, conductivity and permeability (if applicable), $S = [AB; CD] \in \mathbb{C}^{n\times n}$ is the FE-BI system matrix and ϕ the exterior source term. The sparse matrix $R \in \mathbb{C}^{n\times d}$ is the discrete numerical integration operator of $\mathbf{M}_s$ and $\mathbf{J}_s$.

To form the adjoint-state calculation of the gradient, we follow again the derivation given by Pratt et al [137]. First, consider an element of the Jacobian matrix,

$$J_{ij} = \frac{\partial \mathcal{F}_i}{\partial m_j} = \frac{\partial}{\partial m_j}\left(RS^{-1}\phi\right)_i = \left(R\frac{\partial S^{-1}}{\partial m_j}\phi\right)_i = R_i\frac{\partial S^{-1}}{\partial m_j}\phi. \tag{8.2.5}$$

Here, R_i represents the i$^{\text{th}}$ row of R, i.e. the row which gives the i$^{\text{th}}$ measurement.

Differentiating the forward operator (8.2.4) with respect to the parameter m_j

$$S\mathbf{E}_h = \phi, \tag{8.2.6}$$

resulting in

$$\frac{\partial S}{\partial m_j}\mathbf{E}_h + S\frac{\partial \mathbf{E}_h}{\partial m_j} = \mathbf{0}, \tag{8.2.7}$$

which we can write as

$$\frac{\partial \mathbf{E}_h}{\partial m_j} = S^{-1}\mathbf{f}^{(i)}, \qquad \mathbf{f}^{(i)} = -\frac{\partial S}{\partial m_i}\mathbf{E}_h. \tag{8.2.8}$$

We have used $\mathbf{E}_h$ to denote the finite element coefficients of both the internal $\mathbf{E}$ field and surface $\mathbf{H}$ field. Note that the Jacobian matrix can be written as

$$J = R\frac{\partial \mathbf{E}_h}{\partial \mathbf{m}} = RS^{-1}[\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(p)}] = RS^{-1}F, \tag{8.2.9}$$

where p is the total number of material parameters. Therefore,

$$\begin{aligned}\nabla_{\mathbf{m}}\mathcal{J} =& \Re\left\{(RS^{-1}F)^T\delta\mathbf{d}^*\right\}\\ =& \Re\left\{F^T(S^{-1})^T R^T\delta\mathbf{d}^*\right\}.\end{aligned} \tag{8.2.10}$$

Unlike in the Helmholtz case, we can be sure that S is a non-symmetric matrix, and so cannot take $(S^T)^{-1}$ to be S^{-1}. However, if we are using and storing an LU decomposition of S, these can still be used to solve the adjoint problem.

Note that the surface integral parts of matrix S, i.e. B, C and D, are not functions of $\mathbf{m}$, so only the mass and stiffness matrices will contribute to $\partial S/\partial m_j$. Differentiating with respect to a permittivity parameter gives a contribution from the mass matrix only,

$$\begin{aligned}\frac{\partial S}{\partial \epsilon_j} &= \begin{bmatrix} \frac{\partial M}{\partial \epsilon_j} & 0 \\ 0 & 0 \end{bmatrix} \\ \frac{\partial M_{mn}}{\partial \epsilon_j} &= \begin{cases} -k_0^2 \displaystyle\int_{\Box_j} \mathbf{L}_m \cdot \mathbf{L}_n \, dv, & \mathbf{L}_m, \mathbf{L}_n \text{ supported in } \Box_j \\ 0 & \text{elsewhere} \end{cases} .\end{aligned} \tag{8.2.11}$$

We have used the notation $\Box_j$ to denote the one or more tetrahedral elements $\triangle_i$ that form the volume of parameter m_j.

Similarly, one can differentiate with respect to a conductivity parameter to give

$$\begin{aligned}\frac{\partial S}{\partial \sigma_j} &= \begin{bmatrix} \frac{\partial M}{\partial \sigma_j} & 0 \\ 0 & 0 \end{bmatrix} \\ \frac{\partial M_{mn}}{\partial \sigma_j} &= \begin{cases} \mathrm{i}k_0 Z_0 \displaystyle\int_{\Box_j} \mathbf{L}_m \cdot \mathbf{L}_n \, dv, & \mathbf{L}_m, \mathbf{L}_n \text{ supported in } \Box_j \\ 0 & \text{elsewhere} \end{cases} ,\end{aligned} \tag{8.2.12}$$

as well as by a permeability parameter,

$$\begin{aligned}\frac{\partial S}{\partial \mu_j} &= \begin{bmatrix} \frac{\partial K}{\partial \mu_j} & 0 \\ 0 & 0 \end{bmatrix} \\ \frac{\partial K_{mn}}{\partial \mu_j} &= \begin{cases} -(\mu_r)^{-2} \displaystyle\int_{\triangle_j} (\nabla \times \mathbf{L}_m)(\nabla \times \cdot \mathbf{L}_n) \, dv, & \mathbf{L}_m, \mathbf{L}_n \text{ supported in } \Box_j \\ 0 & \text{elsewhere} \end{cases} .\end{aligned} \tag{8.2.13}$$

Ordering $\mathbf{m}$ such that $m_j = \epsilon_j$ for $j \in [1, p_1]$, $m_j = \sigma_{j-p_1}$ for $j \in [p_1 + 1, p_2]$, and $m_j = \mu_{j-p_1-p_2}$ for $j \in [p_1 + p_2 + 1, p]$ (if applicable), we have

$$\frac{\partial A}{\partial m_j} = \begin{cases} \partial M/\partial \epsilon_j & j \in [1, p_1] \\ \partial M/\partial \sigma_{j-p_1} & j \in [p_1 + 1, p_2] \\ \partial K/\partial \mu_{j-p_1-p_2} & j \in [p_1 + p_2 + 1, p] \end{cases} \tag{8.2.14}$$

and so

$$\frac{\partial S}{\partial m_j} = \begin{bmatrix} \frac{\partial A}{\partial m_j} & 0 \\ 0 & 0 \end{bmatrix}. \tag{8.2.15}$$

We have allowed a different discretisation for permittivity, conductivity and permeability, and so could have a finer discretisation for permittivity than for conductivity (say).

8.3 Numerical experiments

8.3.1 Introduction

In this section we present the first 3D FWI results for GPR imaging. Our aim is to show that it is possible to use FWI in 3D to gain both quantitative and qualitative information about detected objects, with the additional information being useful to determine if a target is or is not a landmine.

We again restrict ourselves to simulated data from hand-held equipment, for which we place the (artificial) limit of 8 receivers in an array no larger than 30 cm^2, though demining equipment may need to be smaller. Following our results in Chapter 5 we consider only multi-static arrays, for which in 3D we consider the simplest case to be a 2×2 array. Following also the reasoning in Chapter 7, we always consider systems which have cross polar measurements.

Ideally, similar analysis to that carried out in Chapter 5 would be carried out for 3D FWI, to determine what form of acquisition was required and to characterise the nullspace. Unfortunately such analysis is not currently sensible for the 3D electromagnetic inverse problem, due to both time and memory limitations in calculating the Jacobian matrix. These results are presented then as a proof of principle for small scale 3D FWI of GPR data. We begin with simple reconstructions to show that one can find the shape, size and permittivity of a single inclusion using data with simulated noise (i.e. determine what a target is), before considering domains which contain a highly variable background resulting in multiple scattering. Parameters used for the reconstructions throughtout the rest of this thesis are given in Table 8.1, unless otherwise stated.

Linesearch	type	strong Wolfe
	c_1	10^{-4}
	c_2	0.8
	δ (safeguard)	10^{-5}
	max iterations	5
Regularisation	type	TV
	λ	5
	ε (initial)	10^{-6}
	ε decrease	0.8×
Source	function	$\overline{\overline{\mathbf{G}}}\mathbf{p}$ within beam
	beam width	$\pi/8$ radians
	rolloff start	$\pi/12$ radians
	rolloff function	cubic
PML	width	0.2 m
	s''_x, s''_y, s''_z (maximum)	1
	s''_i increase distance	0.02 m
	s''_i increase function	quadratic
Miscellaneous	boundary integral	TETH CFIE
	mesh max edge length	0.0075 m
	frequency weighting	$1/\omega_i^2$

Table 8.1: Parameters used for 3D inversion. Note that the stopping criteria are not given, as in all numerical experiments inversion was run for a maximum number of iterations.

8.3.2 Recovering a single inclusion

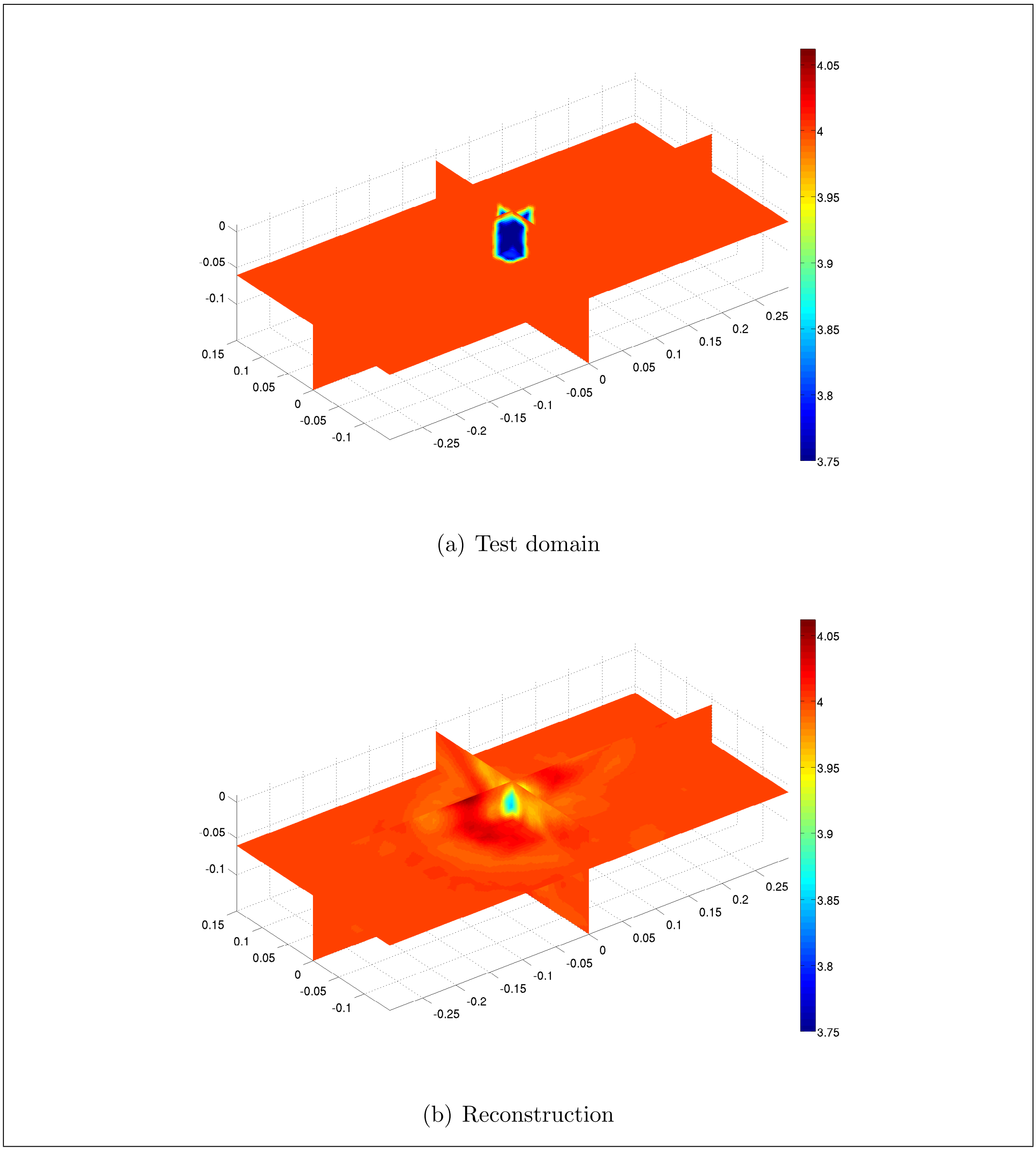
We begin by simulating data for a 4 cm cube of relative permittivity $\epsilon_r = 3.75$, buried at a depth of 2 cm (to the top) in a background with constant $\epsilon_r = 4$, as shown in Figure 8.1(a). Note that the object does not have a smooth boundary: this is an artefact of MATLAB's `slice` image function. The GPR system is a 2×2 array of antennas arranged in a 15×15 cm square, each measuring both polarisations. Measurements are taken every 1 cm in the $\hat{\mathbf{x}}$ direction and every 2 cm in the $\hat{\mathbf{y}}$ direction (simulating a single zig-zagging pass over the object), from -10 to $+10$ cm relative to the centre of the object in each direction. Data is simulated at 12 frequencies between 750 MHz and 1.5 GHz, all together resulting in a data set which is smaller than a realistic (increasing the difficulty of inversion). 1% Gaussian white noise is added to both the real and imaginary parts of the data, and the inversion is carried out using a different coarser mesh for the forward operator. We have been less ambitious in the amount of noise added for this initial test than for the 2D FWI results in Chapter 4, due to having less data redundancy to save computation time.

The reconstructed image after 10 iterations of l-BFGS, using Laplace regularisation (8.2.2) and $\lambda = 1.5$, is shown in Figures 8.1(b) and 8.2. In the latter, we see that we are well able to determine the approximate height and permittivity of the object, but have underestimated the contrast to the background at this point and are less able to determine the object's width. We would not expect to exactly reconstruct the object after only 10 iterations (which is not enough for even a 2D reconstruction), and would likely underestimate the contrast regardless due to the regularisation term. Nevertheless, we have a stable reconstruction in the presence of (some small amount of) noise for a single target, using Maxwell's equations in 3D as the forward model and a data simulated for a small hand-held array of one-sided measurements: this is in itself a significant step towards our goal.

The generalised Tikhonov regularisation using the discretised Laplace operator results in a smooth reconstructed boundary, not the jump change actually present, and allows some smooth wave-like artefacts surrounding the object. These artefacts are present in the descent direction. Similar artefacts were present in early iterations of 2D reconstructions (see for example Figure 4.3), but the TV regularisation used was better able to compensate for them as they have a large total variation. The inability of Laplace regularisation to counter these smooth artefacts, and since the object we are trying to reconstruct does *not* have a smooth boundary, make it an inappropriate choice. However, it is useful to use a linear type of regularisation (such as a generalised Tikhonov function) for prototyping and testing the reconstruction algorithm in 3D, since a descent direction can be found via linear algebra; non-linear regularisation such as TV cannot be solved via linear algebra alone. The amount of regularisation was chosen as in our experience it was sufficient to result in a stable reconstruction in the presence of noise. A more thorough study of the amount of regularisation required for the 3D problem is still required.

8.3.3 Eigenvalue decomposition of the approximate Hessian

As noted by Thacker [168], and considered by Ma and Hale [105] for the FWI problem (see also [104]), the eigenvectors corresponding to the largest and smallest eigenvalues



(a) Test domain

(b) Reconstruction

Figure 8.1: 3D reconstruction of a single cubic scattering object in a homogeneous background after 10 iterations of l-BFGS, with Laplace regularisation and $\lambda = 1.5$.

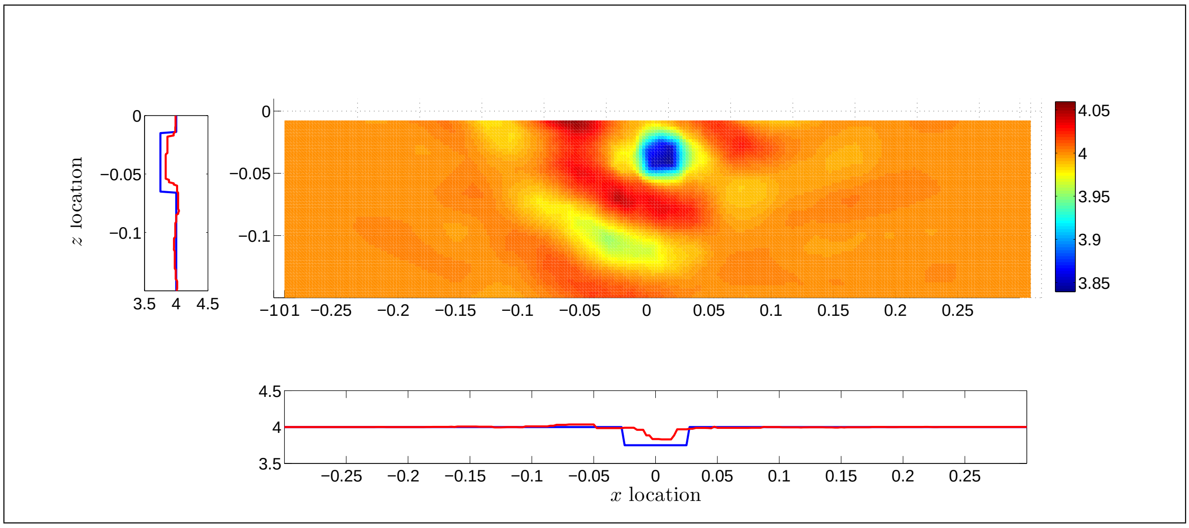


Figure 8.2: Slices of a reconstruction of a single cubic scattering object in a homogeneous background after 10 iterations of l-BFGS, with Laplace regularisation and $\lambda = 1.5$. In the horizontal and vertical line slices, the blue line shows the true permittivity and the red line shows reconstructed. The horizontal slice was taken at $y = 0$, $z = 5$ cm, and the vertical slice at $x = 0$, $y = 0$.

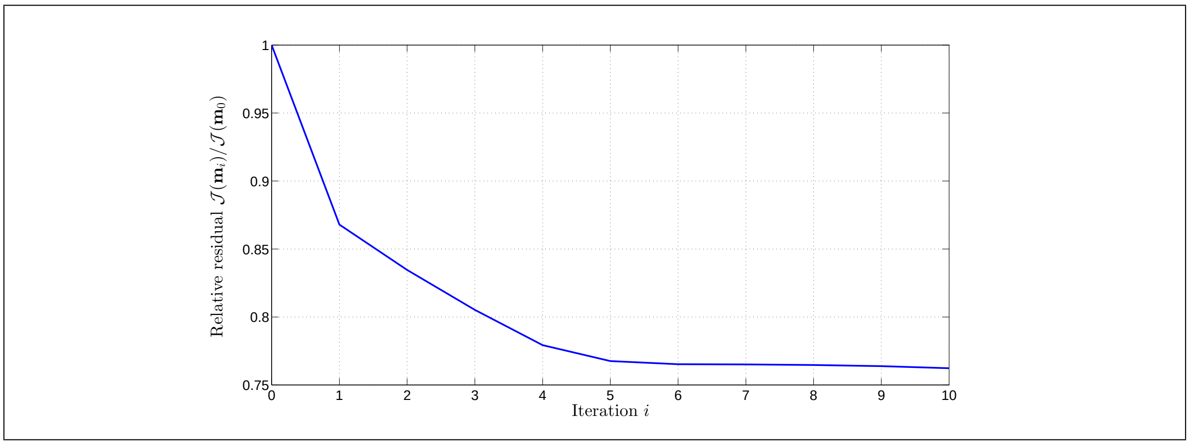


Figure 8.3: Relative residuals of the resconstruction of a single cubic object .

of the Hessian matrix correspond to the parts of the model $\mathbf{m}$ which can be best and worst determined, respectively. This is since these correspond to the directions in which $\mathcal{J}$ changes either the greatest, for the largest eigenvalues, or least, for the smallest eigenvalues. Figure 8.5 shows the eigenvectors corresponding to the largest and smallest eigenvalues after 10 iterations.

Figure 8.4 also shows the unique eigenvalues calculated by MATLAB's `eigs` function. Both the smallest and largest 30 eigenvalues were calculated, and since these overlapped we have all numerically unique eigenvalues of the l-BFGS approximate Hessian. We see from the eigenvectors of the Hessian, Figure 8.4, that the reconstruction is well-resolved in the direction of eigenvector corresponding to the largest magnitude eigenvalue. The eigenvector corresponding to the smallest eigenvalue has contributions mostly on the below and to the left of the target location – the area in the shadow of the source antenna – as well as the top-right corner of the target. This is because these are the parts of the target for which there will be the least direct reflections from in the data.

Moreover, Thacker demonstrates that in the linear approximation the inverse Hessian matrix is the covariance matrix [168]. For any linear function of the model parameters $y = \mathbf{b}^T\mathbf{m}$, the variance of y can be estimated by inverse Hessian as

$$\mathrm{var}(y) \approx \mathbf{b}^T B \mathbf{b}, \tag{8.3.1}$$

where B is the l-BFGS approximate inverse Hessian. The variance of each parameter solution are then approximated simply by the diagonal components of the approximate inverse Hessian, since $\mathbf{b}_i = \hat{\mathbf{e}}_i$. Figure 8.6 shows the variances for the inversion result in Figure 8.1(b). Note though that our approximate Hessian, containing only a few eigenvalues of the true Hessian, may be inaccurate, and that the linear approximation may not hold.

8.3.4 Discussion

We have presented here the first (published) 3D FWI results for GPR. The 3D inverse problem is not generally considered due to its large computational cost: to gain meaningful results for most applications, one would require a computational domain beyond memory limits. Not only is the 3D inverse problem computationally expensive,

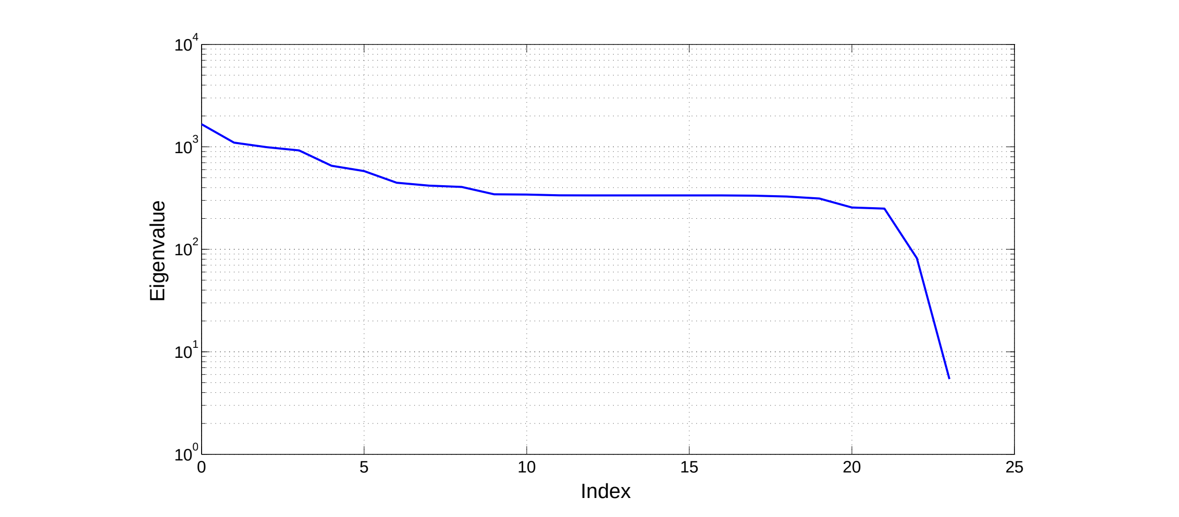
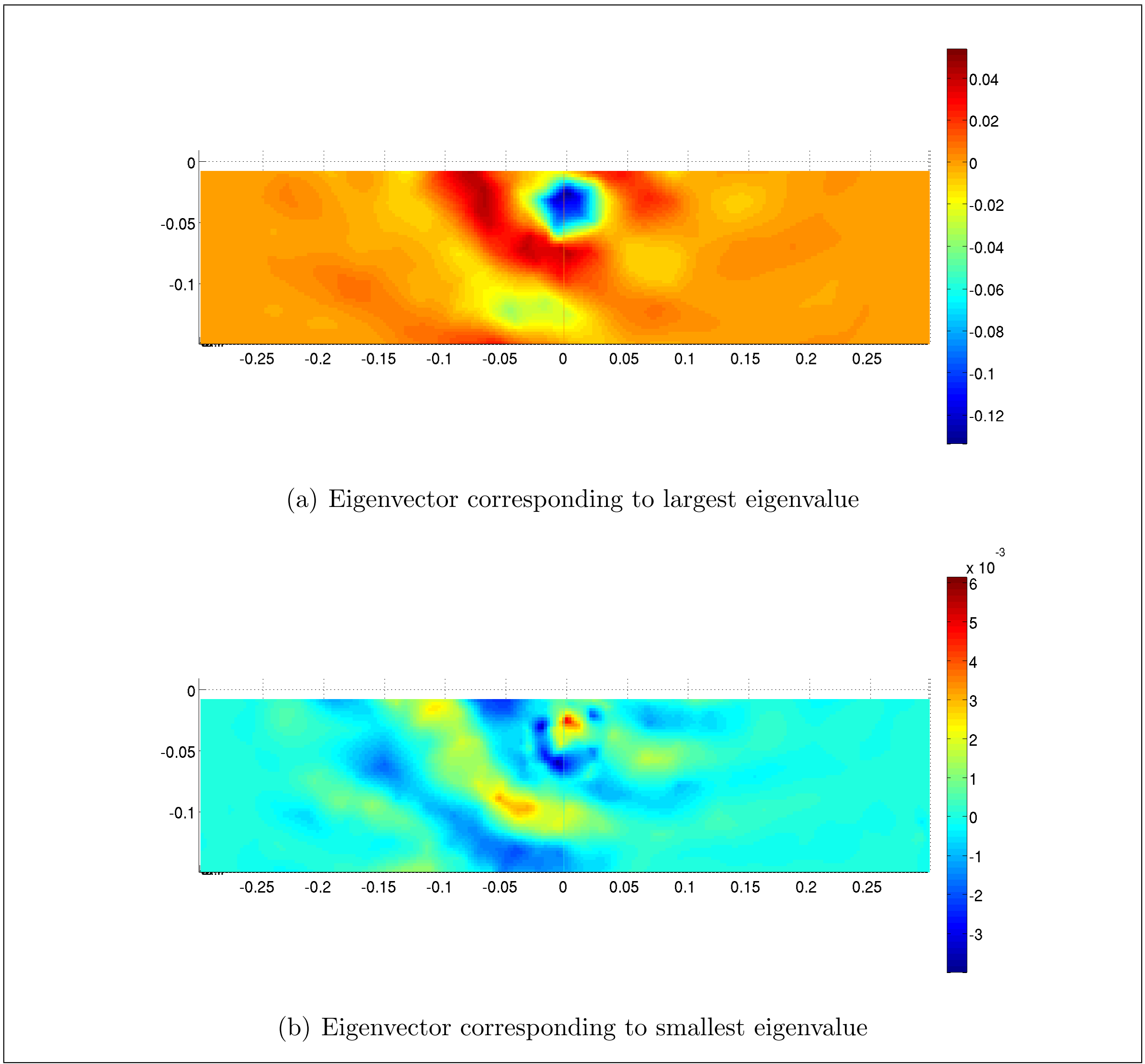


Figure 8.4: Eigenvalues of the l-BFGS approximate Hessian after 20 iterations for a coarse reconstruction of a single inclusion



(a) Eigenvector corresponding to largest eigenvalue

(b) Eigenvector corresponding to smallest eigenvalue

Figure 8.5: Eigenvectors of the l-BFGS approximate Hessian after 10 iterations, corresponding to (a) the largest and (b) the smallest eigenvalues. Slices at $y = 0$.

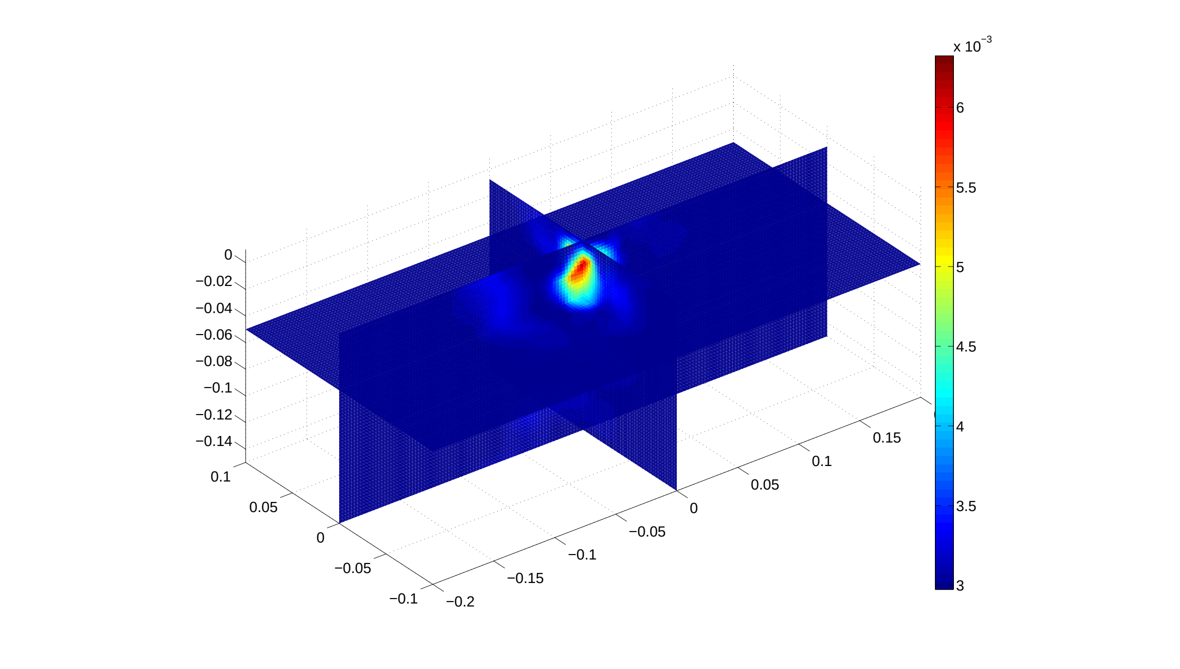


Figure 8.6: Estimation of the variance of the 3D reconstruction of a single target in homogeneous background, using the l-BFGS approximate Hessian

but the underlying scattering problem is also more complex. For 2D FWI, one of the assumptions is that there is no out of plane scattering in the data, but even in this simple single target example out of plane scattering will be a significant feature in the data almost everywhere. Polarization effects and refraction at the ground surface also contribute to the complexity of the scattering problem, and therefore of the inverse problem.

One of the features the more complex scattering gives rise to is a ‘shadow’ region on the far side of the scatterer to the antennas. We noted in Section 8.3.2 this feature was also seen in early iterations of the 2D inverse problem, but it was easier to account for in 2D as we had better parameter illumination over each all possible ray angles (1 dimension of data). In 3D, each parameter can be illuminated over 2 sets of angles (2 dimensions of data) and from 2 possible source polarizations. Ideally, each parameter would be illuminated from all possible angles and all polarizations so that the position and orientation of surfaces can be properly determined. Any small array which has a single source antenna will therefore be more limited as to seeing the opposite sides of objects, as highlighted by the eigenvector analysis in the previous section.

The poor descent direction beneath and around the target becomes far more prevalent in more realistic examples. Figure 8.7 shows a slice of the gradient of the objective

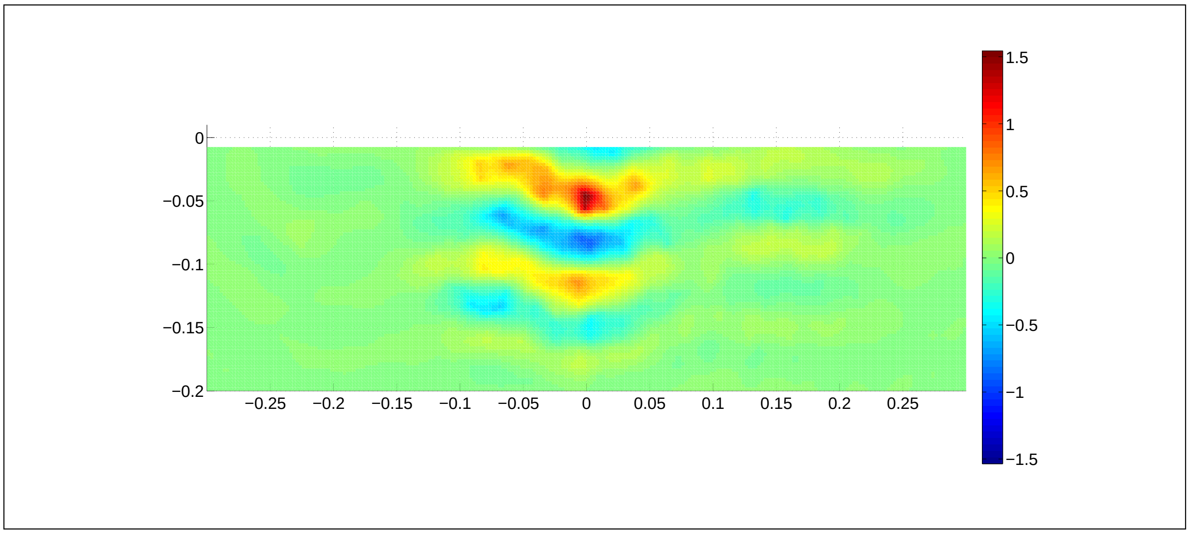


Figure 8.7: Gradient of the objective function $\nabla\mathcal{J}(\mathbf{m}_0)$ for a single target in stochastic background medium, as shown in Figure 8.8, for a homogeneous domain $\mathbf{m}_0$. Slice at $y = 0$.

function for a single target in a more realistic stochastic background medium (see later Figure 8.8 for the true domain). The more complex scattering, including out of plane scattering, multiples, and misplaced perturbations due to an incorrect background permittivity estimation and curving of ray paths in the true medium, results in a descent direction with many incorrectly placed perturbations.

We are severely limited as to the size of dataset we are able to simulate due to computation time and memory constraints. The main parameters we have then been able to test, so as not to increase the size of computation, are antenna offsets and position (but not number of source locations), their respective polarisations, and the beamwidth of the antennas. Anecdotally, we found the beamwidth has had perhaps the largest effect on reconstruction results, with a much narrower beam resulting in a superior reconstruction. This was contrary to expectations, on the assumption that a wider beam would allow illumination of a target from more angles and result in higher resolution as in SAR imaging [36, pp 100-101]. The improved results were likely due to each data point being related primarily to a smaller set of parameters (within the beam) and so a better determined system. We can also consider that the more focussed the beam, the closer it becomes to a point source on the ground surface – as was the approximation for 2D reconstructions. We expect that with a more complete coverage of source locations and/or frequencies (better wavenumber coverage in the image domain), the reverse may begin to hold.

A much more thorough investigation into sensitivity and the suitability of acquisition system to 3D GPR FWI is required. It is well known that multiple offsets and a large maximum offset improve resolution in 2D FWI, but whether the same is required in both coordinate axis on the surface for 3D FWI is an open question. It may be that a more modest number of antennas in one direction is acceptable, depending on the number in the other direction, as well as the offsets, polarizations and source locations. The results in Section 7.4.2 go some way to answering this question, but ideally we would like to undertake at least a more formal SVD analysis as was done for 2D reconstructions in Chapter 5.

8.4 Nuisance parameters in full-wave inversion

8.4.1 Introduction

For the landmine detection problem, we desire an accurate solution in a reqion of interest $\Omega_{\mathrm{t}} \subset \Omega$, which is some area surrounding detected metal objects, but are not concerned with the solution outside of this subdomain $\Omega_{\mathrm{n}} = \Omega / \overline{\Omega}_{\mathrm{t}}$. Let us divide our parameter set $\mathbf{m}$ into two sets $\mathbf{m}_{\mathrm{t}}$ and $\mathbf{m}_{\mathrm{n}}$, where each of the elements of $\mathbf{m}_i$ refer to an electromagnetic property within subdomain Ω_i. $\mathbf{m}_{\mathrm{t}}$ is the set of parameters we are interested in and $\mathbf{m}_{\mathrm{n}}$ those of little consequence. We refer to the $\mathbf{m}_{\mathrm{t}}$ as target parameters and the $\mathbf{m}_{\mathrm{n}}$ as nuisance parameters. The fit of $\mathbf{m}_{\mathrm{n}}$ to the data will effect our solution to the $\mathbf{m}_{\mathrm{t}}$, for example due to multiple scattering from objects in Ω_{n} into the region of interest, or by an incorrect conductivity resulting in an over/underestimation of the amplitude in simulated data, thereby resulting in a gradient with too large or small a norm. We cannot then expect to simply fix these parameters and obtain accurate estimates for $\mathbf{m}_{\mathrm{t}}$, and so must have a way to estimate $\mathbf{m}_{\mathrm{n}}$ whilst maintaining an emphasis on finding an accurate $\mathbf{m}_{\mathrm{t}}$. We do this by attempting to define the nuisance parameters implicitly as functions of the target parameters.

Such a formulation well studied in inverse problems in general, to determine a subset of parameters for which either the objective function varies linearly, or there is a closed form solution to minimise the objective with respect to. We follow the formulation of Aravkin and van Leeuwen [16] of estimation these nuisance parameters, before considering how it can be applied to our case where the objective function varies non-linearly with nuisance parameters. This leads us to a novel ‘line search’ type algorithm which attempts to invert in the region of interest, while allowing the nuisance parameters to vary if necessary.

8.4.2 Estimating nuisance parameters

Consider a problem of the form

$$x_{\text{inv}} = \operatorname*{argmin}_{x \in \mathcal{X},\ \theta} g(x, \theta), \tag{8.4.1}$$

where $g : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is a twice differentiable function, $x \in \mathcal{X} \subset \mathbb{R}^m$ is a set of parameters of interest, and $\theta \in \mathbb{R}^n$ are a set of nuisance parameters. For a problem in this form, we assume that for any $x \in \mathcal{X}$ we can easily find

$$\tilde{\theta}(x) = \operatorname*{argmin}_{\theta} g(x, \theta), \tag{8.4.2}$$

though this can be relaxed to cinsider $\tilde{\theta}$ only a local minimum. Rather than solving (8.4.1), we instead minimise the reduced objective

$$\tilde{g}(x) = g(x, \tilde{\theta}(x)). \tag{8.4.3}$$

If $\tilde{g}$ is twice continuously differentiable on some domain $\mathcal{U} \times \mathcal{V}$, $\mathcal{U} \subset \mathbb{R}^m$, $\mathcal{V} \subset \mathbb{R}^n$, then [16, 22]

$$\nabla_x \tilde{g}(\tilde{x}) = \nabla_x g(\tilde{x}, \tilde{\theta}(\tilde{x})) \tag{8.4.4a}$$

$$\nabla_x^2 \tilde{g}(\tilde{x}) = \nabla_x^2 g(\tilde{x}, \tilde{\theta}(\tilde{x})) + \nabla_{x,\theta}^2 g(\tilde{x}, \tilde{\theta}(\tilde{x})) \nabla_x \tilde{\theta}(\tilde{x}) \tag{8.4.4b}$$

for $\tilde{x} \in \mathcal{U}$ and $\tilde{\theta} \in \mathcal{V}$ such that $\nabla_\theta g(\tilde{x}, \tilde{\theta}) = 0$ and $\nabla_\theta^2 g(\tilde{x}, \tilde{\theta})$ is positive definite. Aravkin and van Leeuwen [16] note that the first and second derivatives of $\tilde{g}$ with respect to x may exist even where the smoothness hypotheses are not met.

Equations (8.4.4) suggest a natural iterative approach to minimising $\tilde{g}$ in the unconstrained case,

$$x^{k+1} = x^k - p^k = x^k - \gamma_k H_k^{-1} \nabla_x \tilde{g}(x^k) = x^k - \gamma_k H_k^{-1} \nabla_x g(x^k, \tilde{\theta}(x^k)), \tag{8.4.5}$$

where γ_k is a step length. $H_k = I$, the identity matrix, leads to the steepest descent algorithm, while $H_k = \nabla_x^2 \tilde{g}(x^k)$ yields a modified Newton algorithm. If $\mathcal{X} \subset \mathbb{R}^m$ is a closed and bounded set then we can simply project $x^k - p^k$ onto $\mathcal{X}$. Note that this is not the same as an alternating approach, where we switch between solving for x and θ. Instead, for every update to x we update $\tilde{\theta}(x)$.

This formulation has been applied to full-wave inversion, for example for estimating source amplitude [16]. Often the source wavelet is estimated prior to the inversion [86, 113], or can be solved for as part of the optimisation scheme [30]. To define the frequency-dependent source amplitude and phase as nuisance parameters, let $\mathcal{F}_i(\mathbf{m})$ be the unit-amplitude and zero phase-shifted simulated GPR data for $\mathbf{m}$ at frequency ω_i, then the full-wave problem can be written (neglecting regularisation terms) as

$$\mathbf{m}_{\text{inv}} = \underset{\text{m}}{\operatorname{argmin}} \sum_i \|\bar{a}_i \mathcal{F}_i(\mathbf{m}) - \mathbf{d}_i\|^2, \tag{8.4.6a}$$

$$\bar{a}_i(\mathbf{m}) = \underset{a_i}{\operatorname{argmin}} \|a_i \mathcal{F}_i(\mathbf{m}) - \mathbf{d}_i\|^2. \tag{8.4.6b}$$

Indeed, solving (8.4.6b) for the $\bar{a}_i$ given $\mathbf{m}$ is a linear problem with a closed-form least-squares solution

$$\bar{a}_i = (\mathcal{F}_i^T \mathcal{F}_i)^{-1} \mathcal{F}_i^T \mathbf{d}_i = \frac{\langle \mathcal{F}_i, \mathbf{d}_i \rangle}{\langle \mathcal{F}_i, \mathcal{F}_i \rangle}. \tag{8.4.7}$$

Note that this can be calculated without requiring any further forward simulations.

8.4.3 Inverting target parameters

For the landmine detection problem, we are interested in accurately reconstructing a target located by a metal detector, and are less interested in resolving the surrounding soil and clutter. Moreover, we expect de-mining personnel to have explored in greater detail the area immediately around the metal target as they try to determine its make-up, so GPR data will be less sensitive to positions away from the target. We then have poor information about other scatterers away from the target, so may not be able to resolve these features fully, but they cannot be entirely ignored as they will have affected the data (for example through multiple scattering interactions with the target). We wish to pose this in the framework of nuisance parameters, where the target parameters are those surrounding the detected metal object and the nuisance parameters a sufficient distance away.

Separating our parameter vector into two sets $\mathbf{m}_\text{t}$ and $\mathbf{m}_\text{n}$, where $\mathbf{m}_\text{t}$ are the parameters immediately around the target (either permittivity alone or both permittivity and conductivity), and $\mathbf{m}_\mathrm{n}$ are all others, we would like to set

$$\bar{\mathbf{m}}_2(\mathbf{m}_\mathrm{t}) = \operatorname*{argmin}_{\mathbf{m}_\mathrm{n}} \mathcal{J}(\mathbf{m}_\mathrm{t}, \mathbf{m}_\mathrm{n}). \tag{8.4.8}$$

Unfortunately, this would involve a global optimisation problem on (almost) the same scale and complexity as solving for the full parameter vector $\mathbf{m}$ for each trial $\mathbf{m}_\mathrm{t}$, which is prohibitive. Instead, for each trial target values $\mathbf{m}_\mathrm{t}$ we propose taking a linearised step in $\mathbf{m}_\mathrm{n}$, given in equation (4.3.7), such that

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \begin{bmatrix} \alpha_\mathrm{t}\mathbf{p}_\mathrm{t} \\ \bar{\alpha}_2(\alpha_1)\mathbf{p}_\mathrm{n} \end{bmatrix}, \tag{8.4.9a}$$

$$\alpha_\mathrm{t} = \operatorname*{argmin}_{\alpha_\mathrm{t}\in(0,\alpha_\mathrm{max}]} \phi(\alpha_\mathrm{t}, \bar{\alpha}_\mathrm{n}(\alpha_\mathrm{t})), \tag{8.4.9b}$$

$$\bar{\alpha}_\mathrm{n} = \operatorname*{argmin}_{\alpha_2\in(0,\alpha'_\mathrm{max}]} \phi(\alpha_\mathrm{t}, \alpha_\mathrm{n}). \tag{8.4.9c}$$

Here, $\mathbf{p} = [\mathbf{p}_\mathrm{t};\ \mathbf{p}_\mathrm{n}]$ is the l-BFGS descent direction separated into target and nuisance parameters, α_t is a step length for $\mathbf{p}_\mathrm{t}$ found by a (strong) Wolfe linesearch, and $\bar{\alpha}_\mathrm{n}$ is a step length found by the linearisation (4.3.7).

The linearised step length $\bar{\alpha}_\mathrm{n}(\alpha_\mathrm{n})$, for each trial step α_t, is found by interpolation of $\delta\mathbf{d}$ between $\alpha_2 = 0$ and $\alpha_\mathrm{n} = \alpha_\mathrm{t}$. To safeguard this selection, preventing unnecessary forward evaluations at multiple close steplengths, if $\bar{\alpha}_\mathrm{n} < b\alpha_\mathrm{t}$, $0 < b \ll \infty$, it is set to 0, and if $(1-d_1)\alpha_\mathrm{t} < \bar{\alpha}_\mathrm{n} < (1+d_2)\alpha_\mathrm{t}$, $0 < d_1, d_2 \ll 1$, it is set to α_y. This involves at most 3 forward computations per trial step α_t, but may require only 2. Further, we set

$$\alpha'_\mathrm{max} = \min(\alpha_\mathrm{max}, \delta\alpha_t), \tag{8.4.10}$$

where $\delta > 0$ can either be set dynamically throughout the inversion process, beginning with a small number ($0 < \delta < 1$) and gradually increasing it, or a constant $1 \leq \delta \ll \infty$. This prevents a much larger update in $\mathbf{m}_\mathrm{n}$ than in $\mathbf{m}_\mathrm{t}$), and poor scaling between the two parameter sets. If a larger step in α_n was preferred, we would likely do better to try a larger α_t since these are the parameters we wish to resolve. Further, starting with a small value of δ forces the inversion to fit the target parameters as closely as possible without updating the nuisance parameters in early iterations. The scheme is shown in Algorithm 6 and Matlab code is given in Appendix B.1 [194].

Algorithm 6 Nuisance parameter linesearch

```
Input $m$, $p_t$, $p_n$, $\alpha_{\max} > 0$, $\delta > 0$;
while Linesearch criteria are not met do
    Select an $\alpha_t \in (0, \alpha_{\max}]$;                      ▷ e.g. from Algorithm 3 or 4
    $\phi_0 \leftarrow \mathcal{J}(m + \alpha_t p_t)$;                 ▷ Retain data residual $\delta d_0$
    $\phi_1 \leftarrow \mathcal{J}(m + \alpha_t p_t + \alpha_t p_n)$;  ▷ Retain data residual $\delta d_1$

    $\alpha_n \leftarrow \alpha_t \Re \left\{ \frac{(\delta d_0 - \delta d_1)^T \delta d_0}{\|\delta d_0 - \delta d_1\|_2} \right\}$;   ▷ Linear step from equation (4.3.7)

    if $\alpha_n > \alpha_{\max}$ or $\alpha_n > \delta\alpha_t$ then
        $\alpha_n \leftarrow \min(\alpha_{\max}, 1.5\alpha_t)$;
    end if

    if $\alpha_n \leq 0.25\alpha_t$ then
        $\alpha_n \leftarrow 0$;
        $\phi_\alpha \leftarrow \phi_0$;
    else if $0.9\alpha_t \leq \alpha_n \leq 1.25\alpha_t$ then
        $\alpha_n \leftarrow \alpha_t$;
        $\phi_\alpha \leftarrow \phi_1$;
    else
        $\phi_\alpha \leftarrow \mathcal{J}(m + \alpha_t p_t + \alpha_n p_n)$;
    end if
end while
Increase $\delta$
```

8.4.4 Remarks

Algorithm 6 attempts to reduce the dimensionality of the optimisation problem, by considering the parameters outside of a region of interest as nuisance parameters. The amount in which these are updated is limited by the size of update in parameters describing the region of interest. To justify the approach, we can consider (8.4.9) as a globalisation method for the more familiar constrained optimisation problem

$$\begin{bmatrix} \mathbf{m}_{\mathrm{t,inv}} \\ \mathbf{m}_{\mathrm{n,inv}} \end{bmatrix} = \underset{\mathbf{m}_{\mathrm{t}} \in \mathcal{D},\, \mathbf{m}_{\mathrm{n}} \in \chi(\mathbf{m}_{\mathrm{t}})}{\operatorname{argmin}} \quad \mathcal{J}(\mathbf{m}_{\mathrm{t}}, \mathbf{m}_{\mathrm{n}}). \tag{8.4.11}$$

Here, $\mathcal{D}$ is the domain of $\mathbf{m}_{\mathrm{t}}$ enforcing physical parameters, and $\chi(\mathbf{m}_{\mathrm{t}})$ are a set of non-linear constraints enforcing the implicit relationship $\bar{\mathbf{m}}_{\mathrm{n}} = \bar{\mathbf{m}}_{\mathrm{t}}(\mathbf{m}_{\mathrm{n}})$. When this non-linear constraint is not held, the step $\bar{\alpha}_{\mathrm{n}}\mathbf{p}_{\mathrm{n}}$ is taken towards $\bar{\mathbf{m}}_{\mathrm{n}} = \bar{\mathbf{m}}_{\mathrm{n}}(\mathbf{m}_{\mathrm{t}})$ (changing the ratio between α_{t} and α_{n}).

Note that since $\mathbf{p} = [\mathbf{p}_{\mathrm{t}}; \mathbf{p}_{\mathrm{n}}]$ is a descent direction, and our scheme always tests $\alpha_1 = \alpha_2$ for $\delta \geq 1$, Algorithm 6 guarantees reduction in the objective function

$$\mathcal{J}(\mathbf{m}_k + \alpha_{\mathrm{t}}\mathbf{p}_{\mathrm{t}} + \alpha_{\mathrm{n}}\mathbf{p}_{\mathrm{n}}) \leq \mathcal{J}(\mathbf{m}_k + \alpha\mathbf{p}) \leq \mathcal{J}(\mathbf{m}_k), \tag{8.4.12}$$

where the final inequality is strictly provided such an $\alpha > 0$ can be found by the linesearch method.

Meles et al [110] give an algorithm for simultaneously updating permittivity and conductivity, in which two linear step lengths are calculated for the two parameter types, both about $\mathbf{m}_k$. This accounts for the different scaling between the two parameter types. If we consider permittivity the target parameter, and conductivity the nuisance parameter, then Algorithm 6 is similar to their approach. The difference being that in Algorithm 6, the step length calculated for conductivity takes its linearisation about $\mathbf{m}_k + \alpha_{\mathrm{t}}\mathbf{p}_{\mathrm{t}}$, and not about $\mathbf{m}_k$. If the update direction $\mathbf{p}_{\mathrm{t}}$ is well scaled (which it ought to be after at least a few iterations of l-BFGS) and the first trial step length α_{t} is acceptable, then our algorithm has a similar computational cost to that of Meles et al.

8.4.5 Numerical experiment

To demonstrate the applicability of Algorithm 6, we present a numerical experiment in reconstructing a target in a known location, surrounded by a highly inhomogeneous background. For this, we take a spherical target of $\epsilon_r = 3.25$ and 2.5 cm radius, in a stochastic background medium described by an ellipsoidal autocorrelation function [73, 78] (see Appendix A.4). Such a function gives rise to multi-scale random medium, and by varying autocorrelations length scales in the horizontal and vertical directions (denoted a and b respectively) we can generate a range of ground types including layered or granular media. We use length scales a and b which are also 5 cm, the same as the target. We expect that with such a selection will make the reconstruction problem more difficult than either much larger or smaller scale features, which would have a different affect across the bandwidth to the target.

Figure 8.9 shows the reconstruction using Algorithm 6 after only 10 iterations, and Figure 8.10 the relative residuals and step lengths at each iteration of l-BFGS. We see that in the centre of the target we have already found close to the correct permittivity, but have not yet correctly resolved the shape. In the first iteration, the nuisance parameters are not updated at all. In the next 4, the update in these parameters is limited by the maximum step length α'_{max}.At this point, some of the background stochastic medium has been resolved, specifically the lower permittivity directly above and to the sides of the target. Presumably, these nuisance parameters had the greatest effect on the ability to resolve the target parameters, as after iteration 5 the nuisance parameters were updated little. We would not expect to do much to accurately resolve the background medium, particularly after so few iterations.

The ability of Algorithm 6 to adapt the descent direction for an appropriate update outside of the ROI is highlighted by the contour plot shown in Figure 8.11. It is quite clear that in the first iteration, the steepest-descent direction ($\alpha_{\mathrm{t}} = \alpha_{\mathrm{n}}$) is only a good direction for a short distance ($\alpha < 0.02$), after which the change in curvature makes any step in α_{n} yield little reduction in the objective function $\mathcal{J}$. By $\alpha_{\mathrm{t}} = 0.1$, any $\alpha_{\mathrm{n}} > 0$ increases the objective value (though only a small amount). In this particular case, a quadratic approximation to the objective function $\mathcal{J}$ along $\mathbf{p}_{\mathrm{t}}$ is also quite a good one (we have a single, smooth minimum close to parabolic), so a line search can quickly find the minimum via quadratic interpolation.

Our experience from 2D FWI is also that we would be unlikely to resolve the exact shape of the target so soon. It takes many iterations of FWI to accurately determine the shape of targets, and the implementation of TV is likely insufficient for 3D FWI: a

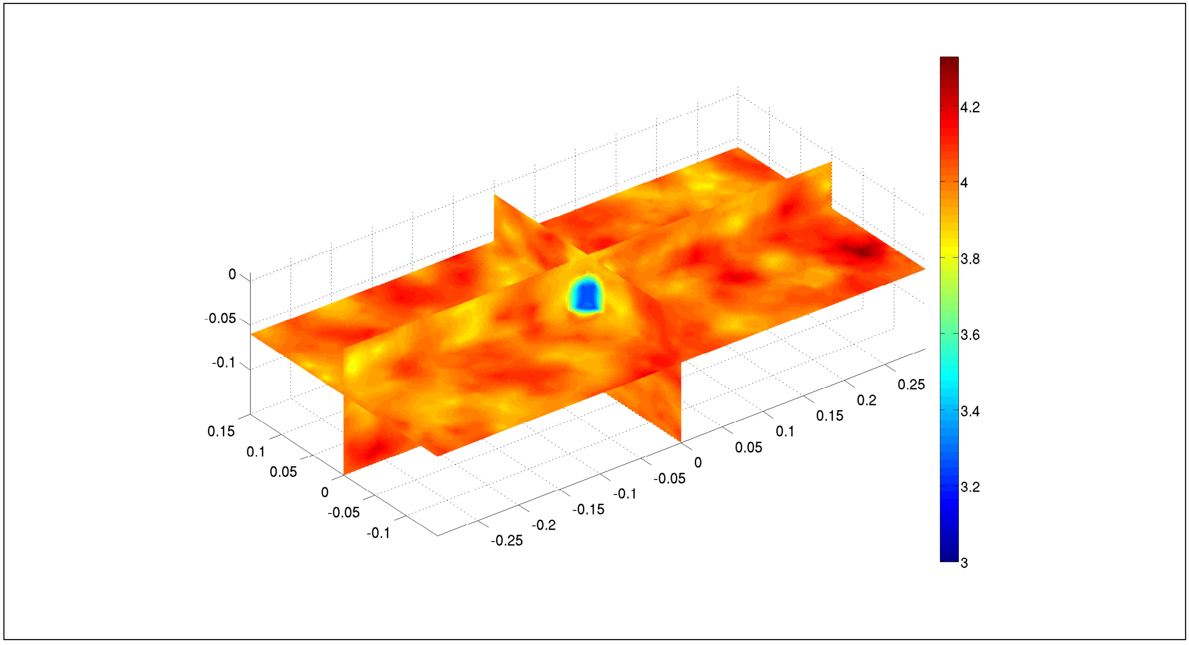


Figure 8.8: Slice image of a single cubic target of permittivity $\epsilon_r = 3.25$ in stochastic background medium with mean permittivity $\epsilon_r = 4$.

primal dual method may be required due to the complexity. Yet, the ability to quickly resolve the permittivity within the region of interest is significant for the landmine detection problem. In this numerical experiment, we are able to extract the relevant information from the non-linear and noisy data quickly. Each iteration is more costly, requiring a minimum of 2 evaluations of the objective function, yet we determine the permittivity of the target in far fewer iterations than even a 2D reconstruction, saving on computational cost overall.

8.5 Discussion

In the preceding section, we presented an algorithm to resolve the shape and permittivity of a target object in a region of interest. This can be defined to be some volume surrounding a metal object located by a metal detector, and while the reconstruction in Figure 8.9 is not fully resolved (we allowed only a small number of iterations), we expect that if it were allowed to continue this would be able to tell us precisely what was surrounding the metal object. We have only considered reconstruction of a domain with a flat ground surface. While a rough and varying terrain could cause significant difficulties in resolving the subsurface, considering such features was beyond the scope of this thesis.

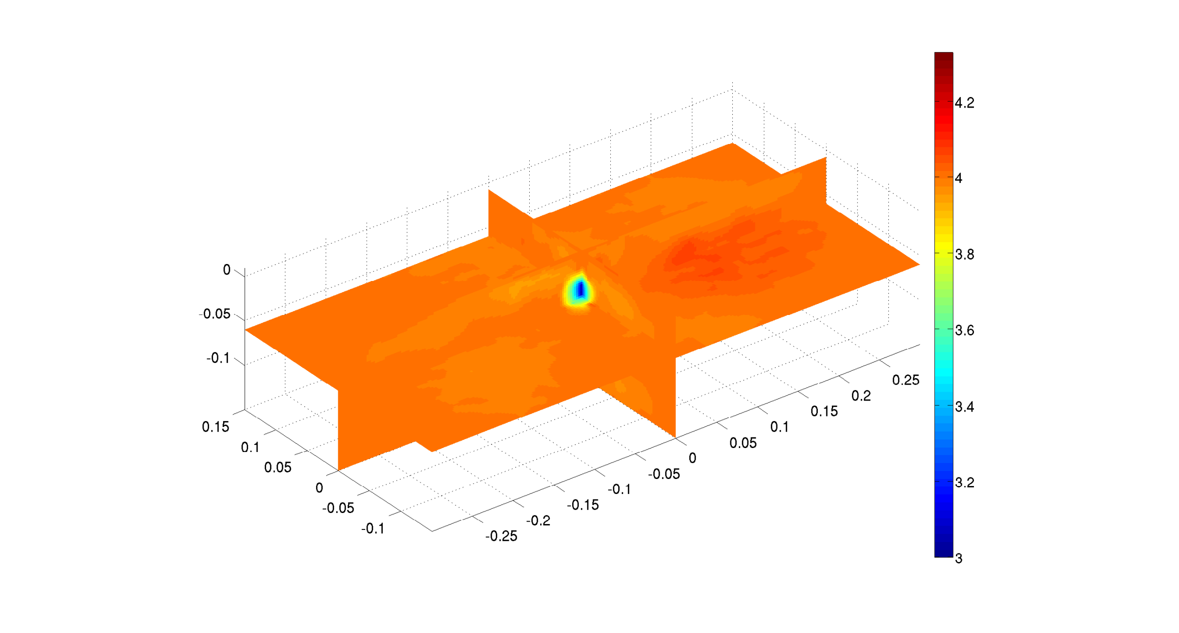
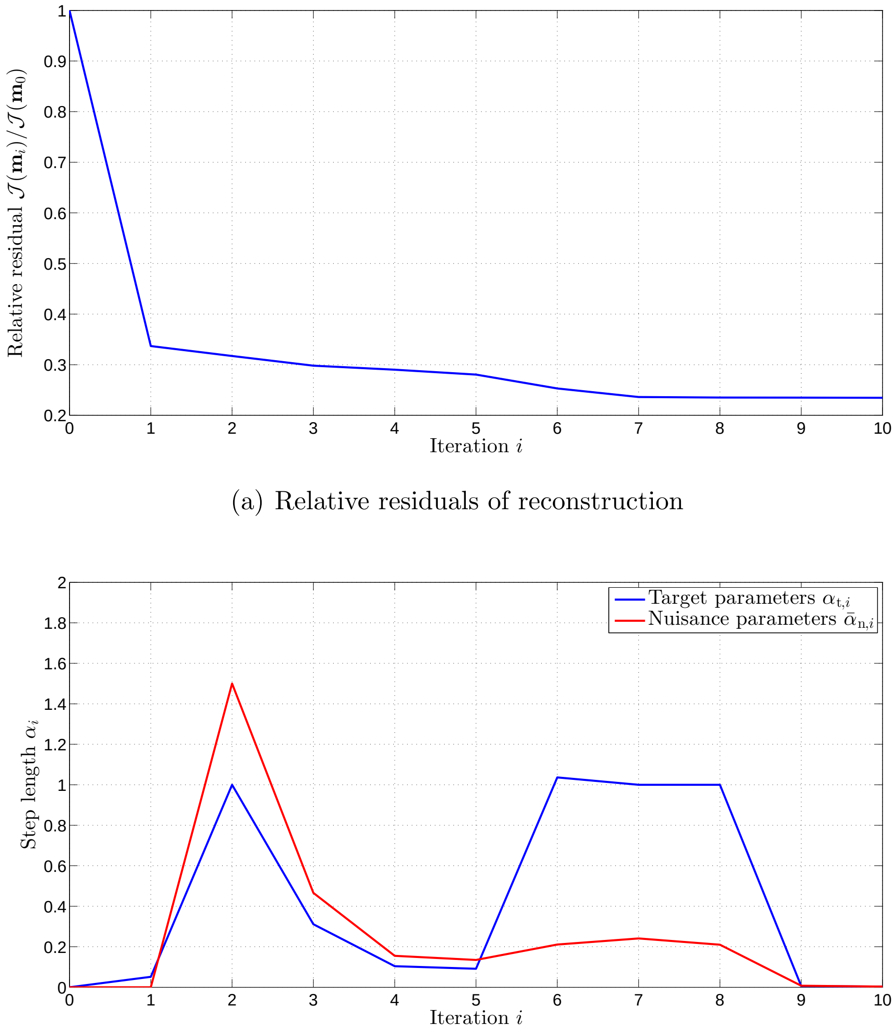


Figure 8.9: Reconstruction after 10 iterations of a single cubic target of permittivity $\epsilon_r = 3.25$ in stochastic background medium with mean permittivity $\epsilon_r = 4$, using the nuisance parameter linesearch Algorithm 6

Why did we not test our algorithm's ability to reconstruct a landmine? Being able to accurately reconstruct landmines is unlikely to improve clearance rates at all. If mine clearance personnel use a metal detector alone, then they must carefully remove every piece of metal from the ground. This process can only be sped up by safely showing what is *not* a landmine: given a piece of metal which could be a firing pin, we want our algorithm to be able to demonstrate that the material surrounding it *can not* be a landmine so that it can be ignored. Particularly, we want this to be possible in the cases which systems such as MINEHOUND can not account for. That is, when there is some object large enough for the GPR to detect around the metal fragment, but when this is not a landmine. We might still like the system to be able to resolve a landmine, so that if one is present (perhaps a non metallic mines) de-mining personnel are sure of what the target is, and how it is oriented, and so can operate with more caution. This is only a secondary objective though, as de-mining personnel will treat *all* targets which they cannot me sure are *not* landmines as though they are.

How can we be sure that our reconstruction results rule out there being a landmine present? We must be sure of the accuracy of our reconstruction, and that the reconstructed model fits the data *significantly* better than the presence of any landmine could. This would require a long term testing period, creating a large bank of data



(a) Relative residuals of reconstruction

(b) Step lengths in target and nuisance parameters

Figure 8.10: (a) Relative residuals and (b) step lengths for the first 10 iterations of Algorithm 6 for a single target in a stochastic background medium.

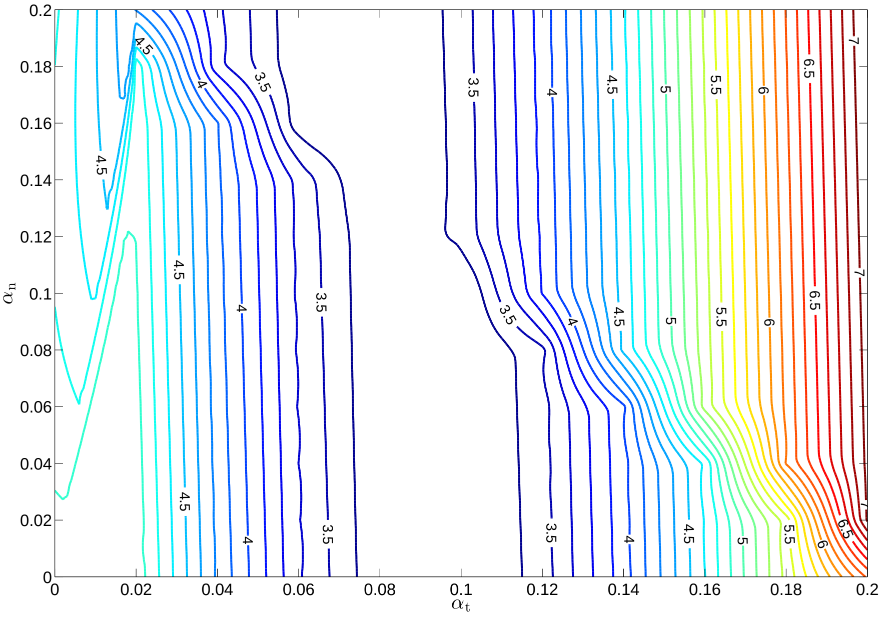


Figure 8.11: Contour plot for $\mathcal{J}(\mathbf{m} + \alpha_t \mathbf{p}_t + \alpha_n \mathbf{p}_n)$ in the first iteration of l-BFGS.

and reconstructions, to determine how accurate a reconstruction is likely to be given how closely it fits the data. We would need to take into account how accurately we are able to simulate the GPR data, including how accurately we can simulate landmine data and what its effect on the data would be. See for example Simonson [158] for a discussion on the statistical significance of mine clearance trial data

Figure 8.12 shows the absolute difference between data simulated on the reconstruction in Figure 8.9, to data simulated on the test domain in Figure 8.8 in which we vary the target permittivity between $\epsilon_r = 1$ and 5, but keep the same stochastic background medium. Since the reconstruction fits much poorer a true target permittivity of less than 3 or greater than 3.5, we can be sure of correctly identifying a range of possible material permittivities. Figure 8.14 compares some of the data simulated for the domain shown in Figure 8.8 (blue) to that simulated in the reconstructed domain (green), and to that simulated for a mine-like target in a stochastic background (shown in Figure 8.13). It is clear that the simulated and reconstructed datasets are far more similar to one-another than either is to the data from the mine-like target. If we have complete bank of possible mine-like targets available to compare to, we could be sure that this GPR data was not due to a landmine, and safely ignore the target. A system such as MINEHOUND however would record a signal, and the de-mining personnel

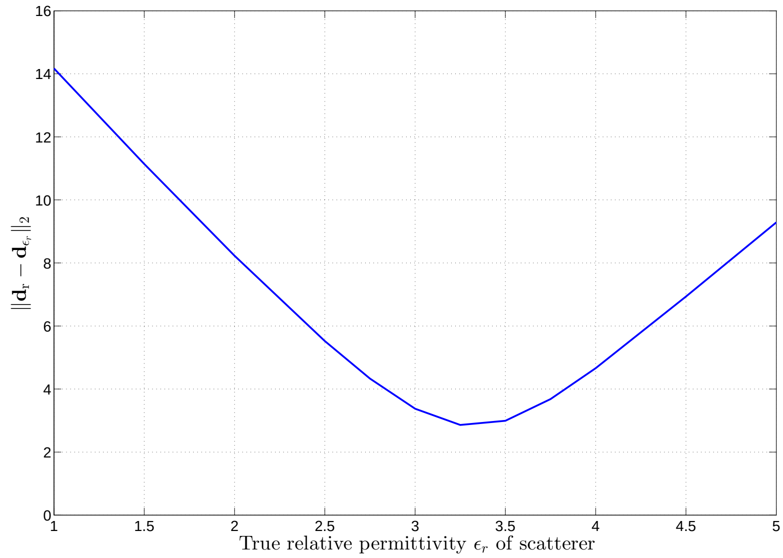


Figure 8.12: Absolute difference of reconstructed data $\mathbf{d}_{\mathrm{r}}$, and data simulated for a single target $\mathbf{d}_{\epsilon_r}$ of permittivity $\epsilon_r \in [1, 5]$ in a stochastic background medium as with Figure 8.8, $\|\mathbf{d}_{\mathrm{r}} - \mathbf{d}_{\epsilon_r}\|_2$.

would need to carefully excavate regardless. To understand the statistical significance of reconstructed results we would need to undertake a large number of comparisons such as those shown in Figures 8.12 and 8.14, which would tell us statistically how closely we have resolved the domain, and when we are able to say with certainty that features in the data can not be due to a landmine rather than the reconstructed target.

Why not then simply use a look-up table to compare recorded GPR traces to those of known targets? The answer is the non-linearity of the inverse problem: a look-up table cannot take into account a highly inhomogeneous or contaminated background, causing multiple scattering, or local variations in soil type and topography of the ground surface. A lookup table may perform poorly where the data due to a target is significantly different in a cluttered environment (due to multiple scattering or curving of the ray-path) to how it is in a homogeneous or known background. We refer again to Figure 8.7, showing a slice of the gradient for a single target in stochastic background, and recall that the gradient is equivalent to a linear inversion (back-projection). Direct comparison of the data is essentially equivalent to comparison of a linear inversion, since we can define a linear inversion such that the map between data and image is invertible (e.g. using a Radon transform). Noting the large perturbations away from the target in the gradient, owing to the stochastic medium, we would be unlikely to

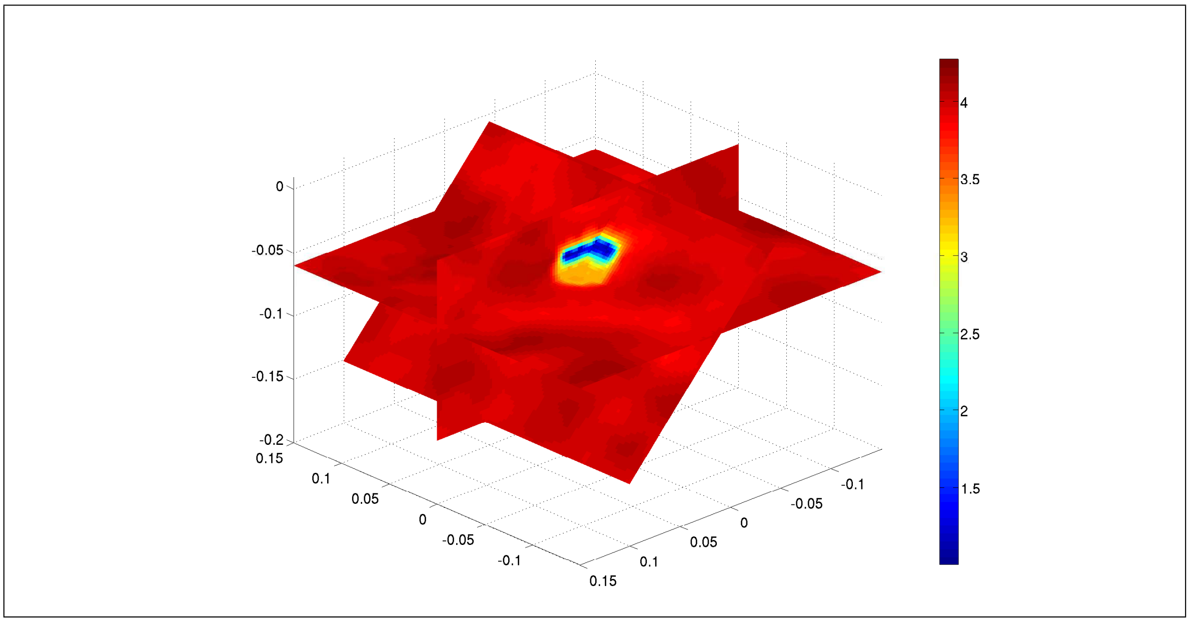


Figure 8.13: Surrogate mine target in a stochastic background medium

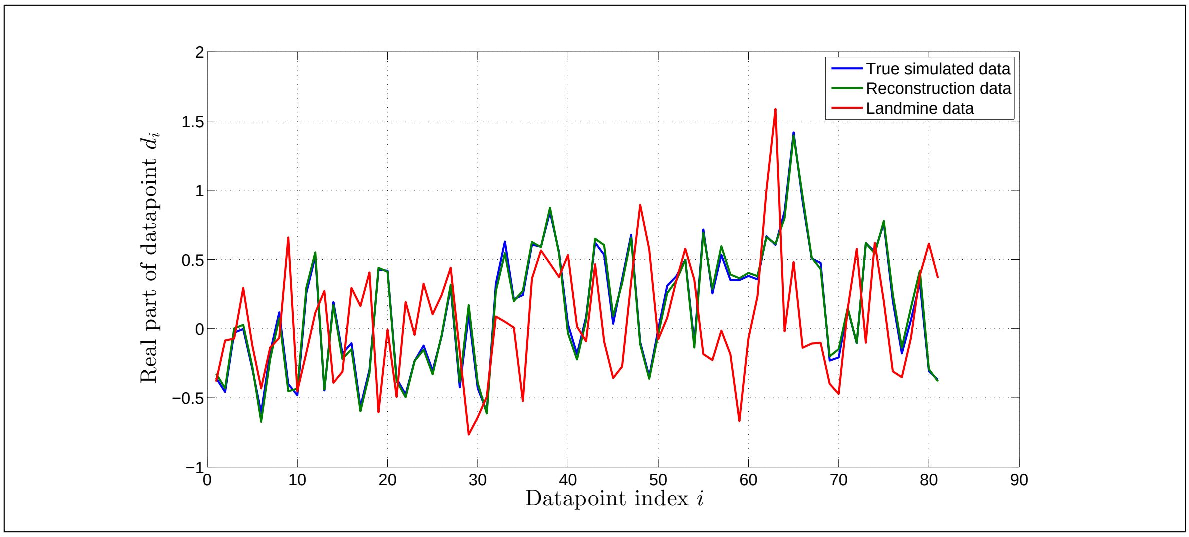


Figure 8.14: Comparison of data simulated on test domain Figure 8.8 (blue), reconstructed domain Figure 8.9, and for a mine-like target in stochastic medium (red), at 1.5 GHz and corresponding to a single receiver passing over the target

be able to rule this out as not a landmine from the simple image (or traces) alone. We would argue then, that it is more sensible to compare reconstructed results to possible landmines in some way.

8.6 Conclusions and outlook

In this chapter, we have undertaken numerical experiments in 3D Full-Wave Inversion for GPR, using a combined Finite-Element Boundary-Integral solver to simulate the forward problem. These are both the first published 3D results, and the first published use of a FE-BI method as the forward model, for GPR FWI. The study was of course limited due to the extremely large computational cost of solving Maxwell's equations in 3D: a full comparison of the inversion parameters, such as regularisation type and amount, l-BFGS memory parameter, Wolfe condition parameters, or other optimisation methods, could not be undertaken. Rather, the implementation and experiments demonstrated in this chapter are a proof of principle, and a useful starting point, for future work in 3D GPR FWI. We also believe that moving towards a full 3D model in FWI is the right way to solve the problem, as a 2D model neglects many scattering effects which may be seen in the data. It is particularly important if FWI is to be used for the mine detection problem, since a 2D model of the subsurface cannot represent a landmine.

Due to the computational cost, we were limited as to the size of dataset we could simulate and expect to invert within given time constraints. This generally resulted in a poorer descent direction than we would expect from a 'full' GPR dataset. We noticed (in early iterations) similar shadows in the update direction below the target position as those observed in the 2D problem, likely due to reflections off the bottom of the target being misplaced due to an incorrect wavespeed in the current model, as well as multiple reflections within the target itself. Due to the much more complex scattering problem in 3D, the l-BFGS optimisation (and likely choice or implementation of regularisation) struggled to account for these coherent artefacts in a small number of iterations. Anecdotally, we found a narrower GPR source beam provided a better descent direction, presumably because the information content of each data-point was associated with a smaller region. We expect more complete data coverage would also alleviate this problem.

Finally, we considered the applicability of 3D FWI to the landmine detection problem, questioning what we really want to be able to find from such a method and in what circumstances. The answer is we are only interested in determining the properties of some detected target, particularly in a heavily cluttered environment. This led us to develop a novel linesearch type algorithm, which attempts to treat parameters away from the target as a nuisance. The step length taken in the nuisance parameters is defined implicitly in terms of the parameters describing the target, and so we concentrate the reconstruction on determining the detected target. We tested the algorithm in the case of a single target in a known location in a highly stochastic background. We were quickly able to determine the permittivity of the target, while the background minimum had a minimal update, and so the algorithm performed as wanted. For landmine detection, we believe such an algorithm could therefore be useful in providing additional quantitative information about a target, while being able to account for multiple scattering from clutter and local soil variations.

The reconstructions presented were in a necessarily small domain, due to memory constraints. As such, both 3D FWI and the results presented here are not yet applicable to many applications of GPR which require knowledge of a much larger area. As processing power improves, and large amounts of memory become more available, we expect to be able to carry out much more complex 3D inversions of larger GPR datasets. Particularly, we do not believe it is an unrealistic aim to treat the mine detection problem using FWI, with a small scale inversion such as Algorithm 6, as clearance of the worlds landmines is a very long term goal. We can expect landmines still to be a significant problem in (say) 10 years time, when perhaps computer power is sufficient for such an imaging method in the field. Eventually, whether FWI is computationally worthwhile will cease to become a concern. One must then determine if it does enable a reduction in the rate of false positives (we believe that it must), or even if it is useful as a standalone method (i.e. without a metal detector) to detect no metal landmines. Finally, there is a question of how the information provided by FWI ought to be presented to de-mining personnel: a permittivity distribution of the subsurface is almost certainly inappropriate.

Chapter 9

A Polarization Tensor Preconditioner for l-BFGS

9.1 Introduction

In Chapter 7, we presented a polarization tensor expansion for Maxwell's equations which was suitable for GPR. The expansion is valid where one considers scatterers which are of a small volume, but may be of any material contrast to the background medium. We discussed this asymptotic expansion in terms of sensitivity, and used this idea for numerical experiments into increasing distinguishability of landmines. This is unlike the more common understanding of sensitivity, based on the Taylor expansion (or Born series), which is valid for any size scatterer but only of a small material contrast. The two must of course coincide, when we have both small scatterers and of low contrast. Instinctively then, we would like to know if this idea that polarization tensors are related to sensitivity can be used to help our non-linear reconstruction.

Often, the first iteration of l-BFGS results in only a small reduction in residual when initialising the approximate Hessian as the identity matrix. It may also take many evaluations of the cost function to find a suitable first step length if the gradient is poorly scaled. As we discussed in Section 2.5.5, this is because the Hessian contains information about some physical properties of the scattering problem: first order multiples, and data illumination. In this section, we will consider the polarization tensor expansion may be used to precondition the first iteration of an l-BFGS scheme, accounting for data illumination. We derive an approximate Hessian matrix based on the polarization tensor expansion. We then present a trust-region algorithm, in which the objective function is approximated by the Taylor series expansion but using the approximate Hessian previously derived. The algorithm returns an updated initial parameter model and an initial Hessian approximation for this model. The initial Hessian approximation can be used to precondition an l-BFGS scheme, which is an improvement over initialising with the identity matrix. Using a polarization tensor expansion to precondition the l-BFGS method for solving a non-linear optimisation problem is an idea novel to this thesis.

9.2 A Gauss-Newton type preconditioner

In a quasi-Newton inversion method, we require an initial approximation to the Hessian H_0 for the initial parameterisation $\mathbf{m}_0$. There is no general rule as to what this ought to be for best results, but there are certain features we would like it to contain: it ought to be positive definite (so that $\mathbf{p} = -H^{-1}\mathbf{g}$ is a descent direction) and symmetric; and it ought to mimic some of the physical attributes contained in the true Hessian, such as compensating for parameter illumination. We can think of choosing an initial Hessian as preconditioning the gradient descent direction, or indeed as a form of regularisation.

One choice is the Gauss-Newton approximation of the Hessian,

$$B(\mathbf{m}) = \Re\left\{J^T J^*\right\}, \qquad B_{ij} = \Re\left\{\sum_{r=1}^{n} \frac{\partial \mathcal{F}_r}{\partial m_i}\frac{\partial \mathcal{F}_r^*}{\partial m_j}\right\},$$

where $\mathcal{F}_r$ are the simulated data, or some diagonals of B. Recall from Section 2.5.5 that the Gauss-Newton approximation of the Hessian compensates for the different illumination of parameters by the source–receiver array, as well as some of the non-linearity from the saturation effect. Unfortunately, this matrix is prohibitively large to calculate and store for 3D reconstructions. Indeed, this is still the case if we only wished to store just some diagonals of B, since one is still required to solve $p = \text{length}(\mathbf{m})$ additional forward problems regardless of how many off-diagonals are stored. To alleviate this problem, Shin et al [154] suggest using the virtual source matrix

$$F = \left[-\frac{\partial A}{\partial m_1}\mathbf{u}, \ldots, -\frac{\partial A}{\partial m_p}\mathbf{u}\right],$$

where A is the finite element or finite difference system matrix and $\mathbf{u}$ is the numerical solution to the forward problem. Instead of using the Gauss-Newton Hessian B, they use a pseudo-Hessian $\mathrm{diag}(F^TF)$ which avoids the p additional forward solutions. This matrix does account for the coverage of parameters by the source fields, but not the coverage by receivers (i.e. it negates the correlation with the adjoint fields). One can also approximate the Gauss-Newton terms by the Born approximation for scalar-valued waves, as with equation (7.4.4) (where this expression assumes point scatterers).

To derive an approximate Gauss-Newton Hessian for the electromagnetic inverse problem, let us suppose we have started our inversion procedure from a homogeneous domain $\mathbf{m}_0$, which we have discretised into small volumes $\alpha^3 B_i$ with centres $\mathbf{r}_i$, which all have approximately equal sizes in each dimension. We approximate the cost function by the polarization tensor approximation,

$$\mathcal{J}(\mathbf{m}) = \frac{1}{2}\|\mathcal{F}(\mathbf{m}) - \mathbf{d}\|^2 \approx \frac{1}{2}\|\mathbf{u}_0 + \mathbf{u}(\delta\mathbf{m}) - \mathbf{d}\|^2 = \frac{1}{2}\|\mathbf{u}(\delta\mathbf{m}) - \delta\mathbf{d}_0\|, \tag{9.2.1}$$

where $\mathbf{u}_0 = \mathcal{F}(\mathbf{m}_0)$ and $\mathbf{u}$ is the asymptotic approximation of the scattered fields, the j^{th} component of which is given by

$$\begin{aligned} u_j(\delta\mathbf{m}) = \sum_i \mathbf{p}_j \cdot \Bigg(&\int_{\partial\Omega} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}_j, \mathbf{r}') \cdot \left(\alpha^3\omega^2\epsilon^0\mu^0 \left(\frac{\epsilon^0}{\epsilon^i} - 1\right) \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i) \mathscr{M}_{B_i}\left(\frac{\epsilon^0}{\epsilon^i}\right) \mathbf{E}^{\text{in}}(\mathbf{r}_i)\right) \mathrm{d}S' \\ &- \mathrm{i}k_0 Z_0 \int_{\partial\Omega} \overline{\overline{\mathrm{G}}}(\mathbf{r}_j, \mathbf{r}') \cdot \left(-\mathrm{i}\alpha^3\omega\epsilon^0 \left(\frac{\epsilon^0}{\epsilon^i} - 1\right) \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i) \mathscr{M}_{B_i}\left(\frac{\epsilon^0}{\epsilon^i}\right) \mathbf{E}^{\text{in}}(\mathbf{r}_i)\right) \mathrm{d}S' \Bigg), \end{aligned} \tag{9.2.2}$$

in which we have $\epsilon_0 \delta m_i = \epsilon^i$ and $\epsilon_0 m_0 = \epsilon^0$, and $\mathbf{r}_j$ and $\mathbf{p}_j$ are the location and polarization of the j^{th} measurement. The approximation is valid where the vector $\delta\mathbf{m}$ is sparse in such a way that the δm_i are sufficiently separated – certainly it is exact at $\delta\mathbf{m} = \mathbf{0}$ and as the size of parameter volumes tends to zero, and valid for a perturbations in a single parameter.

Let us use this expansion to approximate the Jacobian components

$$J_{ij}(\mathbf{m}_0 + \delta\mathbf{m}) = \left.\frac{\partial \mathcal{F}_i}{\partial m_i}\right|_{\mathbf{m}_0 + \delta\mathbf{m}} \approx \left.\frac{\partial u_r}{\partial m_i}\right|_{\delta\mathbf{m}}.$$

Considering a change only in the i^{th} volume in real permittivity $m_i\epsilon_0 = \epsilon^i$, where in all other volumes $m_j\epsilon_0 = \epsilon^j = \epsilon^0$, $i \neq j$, we have from (7.3.10) that the rate of change in a data term is given by

$$\begin{aligned} \frac{\partial u_r}{\partial m_i} = \frac{d\epsilon^i}{dm_i}\frac{\partial}{\partial \epsilon^i}\Bigg[\mathbf{p}_r \cdot \Bigg(&\int_{\partial\Omega} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(\alpha^3\omega^2\epsilon^0\mu^0 \left(\frac{\epsilon^0}{\epsilon^i} - 1\right) \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i) \mathscr{M}_{B_i}\left(\frac{\epsilon^0}{\epsilon^i}\right) \mathbf{E}^{\text{in}}(\mathbf{r}_i)\right) \mathrm{d}S' \\ &- \mathrm{i}k_0 Z_0 \int_{\partial\Omega} \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(-\mathrm{i}\alpha^3\omega\epsilon^0 \left(\frac{\epsilon^0}{\epsilon^i} - 1\right) \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i) \mathscr{M}_{B_i}\left(\frac{\epsilon^0}{\epsilon^i}\right) \mathbf{E}^{\text{in}}(\mathbf{r}_i)\right) \mathrm{d}S' \Bigg)\Bigg]. \end{aligned} \tag{9.2.3}$$

As mentioned in Section 7.2.3, yhere exists an ellipsoid with the same polarization tensor as for B_i, and since we have prescribed that the B_i are approximately equal in each dimension then this ellipsoid must be close to spheical. The polarization tensor is therefore of the form

$$\mathscr{M}_{B_i}\left(\frac{\epsilon^0}{\epsilon^i}\right) = \frac{|S_i|\epsilon^i}{\epsilon^0 + (\epsilon^i - \epsilon^0)/3} I. \tag{9.2.4}$$

Here, $|S_i|$ is the volume of the sphere with the same tensor as B_i, which may be different to $|B_i|$. Substituting into (9.2.3), we have that

$$\begin{aligned}\frac{\partial u_r}{\partial m_i} = \epsilon_0 \frac{\partial}{\partial \epsilon^i}\Bigg[\frac{V_i\epsilon^0(\epsilon^0 - \epsilon^i)}{\epsilon^0 + (\epsilon^i - \epsilon^0)/3}\mathbf{p}_r \cdot \Bigg(\int_{\partial\Omega} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(\omega^2\mu^0\overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i)\left(\frac{\epsilon^0}{\epsilon^i}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{r}_i)\right) \mathrm{d}S' \\ - ik_0 Z_0 \int_{\partial\Omega} \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(-\mathrm{i}\omega\nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i)\left(\frac{\epsilon^0}{\epsilon^i}\right)\mathbf{E}^{\mathrm{in}}(\mathbf{r}_i)\right) \mathrm{d}S'\Bigg)\Bigg],\end{aligned} \tag{9.2.5}$$

where V_i is the volume $\alpha^3|S^i|$. Carrying out the differentiation, we have

$$\begin{aligned}\frac{\partial u_r}{\partial m_i} = -\frac{\epsilon_0 V_i \epsilon^0}{(\epsilon^0 + (\epsilon^i - \epsilon^0)/3)^2}\mathbf{p}_r \cdot \Bigg(\int_{\partial\Omega} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(\omega^2\mu^0\overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i)\mathbf{E}^{\mathrm{in}}(\mathbf{r}_i)\right) \mathrm{d}S' \\ - ik_0 Z_0 \int_{\partial\Omega} \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(-i\omega\nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i)\mathbf{E}^{\mathrm{in}}(\mathbf{r}_i)\right) \mathrm{d}S'\Bigg).\end{aligned} \tag{9.2.6}$$

Equation (9.2.6) gives an approximation to the Gauss-Newton Hessian terms, which are relatively cheap to calculate (carrying out the integrals numerically), involving only multiplications and not the p additional forward problems required to calculate J. One can directly can directly calculate and store only the diagonals of B required, alleviating the overhead of having to calculate and store (at least briefly) entire columns of J.

Another computational benefit of (9.2.6) is that the data illumination part of B (the integrals) and the non-linear saturation part (the contrast terms) are separated. One may choose to store only the data illumination part, calculating the saturation $(\epsilon^0)^2/(\epsilon^0 + (\epsilon^i - \epsilon^0)/3)^2)$ on the fly for individual parameters m_i, which allows us to efficiently calculate approximations to $B(\mathbf{m}_0 + \alpha\mathbf{p})$ rather than just $B(\mathbf{m}_0)$. This relies on our assumption that our subsurface is discretised into volumes which have a sphere as a best fitting ellipsoid. If they do not, the polarisation tensor is not a multiple of the identity, they do not factorise out of the integrals, and so the integration would have to be carried for each proposed change $\mathbf{p}$. While this does not lead to an efficient algorithm, it may add information about our discretisation strategy on inversion, for example how a discretisation into small thin volumes can have a somewhat anisotropic effect on the forward problem (and therefore reconstruction).

Despite the polarisation tensor expansion assuming inclusions are sufficiently separated (i.e. it is a single scattering approximation), and this not being the case in our discretised domain, it is still both a useful approximation here. When inclusions are sufficiently close, the approximation fails by not accounting for multiple scattering. However, the Gauss-Newton approximation of the Hessian is also a single-scattering assumption (when calculated for a homogeneous background), and so we have no worse an approximation.

We can also calculate the second order derivatives $\frac{\partial^2 u_r}{\partial m_i^2}$

$$\begin{aligned}\frac{\partial^2 u_r}{\partial m_i^2} =& \frac{2\epsilon_0^2 V_i \epsilon^0}{3(\epsilon^0 + (\epsilon^i - \epsilon^0)/3)^3}\mathbf{p}_r \cdot \left(\int_{\partial\Omega} \nabla \times \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(\omega^2 \mu^0 \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i)\mathbf{E}^{\mathrm{in}}(\mathbf{r}_i)\right) \,\mathrm{d}S' \right. \\ & \left. - \mathrm{i}k_0 Z_0 \int_{\partial\Omega} \overline{\overline{\mathrm{G}}}(\mathbf{r}_r, \mathbf{r}') \cdot \left(-\mathrm{i}\omega \nabla' \times \overline{\overline{\mathrm{G}}}(\mathbf{r}', \mathbf{r}_i)\mathbf{E}^{\mathrm{in}}(\mathbf{r}_i)\right) \,\mathrm{d}S' \right),\end{aligned} \tag{9.2.7}$$

but cannot calculate the mixed derivative terms. The diagonal terms of this part of the Hessian matrix are given by

$$C_{ii}(\mathbf{m}_0 + \delta\mathbf{m}) = \Re\left\{ \sum_r \left.\frac{\partial^2 \mathcal{F}}{\partial m_i^2}\right|_{\mathbf{m}_0+\delta\mathbf{m}} \delta d_r(\mathbf{m}_0 + \delta\mathbf{m}) \right\} \approx \Re\left\{ \sum_r \left.\frac{\partial^2 u_r}{\partial m_i^2}\right|_{\delta\mathbf{m}} \delta d_{0,r}(\delta\mathbf{m}) \right\}, \tag{9.2.8}$$

which requires $\delta\mathbf{d}(\delta\mathbf{m})$. We do not then have the same efficient calculation of $C_{ii}(\mathbf{m}_0 + \delta\mathbf{m})$ as for $B_{ij}(\mathbf{m}_0 + \delta\mathbf{m})$, but it is straightforward to calculate $C_{ii}(\mathbf{m}_0)$

9.3 Remarks on the approximate Hessian

In the previous section, we derived an expression for an approximate Hessian matrix $H(\mathbf{p})$. The matrix includes both compensation for data illumination given by $\mathrm{diag}(J^T J^*)(\mathbf{0})$, but also includes the saturation effect for a proposed model change $\delta\mathbf{m}$. When only the diagonal Gauss Newton terms $(\partial u_r/\partial m_i)^2$ are calculated, the matrix is positive definite and so any $H^{-1}\mathbf{p}$ is a descent direction. This is not guaranteed if we include the $\partial^2 u_r/\partial m_i^2$ terms which may be negative. As the size of parameter volume tends to zero, then $H(\mathbf{0}) \to \mathrm{diag}(J^T J^*)$, where $H(\mathbf{0})$ is the approximate Hessian including only the $(\partial u_r/\partial m_i)^2$ terms.

Since we have an approximation to $H(\alpha\mathbf{p})$, and not just $H(\mathbf{0})$, we would like to incorporate this additional information into a pre-conditioning matrix/initial Hessian to begin an l-BFGS optimisation scheme. Performing a line search along $\mathbf{p}_0 = -H(\mathbf{0})^{-1}\mathbf{g}$ to precondition l-BFGS neglects information about the saturation effect. Consider

Taylor's theorem,

$$\mathcal{J}(\mathbf{m}+\mathbf{p}) = \mathcal{J}(\mathbf{m}) + \mathbf{p}^T\nabla\mathcal{J}(\mathbf{m}_k) + \frac{1}{2}\mathbf{p}^T\nabla^2\mathcal{J}(\mathbf{m}+t\mathbf{p})\mathbf{p}, \tag{9.3.1}$$

where $\mathcal{J}$ is the cost function, for some $t \in (0,1)$. From (9.3.1), to incorporate the saturation effect, we might consider an update of the form

$$\alpha\mathbf{p} = -\alpha H(\alpha\mathbf{p})^{-1}\mathbf{g}, \tag{9.3.2}$$

which would be an implicit scheme. Making use of the implicit function theorem, we can show that the curve $\phi(\alpha)$,

$$\phi(\alpha) := \mathcal{J}(\mathbf{m}_0 + \alpha\mathbf{p}_\alpha), \qquad \mathbf{p}_\alpha = -H^{-1}(\alpha\mathbf{p})\mathbf{g} \tag{9.3.3}$$

is continuous and has a continuous first derivative for $(\mathbf{m}_0 + \mathbf{p}) \in \mathbb{R}^p$. Consider the function

$$\mathbf{f}(x,\mathbf{y}) = H^{-1}(x\mathbf{y})\mathbf{g} - \mathbf{y}, \qquad \mathbf{f} : \mathbb{R}^{1+p} \to \mathbb{R}^p. \tag{9.3.4}$$

Denote by $D\mathbf{f}(\alpha, \mathbf{p}_\alpha) = [X|Y]$ the Jacobian matrix X the derivatives $\dfrac{\partial f_i}{\partial x}$ and Y the matrix of derivatives $\dfrac{\partial f_i}{\partial y_j}$. For a point $(\alpha, \mathbf{p}_\alpha)$ for which $\mathbf{f}(\alpha, \mathbf{p}_\alpha) = \mathbf{0}$, the implicit function theorem states that if Y is invertible, then there exists an open set U containing α and an open set V containing $\mathbf{b}$ and a unique continuously differentiable function $\mathbf{g}(x) : U \to V$ such that

$$\{(x, \mathbf{g}(x)) | x \in U\} = \{(x, \mathbf{y}) \in U \times V | \mathbf{f}(x,\mathbf{y}) = \mathbf{0}\}. \tag{9.3.5}$$

Since $\mathbf{f}$ is continuously differentiable (by the definition of H), it remains only to show that Y is invertible and that solutions exist. By the definition of H, Y will be a full-rank diagonal matrix. Finally, we can begin at $\alpha = 0$, $\mathbf{p} = \mathbf{0}$, and by the implicit function theorem $\mathbf{p}_\alpha$ is defined implicitly by α in an open neighbourhood of $\alpha = 0$, and is continuously differentiable. The composition of continuous functions is also continuous, so $\phi(\alpha)$ is continuous, with continuous first derivative, in an open neighbourhood of $\alpha = 0$.

However, while $\phi(\alpha)$ defined by (9.3.2) is continuous, it is not clear how best to search along this curve. Particularly, it may only be well defined in too small a neighbourhood of $\alpha = 0$, and it is not clear what stopping criteria to use for such a 'line search' (the Wolfe conditions would need to be modified). Instead, it is more appropriate to incorporate the additional information $H(\delta\mathbf{m})$ in a trust-region type algorithm, which we present in the next section.

9.4 A Trust-region algorithm with the approximate Hessian

A trust-region method first finds the minimum of the model (9.4.2), before checking the step which minimises the model provides an adequate reduction in the true cost function. The contrast to a line search method (such as lBFGS) is that the model is used only to find a descent direction, and the true cost function used to find a minimum along this direction.

To determine a model, consider again Taylor's theorem

$$\mathcal{J}(\mathbf{m}_0 + \mathbf{p}) = \mathcal{J}(\mathbf{m}_0) + \mathbf{p}^T \nabla \mathcal{J}(\mathbf{m}_0) + \frac{1}{2}\int_0^1 (1-t)\mathbf{p}^T \nabla^2 \mathcal{J}(\mathbf{m}_0 + t\mathbf{p})\mathbf{p}\ \mathrm{d}t. \tag{9.4.1}$$

Based on the (9.4.1), we propose the model

$$\mathcal{M}(\mathbf{p}) := \mathcal{J}(\mathbf{m}_0) + \nabla \mathcal{J}(\mathbf{m}_0)^T \mathbf{p} + \frac{1}{2}\int_0^1 (1-t)\mathbf{p}^T H(t\mathbf{p})\mathbf{p}\ \mathrm{d}t, \tag{9.4.2}$$

for use in a Trust region method, in which the integral can be evaluated numerically, together with the search direction

$$\mathbf{p} = -H^{-1}(\mathbf{0})\mathbf{g}. \tag{9.4.3}$$

Such a choice includes the saturation affect in the cost evaluation, so the proposed minima along $\mathbf{p}$, estimated by (9.4.2), may be closer to the true minimum than basing the model on (9.3.1).

Note that (9.4.2) is cheap to evaluate: for length($\mathbf{p}$) $= p$, and evaluating the integral numerically at k integration points, the matrix operations involve only $(2k+1)p$ multiplications for a diagonal H^{-1}. Because of this low cost, we propose evaluating $\mathcal{M}(\mathbf{x})$ at the points

$$\mathbf{x}(\alpha_i) = \alpha_i \mathbf{p}, \qquad \mathbf{p} = H^{-1}(\mathbf{0})\mathbf{g}, \tag{9.4.4}$$

for some fixed $s \in [0,1]$ and $0 \leq \alpha_i \leq \alpha_{\max}$, where $\alpha_{\max}$ is a predetermined trust limit of the model (9.4.2). We may then interpolate between these points to find the α which minimises (9.4.2) along $\phi(\alpha) = \mathcal{M}(\alpha\mathbf{p})$, returning both a parameter update

$$\mathbf{m}_1 = \mathbf{m}_0 + \alpha\mathbf{p},$$

and an approximate Hessian matrix $H(\alpha\mathbf{p})$ to precondition a subsequent l-BFGS optimisation scheme.

We must ensure the parameter update $\alpha\mathbf{p}$ has reduced the true cost function $\mathcal{J}$, and that there was sufficient agreement between the true and model function reductions. This is done by calculating

$$\rho = \frac{\mathcal{J}(\mathbf{m}_0) - \mathcal{J}(\mathbf{m}_1)}{\mathcal{M}(0) - \mathcal{M}(\alpha)}. \tag{9.4.5}$$

If the result of (9.4.5) is negative, an increase in cost was found. If ρ is too small, less than some $\eta \in [0, \frac{1}{4})$ say, then the agreement between the model and cost function was poor, in which case we would decide to reduce $\alpha_{\max}$ and recalculate the constrained minimum. This also ensures that the Hessian approximation $H(\alpha\mathbf{p})$ is a good approximation to the true Hessian in the norm

$$||H||_{\mathbf{p}} = |\mathbf{p}^T H \mathbf{p}|. \tag{9.4.6}$$

Since we only calculate the leading diagonals of H, neglecting large parts of the true Hessian, we generally take a tolerance close to 0, $0 < \eta \ll 0.25$.

The trust region scheme proposed is given in Algorithm 7, based on the basic trust-region approach given by Nocedal and Wright [123, pp 68-69], and MATLAB code given in Appendix B.2 [194]. An initial minimiser α is first found via interpolation of model function values. If this fails to sufficiently reduce the objective function, $\alpha_{\max}$ is reduced, interpolation points greater than $\alpha_{\max}$ and corresponding model values are removed, which are replaced by the pair $\{\alpha, \mathcal{J}(\mathbf{m}_0 + \alpha\mathbf{p})\}$, and a new constrained minimum is found. The algorithm returns a new initial model $\mathbf{m}_1$ and approximate Hessian H to precondition an l-BFGS scheme. Optionally, the initial Hessian approximation could include a single BFGS update, provided this is positive definite (this is not guaranteed by Algorithm 7).

The reduction of $\alpha_{\max}$ is safeguarded to ensure sufficient decrease of the trial step length α, such that $\alpha_{\max} \leq \lambda\alpha$ for some $\lambda \in (0, 1)$. Further, the removal of estimated interpolation values satisfying

$$\{\alpha_i, c_i\} : |\alpha_i - \alpha| < \beta e^{\eta-\rho}\alpha, \qquad 0 < \beta < 0.5,$$

where $c_i = \mathcal{J}(\mathbf{m}_0 + \alpha_i\mathbf{p})$, rejects more estimated interpolation values the smaller (and more negative) ρ is compared with η (and therefore the worse the model $\mathcal{M}$). Specifically, the exponential value is large when ρ is negative – that is, when the function increased in value, and so the estimated function values are particularly poor.

We can view the algorithm as a backtracking line search, which will satisfy (approximately) the sufficient decrease condition. The estimated values are given by

$$\mathcal{M}(\alpha) = \mathcal{J}(\mathbf{m}_0) + \nabla\mathcal{J}(\mathbf{m}_0)^T\mathbf{p} + \mathcal{O}(|\mathbf{p}|^2). \tag{9.4.7}$$

Then the sufficient agreement condition (9.4.5) can be written as

$$\begin{aligned} \eta \leq & \frac{\mathcal{J}(\mathbf{m}_0) - \mathbf{J}(\mathbf{m}_0 + \alpha\mathbf{p})}{\mathcal{M}(0) - \mathcal{M}(\alpha)} \\ \approx & \frac{\mathcal{J}(\mathbf{m}_0) - \mathbf{J}(\mathbf{m}_0 + \alpha\mathbf{p})}{-\alpha\nabla\mathcal{J}(\mathbf{m}_0)^T\mathbf{p}}, \end{aligned}$$

and so we ensure

$$\mathcal{J}(\mathbf{m}_0 + \alpha\mathbf{p}) \leq \mathcal{J}(\mathbf{m}_0) + \eta\alpha\nabla\mathcal{J}(\mathbf{m}_0)^T\mathbf{p}, \tag{9.4.8}$$

which is the sufficient decrease condition for $\eta = c_1$. The contribution of the Hessian, when included in (9.4.8), is to reduce the amount of 'sufficient decrease' required at longer step lengths.

Algorithm 7 Trust-region initial Hessian

```
Input $\lambda \in [0,1)$, $\alpha_{\max} > 0$, $0 < \alpha_1 < \ldots < \alpha_n = \alpha_{\max}$, $\eta \in [0, 0.25)$, $\beta \in (0, 0.5)$;
Calculate terms $S_i = \left(\sum_r \frac{\partial u_r}{\partial m_i}\right)^2$ via (9.2.6);
$H \leftarrow \mathrm{diag}(S_i) + \lambda I$;                    ▷ Store $H_{ii}$ as coefficients of $\delta m_i$
$c_0 \leftarrow \mathcal{J}(m_0)$, $g \leftarrow \nabla\mathcal{J}(m_0)$;
$p \leftarrow H^{-1}(0)g$;
for $i = 1 \ldots n$ do
    $H_i \leftarrow H(m_0 + \alpha_i p)$;
    $c_i \leftarrow c_0 + \alpha_i g^T p + \frac{1}{2}\sum_{j=0}^{i} w_{ij}(1 - \frac{j}{i})\alpha_i^2 p^T H_j^{-1} p$;    ▷ Evaluate (9.4.2)
end for
Interpolate $c_i$ to find approximate minimiser $\alpha$ of $\mathcal{M}$;
$\rho \leftarrow (\mathcal{J}(m_0) - \mathcal{J}(m_0 + \alpha p))/(\mathcal{M}(0) - \mathcal{M}(\alpha))$;
while $\rho < \eta$ do
    Reduce $\alpha_{\max}$;
    Remove $\alpha_i : |\alpha_i - \alpha| < \beta e^{\eta-\rho}\alpha$ and corresponding estimated $c_i$;
    Remove $\alpha_i > \alpha_{\max}$ and corresponding estimated $c_i$;
    Insert interpolation point and value ($\alpha$, $\mathcal{J}(m_0 + \alpha p)$) to ($\alpha_i$, $c_i$);
    Interpolate $c_i$ to find approximate minimiser $\alpha$ of $\mathcal{M}$;
    $\rho \leftarrow \big(\mathcal{J}(m_0) - \mathcal{J}(m_0 + \alpha p)\big)/(\mathcal{M}(0) - \mathcal{M}(\alpha))$;
end while
$H_i \leftarrow H(m_0 + \alpha_i p)$;
Return Trial model update $m_1$, approximate initial Hessian $H(m_1)$;
```

9.5 Numerical experiment

9.5.1 Introduction

To demonstrate the applicability of the approximate Hessian (9.2.6) and Algorithm 7, our aims are twofold. Firstly, we wish to show that the descent direction

$$\mathbf{p}_0 = -H^{-1}\mathbf{g}_0 \tag{9.5.1}$$

is well scaled, and that the initial Hessian H improves the rate of convergence of the l-BFGS scheme at least in early iterations. Secondly, we wish to show that Algorithm 7 can efficiently find a suitable step length α. As noted by Leong and Chen [97], it is hard to interpret the influence of the initial Hessian matrix, with a full study needing to take into account line search policies, the choice of memory parameter m, as well as a range of test problems and the structure and spectrum of their (true) Hessian matrices.

Due to the computational cost of FWI in 3D we are limited as to the scope of any numerical study, and so in this section we will consider the effect of incorporating the approximate Hessian to the scenario used in Section 8.4.5, of a single scatterer in a highly inhomogeneous background. Specifically, we will consider the computational cost of finding an initial step length via Algorithm 7 and subsequent convergence of the first 10 iterations of l-BFGS, comparing to initialising l-BFGS with the identity matrix.

9.5.2 Numerical results

Verifying the trust-region algorithm

We begin by evaluating the performance of the trust region search, Algorithm 7, in resolving the single object in a stochastic background medium as used in Section 8.4.5, shown in Figure 8.8. We do not separate the descent direction into target and nuisance parameters, as in Section 8.4. As such, we can expect the initial descent direction to be poor due to scattering in the highly stochastic background medium.

Figure 9.1(a) shows the objective values estimated by the model (9.4.2) (in blue), and compares these values to the values estimated by the second order Taylor expansion,

$$\mathcal{M}'(\mathbf{p}) = \mathcal{J}(\mathbf{m}_0) + \nabla\mathcal{J}(\mathbf{m}_0)^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T H(\mathbf{m}_0)\mathbf{p}. \tag{9.5.2}$$

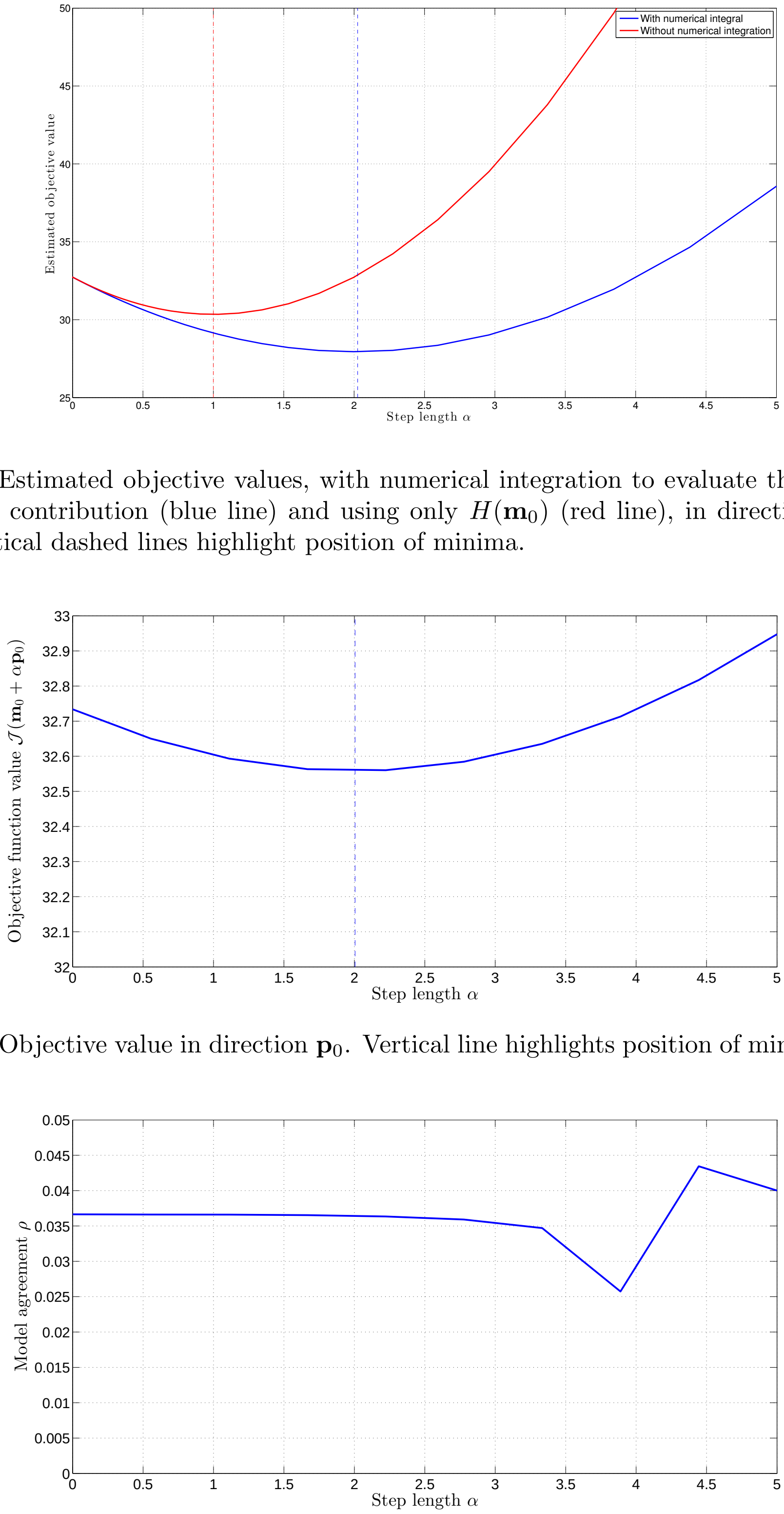
We see the contribution of the numerical integration of the contribution from the Hessian approximation $H(\mathbf{m}_0 + \mathbf{p})$ is significant: the position of the minimum is approximately doubled compared with use of $H(\mathbf{m}_0)$ alone. More importantly, we see the minimum of the model function $\mathcal{M}$ is close to the true minimum position – a relative error of 0.25%. On comparison to the true objective values in Figure 9.1(b), we see an over-estimation of the reduction in objective value by the model. We expect this is largely due to a particularly poor descent direction due to the multiple scattering in the stochastic background medium, which is not accounted for due to the lack of off-diagonal terms in the approximate Hessian.

Figure 9.1(c) shows that the model agreement is consistent at least up until the point of the estimated minimum (larger step lengths would not be chosen by the trust region algorithm). We can see first step length suggested by Algorithm 7 will satisfy the curvature condition as it is close to the true minimum, and will satisfy the sufficient decrease condition for sufficiently small c_1.

The effect of pre-conditioning

We will refer to l-BFGS initialised by the identity matrix as l-BFGS(I), and l-BFGS pre-conditioned by the approximate Hessian matrix as l-BFGS(H). To compare the convergence of l-BFGS(I) to l-BFGS(H), we begin by comparing their convergence in the inversion of a single low-contrast spherical scatterer of $\epsilon_r = 3.05$ in a homogeneous background of $\epsilon_r = 3$ (with the same dimensions and layout as shown in Figure 8.1(a) but different permittivity). In such a domain the linear approximation to the scattering problem should be a good one. This is then the easiest problem we can construct for l-BFGS(I), as the gradient direction should be close to the true perturbation.

Figure 9.2 compares the convergence of l-BFGS(I) and l-BFGS(H), showing the relative residuals, absolute 2-norm error to the true solution, step-length and forward/gradient calculations needed (above the required 2) at each iteration. We note that the first iteration of l-BFGS(I) increased the true error, while decreasing the residual, likely due to the lack of information about data-coverage in this steepest descent



(a) Estimated objective values, with numerical integration to evaluate the Hessian contribution (blue line) and using only $H(\mathbf{m}_0)$ (red line), in direction $\mathbf{p}_0$. Vertical dashed lines highlight position of minima.

(b) Objective value in direction $\mathbf{p}_0$. Vertical line highlights position of minimum.

(c) Agreement ratio between true and estimated objective values.

Figure 9.1: Comparison of objective values (a) estimated by Algorithm 7 and (b) evaluating the cost function, as well as the positions of their minima, and (c) their agreement ratio, for a single object in a stochastic background medium.

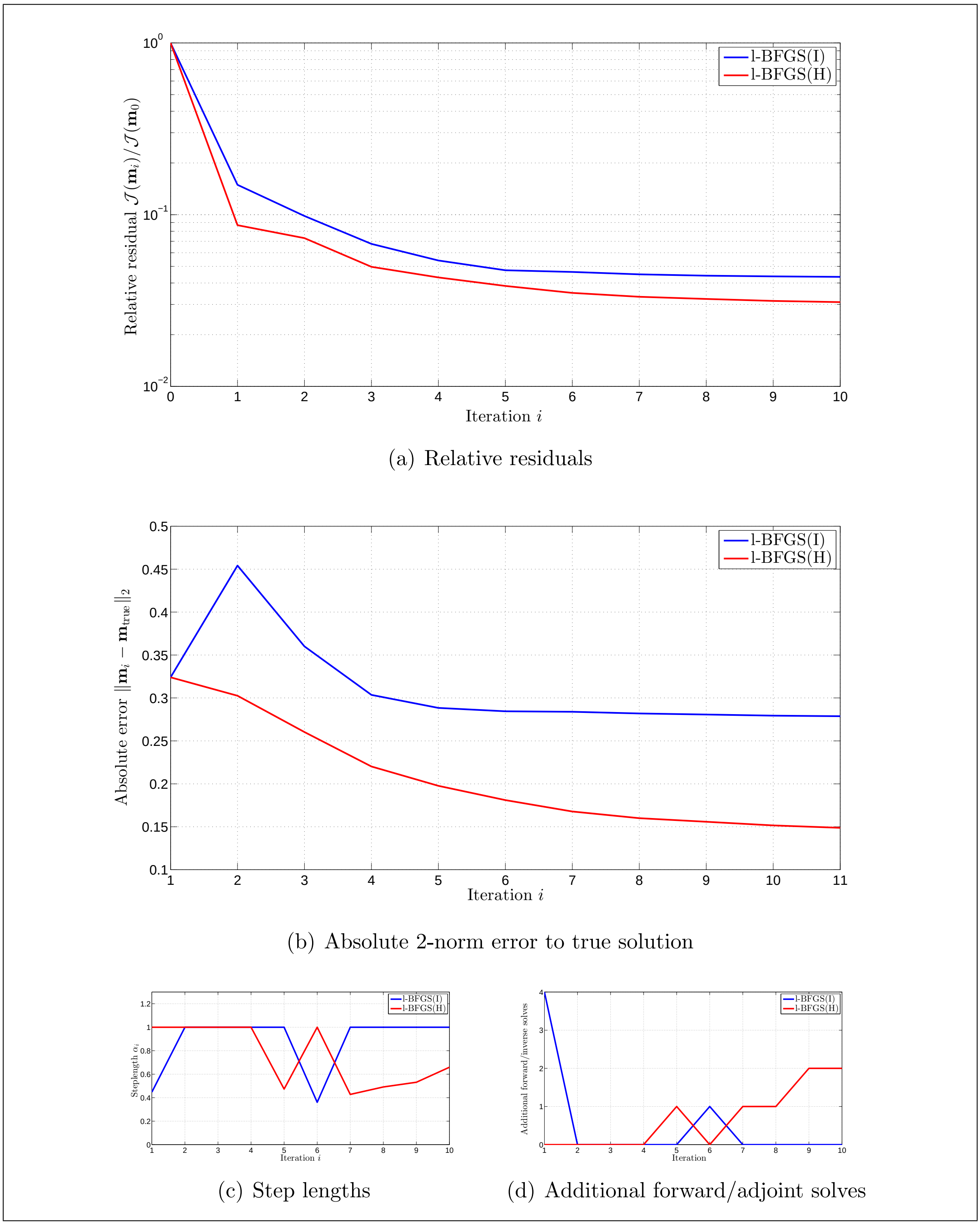
iteration. It is clear that even in this simple test problem, l-BFGS(I) takes several iterations before it is able to move towards the true solution. Contrarily, l-BFGS(H) makes good progress in each iteration, and until iteration 5 requires no additional forward solves to find a suitable step-length. While it appears less efficient after the 5^{th} iteration, requiring more forward/gradient calculations, at this point the residual is already less than l-BFGS(I) reaches after 10 iterations. We expect l-BFGS(H) performs particularly well here because, in this single target domain, the polarization tensor approximation for the scattered fields (which is a single scattering approximation) sufficiently accurate.

We also consider the effect of the preconditioner for reconstructing the target in stochastic domain, as shown in Figure 8.8. Again, we do not use the nuisance parameter search given in Algorithm 6. Figure 9.4 shows the relative residuals of l-BFGS(I) and l-BFGS(H), Figure 9.4 the reconstructed image slices through $y = 0$, and Figure 9.5 reconstruction horizontal and vertical slices at $y = 0$, $z = 0.015$ and $x = y = 0.01$, respectively. Interestingly, we see little or no gain from preconditioning in this example. There was also no improvement in computational cost (nor any loss), and so graphs of step-lengths and forward calculations needed are omitted.

The only benefit we see from the reconstruction slices is that l-BFGS(H) is slightly better able to determine the shape of the target. The lack of improvement is likely due to the complexity of this problem involving multiple scattering, which the Hessian approximation does not account for. The improved ability to reconstruct the edges of the target is also likely due to accounting for data coverage at these locations.

9.6 Towards realistic inversion

Throughout these last two chapters, the numerical results presented have been limited by the computational cost and memory requirements of 3D electromagnetic FWI. We present in this section, for completeness, a reconstruction of a more realistic scenario than previously used for comparison and verification. The domain contains two 4 cm cubic targets, one higher and one lower permittivity than the background medium, offset from each other in x, y and z, but close enough that we expect multiple scattering to be noticeable. These are placed in a stochastic background medium, as shown in



(a) Relative residuals

(b) Absolute 2-norm error to true solution

(c) Step lengths

(d) Additional forward/adjoint solves

Figure 9.2: Comparison of convergence for preconditioned and standard l-BFGS, l-BFGS(H) (red) and l-BFGS(I) (blue) respectively, for reconstruction of a low contrast scatterer in homogeneous background. (a) shows the relative residuals, (b) the absolute 2-norm error to the true solution, (c) the step-length found to satisfy the strong Wolfe conditions and (d) the additional computations required to find the step-length (above the necessary 1 forward and 1 gradient calculation).

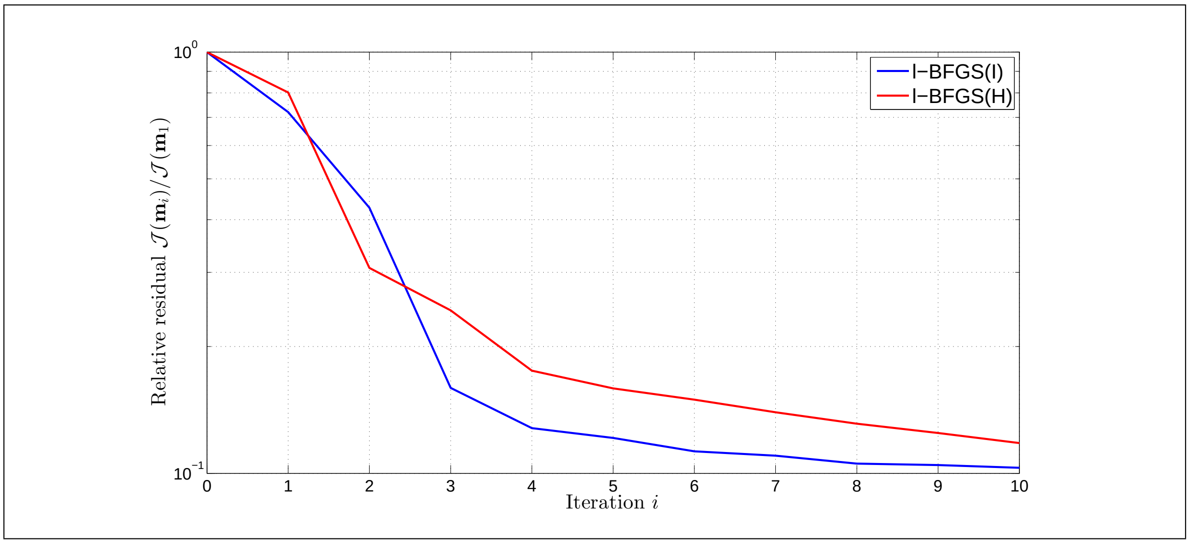


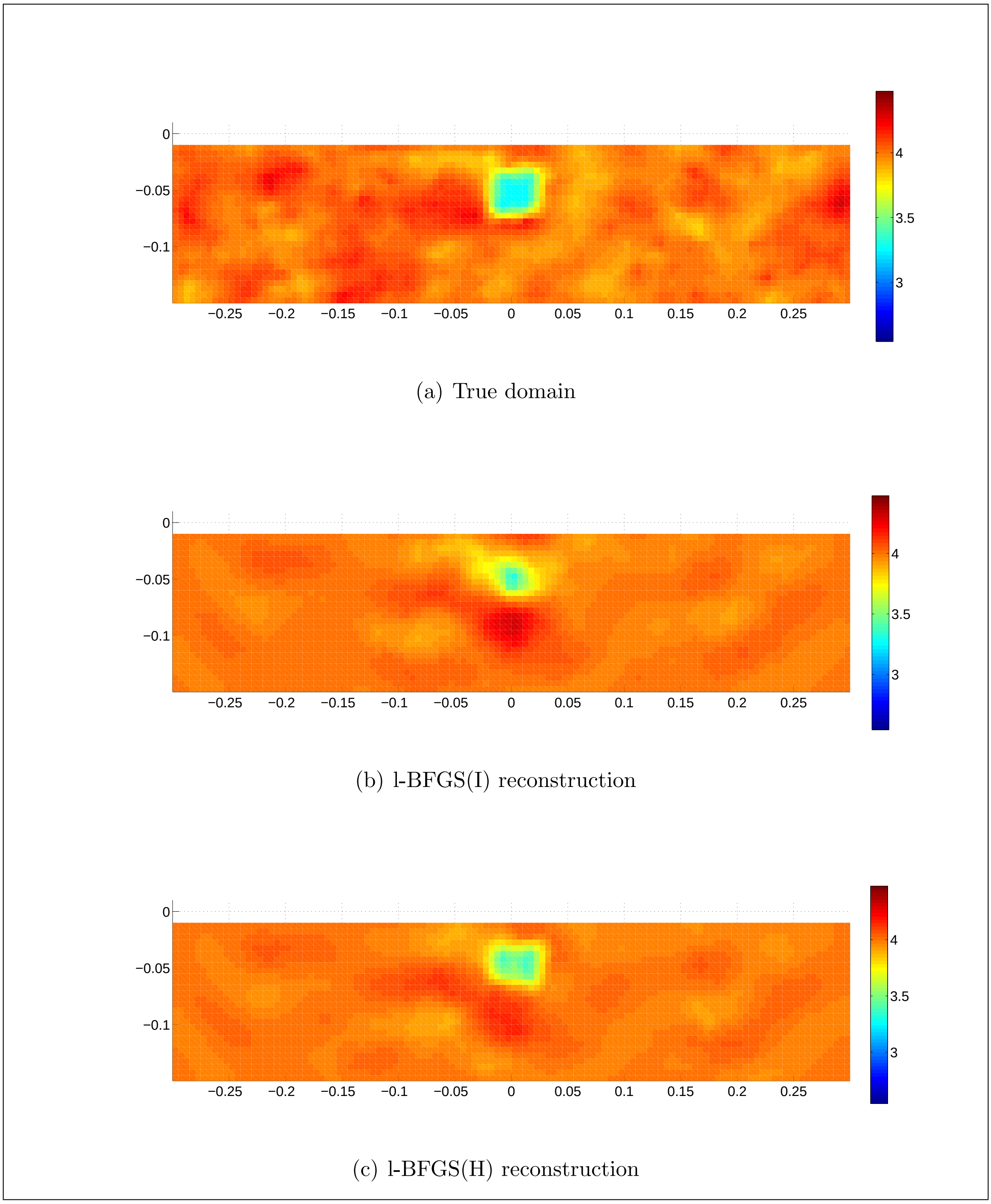
Figure 9.3: Relative residuals for l-BFGS(I) (blue) and l-BFGS(H) (red) for reconstructing a cubic target in a stochastic background medium

Figure 9.6(a). We simulate data at 20 frequencies between 1 and 1.5 GHz, at 180 source locations over $[-0.15, -0.15] \times [0.15, 0.15]$ with a 4×2 array of antennas arranged with polarizations

$$\begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{x}} & \hat{\mathbf{y}} \\ \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{x}} & \hat{\mathbf{y}} \end{bmatrix}.$$

The top-right and bottom-left antennae alternate as source terms, and 5% Gaussian white noise is added to the data. We designate the object at $(0, 0, 0.05)$ as the target of interest (e.g. in the location of a metal fragment), setting the ROI as the 4 cm radius region about this point. As before, we expect the stochastic medium to make resolving the target more difficult, but this should be made more so by the presence of the second target close enough to obscure direct reflections from some source locations. We should also have poor information about the second target as part of it lies outside the footprint of source locations.

The reconstructed results are shown in Figure 9.6, and we see we are successfully able to reconstruct the approximate shape and permittivity of both objects, as well as some of the more dominant features of the stochastic background medium. Interestingly, some parts of the central target are resolved with a higher contrast than the background medium, though the sides of the objects are fairly well captured. The plots of relative residuals, norm of the gradient and absolute error in Figure 9.7 show a steady decay (approximately linear for the residuals and gradient), and by the size of the gradient after 20 iterations we must assume we are close to a (local) minimum.

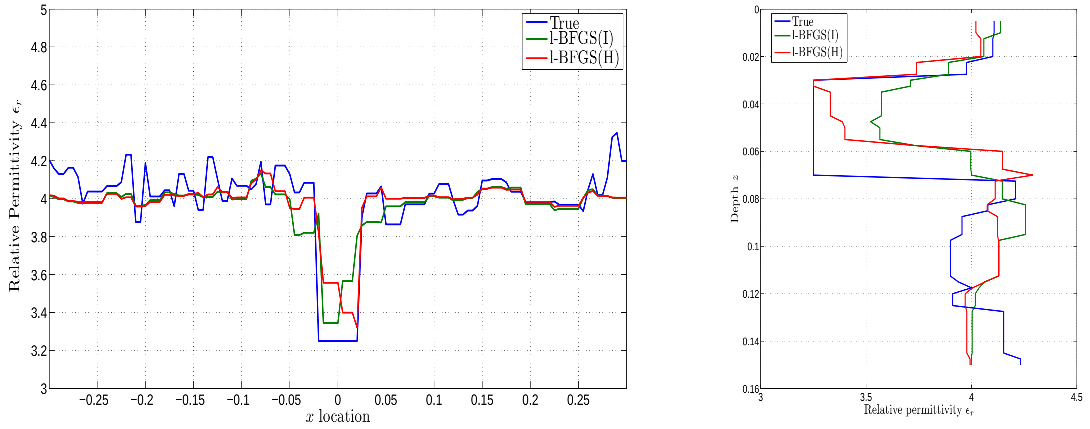


(a) True domain

(b) l-BFGS(I) reconstruction

(c) l-BFGS(H) reconstruction

Figure 9.4: Comparison of reconstructions of a single target in a stochastic background medium. (a) shows the true image at $y = 0$, (b) the l-BFGS(I) reconstruction, and (c) the l-BFGS(H) reconstruction.



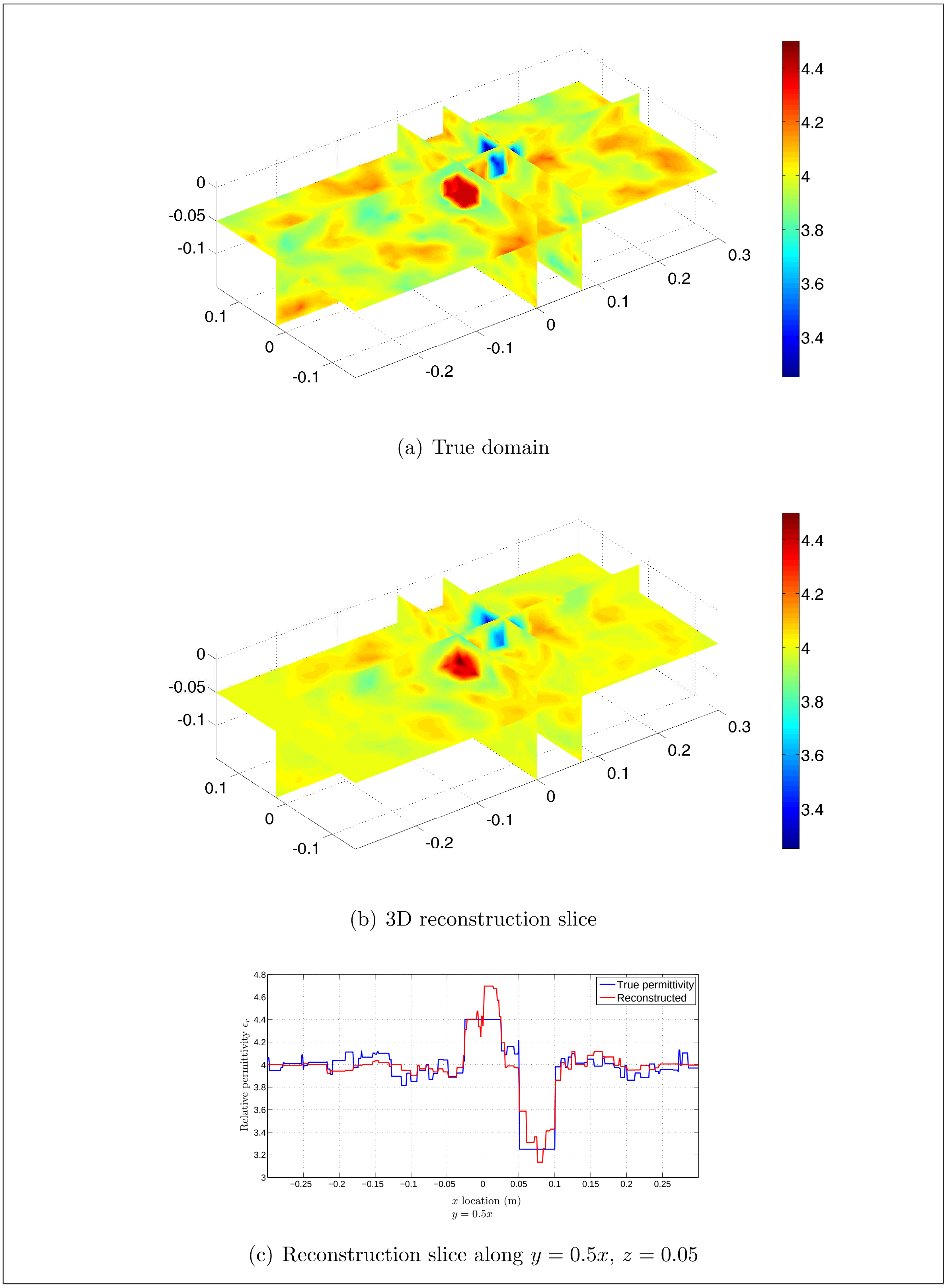
(a) Horizontal slice at $y = 0$, $z = 0.05$

(b) Depth slice at $x = y = 0.01$

Figure 9.5: Comparison of reconstructions of a single target in a stochastic background medium. (a) shows a horizontal slice at $y = 0$, $z = 0.05$, and (b) shows a depth slice at $x = y = 0.01$. Blues shows the true domain, green l-BFGS(I) and red l-BFGS(H).

Figure 9.7(d) also shows the step lengths selected for target and nuisance parameters are fairly independent, as desired.

We simply echo some previous conclusions here: FWI in 3D with a small-scale array of surface measurements is a plausible method for resolving a detected target in an inhomogeneous background; the implementation of TV regularisation is likely insufficient in the 3D setting; and Algorithms 6 and 7 appear to work successfully in the more complex medium. There is nothing to suggest near-surface FWI as a tool to resolve unknown targets is not possible, even when we move towards more realistic settings. A wider range of ground types and scenarios should now be observed.

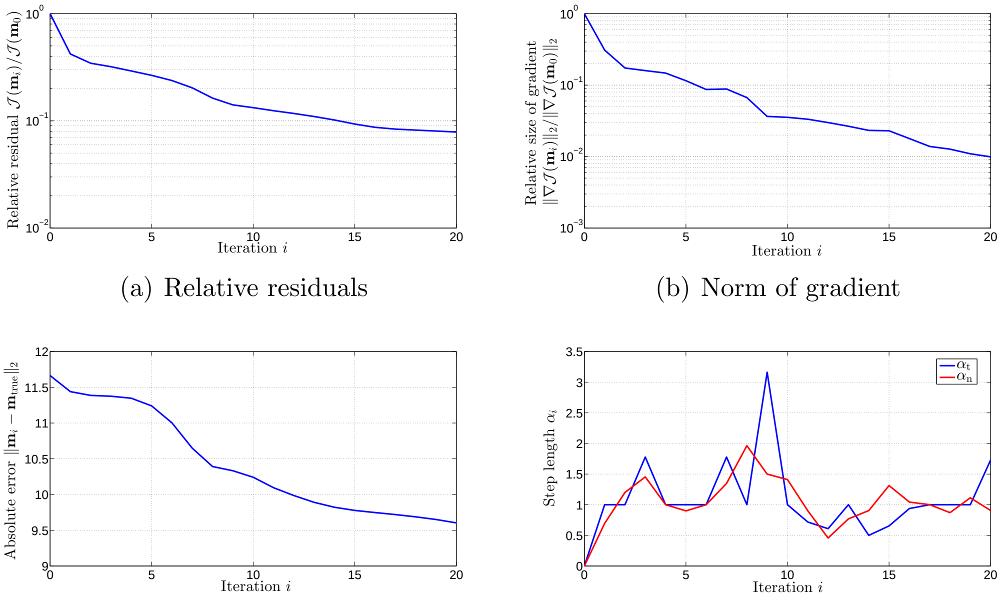


(a) True domain

(b) 3D reconstruction slice

(c) Reconstruction slice along $y = 0.5x$, $z = 0.05$

Figure 9.6: Reconstruction of two targets in a stochastic medium, after 20 iterations of l-BFGS(H) and using the line search Algorithm 6. (a) shows the true domain containing two cubic targets with relative permittivity $\epsilon_r = 4.4$ and 3.25, respectively, in a stochastic background medium with mean $\epsilon_r = 4$. (b) shows the 3D reconstruction slice, and (c) a slice along $y = 0.5x$, $z = 0.05$.



(a) Relative residuals

(b) Norm of gradient

(c) Absolute error

(d) Step lengths

Figure 9.7: (a) Relative residuals, (b) relative norm of gradient, (c) absolute error and (d) target and nuisance parameter step lengths for 20 iterations of l-BFGS reconstructing domain shown in Figure 9.6

9.7 Summary and conclusions

In this chapter, we have used the polarization tensor expansion of Chapter 7 to derive approximations to components of the Hessian matrix of the least squares data-misfit objective function. This extends a commonly used approximation to the Hessian diagonal components for FWI of scalar-valued waves to include polarization and ground reflection terms. It is also able to approximate the change in these terms as the parameters m_i vary, and this approximation holds provided nearby parameters remain unchanged – i.e. when the polarization tensor approximation holds. Since it does not account for multiple scattering, the approximation $H(\mathbf{m}_0 + \delta\mathbf{m})$ may not be good where $\delta\mathbf{m}$ contains nearby perturbations. However, the Gauss-Newton approximation which is often used to precondition l-BFGS also breaks down under this condition.

A computational benefit of the approximate Hessian is that the change in components due to a perturbation in parameters $\delta\mathbf{m}$ is computationally cheap to calculate (involving only multiplication) for parameters m_i that are (assumed to be) of equal length in each coordinate direction. However, where they are not the approximate Hessian also incorporates the effect of scattering from a non-uniformly shaped parameter, particularly on the change in polarization of the electromagnetic wave. This is not accounted for in the usual approximation involving the Green's function (for scalar valued waves), which assumes point reflectors.

Rather than use the Hessian approximation directly to pre-condition l-BFGS, we considered its ability to estimate the changes in Hessian terms could be best utilised via a trust-region type method. The model used for the trust region was the 1st order Taylor expansion, with the exact remainder term in integral form being approximated by numerical integration of the Hessian approximation $H(\mathbf{m}_0 + \alpha \mathbf{p}_0)$. For a descent direction, we took the approximate Newton direction $\mathbf{p}_0 = -H^{-1}(\mathbf{m}_0)\mathbf{g}_0$. In the numerical examples we have tried, the trust region algorithm generally over-estimated the reduction in objective function, but chose a steplength close to the exact minimiser. Where the dominant feature of the data was due to a single target, subsequent iterations of l-BFGS created well-scaled descent directions, and convergence was improved over improved over initialising with the identity matrix. Where there was significant multiple scattering, the benefits of the Hessian approximation were less pronounced, though it did lead to a marginally better approximation of the target's shape for very little additional computational cost.

Not only does the this Hessian approximation contribute to the theory of FWI for GPR data, but this novel approach could be used for any (non-linear) inverse problem for which there is a similar polarization tensor expansion describing the. It remains to more thoroughly verify the approximate Hessian, both through testing convergence of the to true Hessian components as the size of parameter tends to zero (if computationally viable), and to numerically verify the effect of the pre-conditioner on a wider range of inversion problems. While we have tested the Hessian approximation for a single scatterer in a highly inhomogeneous domain, it would be interesting to determine how well the approximation helps convergence for a domain containing a few, distinct, nearby targets. It is likely to be less effective in such a situation, although including some off-diagonal terms could improve its practicality. Doing so would be straightforward, but increase the memory footprint of the matrix. Regardless, we strongly expect it will improve convergence of l-BFGS in a wide range of problems when compared to initialising with the identity matrix.

Chapter 10

Conclusions and Future Research

10.1 Summary and conclusions

This thesis has explored the possibility of using Full-Wave Inversion for GPR data as a means to improve the rate anti-personnel landmines can be cleared. A significant constraint to the speed at which landmines can be cleared is the need to remove every piece of metal found by a metal detector, as one cannot be sure they are not part of a landmine without further information. A GPR can be used to rule out some metal fragments as *not landmines* (for example, when there is no GPR response), and we hold that the additional quantitative information from FWI will be able to reduce the rate of false positives further. A significant portion of this thesis has been considered with the optimisation problem of FWI, as well as understanding sensitivity of GPR measurements.

In Chapter 4, we presented all the material necessary to understand and implement FWI in 2D, for which we used a finite-difference approximation of the Helmholtz equation to model the forward problem. We described the implementation of a suitable optimisation and linesearch method to solve the inverse problem, discussed suitable stopping criteria and modifications of the optimisation method for bound constraints, considered a suitable regularisation strategy using the Total Variation measure, and frequency selection strategies. We presented a near-surface reconstruction of simulated data including significant non-linear effects, owing to the complex shaped and hollow targets, for a small wideband GPR array. The use of TV regularisation and consideration of hollow targets is novel for GPR FWI of simulated data.

As FWI is usually carried out only on data from large arrays, but we had positive reconstruction results of simulated data for small hand-held arrays, we wanted to more formally understand what data was required. In Chapter 5 we explored this question with the aid of a Singular Value Decomposition of the Jacobian matrix, which (in the linear approximation) maps GPR data to the image. Where other studes have looked at the singular values to undestand the conditioning of the inverse problem, in this work we went further and explored the resulting singular vectors which give us a vector basis describing objects we can image for a given noise level. A novel numerical experiment was used to help determine what was required of an array in order to distinguish nearby objects. In this experiment, we found that while a multi-static array was certainly necessary, the degree to which it was multi-static was less important. The results are applicable to imaging methods other than FWI, and so we believe use of a multi-static array is important for landmine detection where there is a need to resolve targets in a highly cluttered environment.

A 2D model of the subsurface, and scattering therein, involves the assumption that it is invariant in one direction. This approximation is neither suitable for landmines nor the highly cluttered environments they can be found in, and so we wish to perform FWI in 3D. This requires a forward model which can accurately simulate the more complex scattering of electromagnetic waves in 3D, and so in Chapter 6 we presented a combined finite-element boundary-integral solver for electromagnetic scattering in the frequency domain. The finite element part describes scattering in the inhomogeneous subsurface, and the formulation was given for first order tetrahedral edge (Nédélec) elements. The infinite subsurface was truncated with a perfectly matched layer (PML), and transmission of electromagnetic waves from the air into the subsurface was governed by coupling to the boundary integral part of the solver, which is a novel way of simulating GPR data. Simulations were compared qualitatively to GPR field data with positive results.

In Chapter 7, we presented an asymptotic formula from the literature describing the scattering of electromagnetic waves from small inclusions using the Pólya Szegö polarization tensor, and by application of the surface equivalence principle we gave a novel demonstration that the expression could be re-written in a more intelligible way which which can be suitable for describing GPR scattering of targets in a homogeneous ground. We used this formula to help us understand sensitivity for FWI of electromagnetic waves in 3D, deriving a novel expression similar to those commonly used to understand wavenumber coverage in the image domain for FWI either in 2D or for scalar valued waves (e.g. [159]). This expression helps to understand how cross-polar measurements can increase resolution in GPR imaging: they add a different class of test functions (like a quadrupole as opposed to a dipole) to describe the subsurface. The idea of the polarization tensor describing sensitivity was contrasted to the more common description using Fréchet derivatives, and this was used to undertake numerical experiments into the sensitivity of GPR arrays to mine-like targets.

Chapter 8 presented the first numerical experiments into 3D FWI for GPR data with a small hand-held array using a FE-BI system to solve the forwad problem. Naturally, there were significant restrictions on the size of domain and dataset we were able to simulate due to the computational and memory cost of inversions in 3D. We re-examined our aims with regards to the landmine detection problem, that we wish to resolve a target in a known location in the presence of clutter, and so developed a novel algorithm for reconstructing the subsurface in a region of interest while allowing the surrounding medium to vary. The algorithm performed well in a numerical experiment to resolve a single target in a highly inhomogeneous stochastic background medium. We then discussed whether the results ought to help in the mine clearance process – safely reducing false positive detections – and an idea of what would need to be done before the method could be used in practice.

Finally, in Chapter 9 we derived a novel approximation to elements of the Hessian matrix using the polarization tensor approximation given in Chapter 7, which was used to precondition the l-BFGS optimisation scheme. This extended the approximation of the Gauss-Newton Hessian used for FWI of scalar valued waves, based on Green's functions, to electromagnetic waves. Since the approximation accounts for contrast of parameters to the background medium, a novel trust region model was used to determine a good initial step length for the optimisation problem. While the exact expression for the Hessian used is only applicable to scattering of electromagnetic waves in a bounded domain, the same principle could be used for any application for which a similar asymptotic approximation exists.

10.2 Future work

In order to realise full-wave inversion as a field-ready imaging method for landmine detection there are three main areas of future work that would need to be undertaken: improvements and verification of the forward model; development of the inversion algorithm; and a long term testing program of field data,

Firstly, the forward model needs verification against well-controlled GPR data, to be sure of its accuracy (particularly, an error bound on how *inaccurate* it is). It could also be made more efficient in several places: the PML region has not had a thorough treatment in terms of finding the most efficient parameters (such as width, the rate at which it is increased); higher order basis functions are almost certainly more appropriate, as one needs many linear elements to accurately represent a wave-like function; and the efficiency of the combined FE-BI system also needs to be considered. A more robust form of CFIE may be required for field data, or one may find a time-domain solver will inevitably be preferred. None of these areas require novel research and development, as are already the topic of other research fields, but it is necessary to undertake such studies both to be sure of the accuracy of a fielded system and to reduce computation time as much as possible.

Secondly, the efficiency of the inversion algorithm ought to be improved in order to deal with real data fast enough not to deter de-mining personnel from using the equipment – i.e. close to real time imaging. There is of course a wait for development of portable computing equipment to be powerful enough to use such a computationally intensive method, but increasing the algorithm (and forward solver) efficiency will both bring forward the day when such methods are plausible, or allow cheaper (less powerful) devices to be used thereafter. One should not be deterred by the long lead time before powerful enough portable computers are available: landmines will be problem for a long time to come without new effective means of clearance. Specific areas of improving algorithm efficiency include appropriate choice of parameters for field data, such as linesearch (Wolfe) parameters, the l-BFGS memory needed, and the amount of regularisation required for the noise level of the GPR equipment used.

We have only considered reconstruction of a domain with a flat ground surface, though a rough or unevan ground can cause significant difficulties in imaging the subsurface. While it was beyond the scope of this work, the affect of a realistic non-flat ground surface on spatial sensitivity and inversion results must be studied. Moreover, to implement FWI in the field using the algorithms we have presented would require a preliminary algorithm to calculate the topography of the ground surface. This could be done using the first time of arrival data, caused by reflections from the ground surface, which is an easier problem to solve than imaging the subsurface (it is a linear mildly ill-posed inverse problem).

For 3D inversions, implementing TV by iteratively re-weighting l-BFGS is almost certainly inefficient, as this will require many iterations in order for the smoothing parameter ε to be reduced significantly so that $TV_\varepsilon(\mathbf{m})$ is a good approximation to $TV(\mathbf{m})$. Instead, a suitable primal-dual method will need to be implemented, such as a split Bregman algorithm [62] which is particularly suited to large-scale problems. There will also be work comparing the rate of convergence of l-BFGS, for near-field surface measurements of GPR field data, to other optimisation schemes. As mentioned in Chapter 4, hybrid quasi-Newton-CG methods may yield better convergence, as might a truncated Newton method. We also need to do more to verify the Hessian approximation presented in Chapter 9, whether by doing more to determine its effectiveness in a wider range of problems, or by directly determining how closely it approximates the components of the true Hessian matrix.

Most importantly, in order to use FWI in a hazardous application such as demining, long term field testing is required to determine statistical accuracies of reconstructions given the fit to the data. Such testing needs to be undertaken in the full range of situations landmines are found, including soil types and humidities, roughness and topography of the ground surface, and range and amount of clutter objects and inhomogeneities in the ground. As discussed in Section 8.5, we need to be sure that when an inversion result shows there is not a landmine, we need to know how different to a mine-like object the reconstruction needs to be in order to declare a target not a mine with statistical certainty. This would likely involve comparing the reconstruction to a bank of known mine types, and showing that if any one of them was present in the reconstructed background (including clutter) that the fit to the recorded GPR data would be *far* worse.

Inevitably, such certainty in reconstruction results also requires development of a multi-static hand-held array suitable for FWI. As discussed in Chapters 5 and 7, we believe a system with multi-static and cross-polar measurements is required for 3D FWI of GPR data, but the number, positioning and polarization of antennas which is optimal (for, say, a maximum equipment cost) is to be determined. A useful measure of 'optimal' would be the system which gives the greatest differences in data between landmines and common clutter objects, but we must also consider the need to resolve targets in highly inhomogeneous ground (and the two measures may conflict). The question of the optimal system is then non-trivial, would need both numerical studies and empirical testing with lab and field data, and likely involve some compromise for the cost or complexity of the system.

Appendices

Appendix A

Definitions and Formulae

A.1 Function spaces

We give here the definitions of function spaces required for this thesis. For further reading, and an introduction to functional analysis, see for example Monk [119, Chapters 2-3]

Let $\Omega \subset \mathbb{R}^n$. The Lebesgue space $L^p(\Omega)$ is the space of all measurable functions $f : \omega \to \mathbb{R}$ (or $\mathbb{C}$) whose absolute value raised to the power p is integrable over Ω,

$$L^p(\Omega) = \left\{ f : \int_\Omega |f|^p \,\mathrm{d}\Omega < \infty \right\}, \tag{A.1.1}$$

equipped with norm

$$\|f\|_{L^p(\Omega)} = \left(\int_\Omega |f|^p \,\mathrm{d}\Omega \right)^{1/p}. \tag{A.1.2}$$

The space $L^2(\Omega)$ is a Hilbert space equipped with inner product

$$\langle f, g \rangle_{L^2(\Omega)} = \int_\Omega f\overline{g} \,\mathrm{d}\Omega. \tag{A.1.3}$$

For vector valued functions $\mathbf{F} : \Omega \to \mathbb{R}^3$ (or $\mathbb{C}^3$), the space $\mathbb{L}^p(\Omega)$ is the space of all measurable vector functions

$$\mathbb{L}^p(\Omega) = \left\{ \mathbf{F} : \int_\Omega |\mathbf{F}|^p \,\mathrm{d}\Omega < \infty \right\}, \tag{A.1.4}$$

equipped with norm

$$\|\mathbf{F}\|_{\mathbb{L}^p(\Omega)} = \left(\int_\Omega |\mathbf{F}|^{1/p} \,\mathrm{d}\Omega \right)^{p}. \tag{A.1.5}$$

The space $\mathbb{L}^2(\Omega)$ is a Hilbert space equipped with inner product

$$\langle \mathbf{F}, \mathbf{G} \rangle_{\mathbb{L}^2(\Omega)} = \int_\Omega \mathbf{F} \cdot \overline{\mathbf{G}} \,\mathrm{d}\Omega. \tag{A.1.6}$$

The Hilbert space $H^1(\Omega)$ is the space of functions f which are square integrable, and whose first derivative is square integrable,

$$H^1(\Omega) = \{f : \; f \in L^2(\Omega), \nabla f \in L^2(\Omega)\}. \tag{A.1.7}$$

This space is alternately written as $\mathbb{L}^2_{\text{grad}}(\Omega)$ in some texts. It is equipped with inner product

$$\langle f, g\rangle_{H^1(\Omega)} = \langle f, g\rangle_{L^2(\Omega)} + \langle \nabla f, \nabla g\rangle_{L^2(\Omega)}, \tag{A.1.8}$$

which defines the norm

$$\|f\|^2_{H^1(\Omega)} = \langle f, f\rangle . \tag{A.1.9}$$

The Hilbert spaces $H^1(\Omega)$ are also the Sobolev space $W^{1,2}(\Omega)$. More generally, $H^k(\Omega) = W^{k,2}(\Omega)$.

For vector valued functions $\mathbf{F} : \Omega \to \mathbb{R}^3$ (or $\mathbb{C}^3$), we have the two Hilbert spaces

$$H^1_{\text{div}}(\Omega) =\{\mathbf{F} : \; \mathbf{F} \in \mathbb{L}^2(\Omega), \nabla \cdot \mathbf{F} \in \mathbb{L}^2(\Omega)\}, \tag{A.1.10}$$

$$H^1_{\text{curl}}(\Omega) =\{\mathbf{F} : \; \mathbf{F} \in \mathbb{L}^2(\Omega), \nabla \times \mathbf{F} \in \mathbb{L}^2(\Omega)\}. \tag{A.1.11}$$

These are equipped with inner products

$$\langle \mathbf{F}, \mathbf{G}\rangle_{H^1_{\text{div}}(\Omega)^2} = \langle \mathbf{F}, \mathbf{G}\rangle_{\mathbb{L}^2(\Omega)} + \langle \nabla \cdot \mathbf{F}, \nabla \cdot \mathbf{G}\rangle_{\mathbb{L}^2(\Omega)} \tag{A.1.12}$$

and

$$\langle \mathbf{F}, \mathbf{G}\rangle_{H^1_{\text{curl}}(\Omega)} = \langle \mathbf{F}, \mathbf{G}\rangle_{\mathbb{L}^2(\Omega)} + \langle \nabla \times \mathbf{F}, \nabla \times \mathbf{G}\rangle_{\mathbb{L}^2(\Omega)}, \tag{A.1.13}$$

respectively. These inner products define the norms

$$\|\mathbf{F}\|^2_{H^1_\cdot(\Omega)} = \langle \mathbf{F}, \mathbf{F}\rangle_{H^1_\cdot(\Omega)} . \tag{A.1.14}$$

These Hilbert spaces are alternatively written $H^1_{\text{div}}(\Omega) = H^1(\text{div}, \Omega) = \mathbb{L}^2_{\text{div}}(\Omega)$ and $H^1_{\text{curl}}(\Omega) = H^1(\text{curl}, \Omega) = \mathbb{L}^2_{\text{curl}}(\Omega)$ in some texts.

A.2 Landau notation

Landau notation introduces two "order of magnitude" operators used throughout this thesis, *little-o* ad *Big-$\mathcal{O}$* notation. Little-o provides a function which is of lower order magnitude. Formally, let $f, g : X \subset \mathbb{R} \to \mathbb{R}$, then if

$$\lim_{x \to c} \frac{f(x)}{g(x)} = 0, \quad c \in \mathbb{R}, \tag{A.2.1}$$

then we say that

$$f(x) = o(g(x)) \text{ as } x \to c. \tag{A.2.2}$$

Big-$\mathcal{O}$ provides a function which is at most the same magnitude. Formally, let $f, g : X \subset \mathbb{R} \to \mathbb{R}$, then if there exists an $M > 0$ such that

$$\lim_{x \to c} \left| \frac{f(x)}{g(x)} \right| = 0, \quad c \in \mathbb{R}, \tag{A.2.3}$$

we say that

$$f(x) = \mathcal{O}(g(x)) \text{ as } x \to c. \tag{A.2.4}$$

Little-o is the stronger statement,

$$f(x) = o(g(x)) \text{ as } x \to c \quad \Rightarrow \quad f(x) = \mathcal{O}(g(x)) \text{ as } x \to c. \tag{A.2.5}$$

A.3 Linearised step length

Pica et al. [132] give a linearised step length for use in FWI based on a finite difference approximation of $\dfrac{\partial \mathcal{F}}{\partial \mathbf{p}_k}$, where $\mathbf{p}_k$ is the descent direction. Consider the Misfit function (for real valued $\mathcal{F}$)

$$\begin{aligned} \mathcal{J}(\mathbf{m}_k + \alpha \mathbf{p}_k) =& \frac{1}{2} [\mathcal{F}(\mathbf{m}_k + \alpha \mathbf{p}_k) - \mathbf{d}]^T [\mathcal{F}(\mathbf{m}_k + \alpha \mathbf{p}_k) - \mathbf{d}] \\ =& \frac{1}{2} [\mathcal{F}(\mathbf{m}_k) + \alpha \nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k - \mathbf{d}]^T \\ & \times [\mathcal{F}(\mathbf{m}_k) + \alpha \nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k - \mathbf{d}] + \mathcal{O}(|\alpha \mathbf{p}_k|^2), \end{aligned} \tag{A.3.1}$$

where $\nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k = \left. \dfrac{\partial \mathcal{F}}{\partial \mathbf{p}_k} \right|_{\mathbf{m}_k}$. The optimal step length α would minimise $\mathcal{J}$, and so has

$$\frac{\partial \mathcal{J}(\mathbf{m}_k)}{\partial \alpha} = 0. \tag{A.3.2}$$

Differentiating (A.3.1) with respect to α, one obtains

$$\frac{\partial \mathcal{J}}{\partial \alpha} = [\mathcal{F}(\mathbf{m}_k) + \alpha \nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k - \mathbf{d}]^T [\nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k] + \mathcal{O}(|\alpha \mathbf{p}_k|^2). \tag{A.3.3}$$

Equating to zero and neglecting higher order terms, one finds the α which minimises $\mathcal{J}$ along $\mathbf{p}_k$ is given by

$$\alpha_k = -\frac{[\mathcal{F}(\mathbf{m}_k) - \mathbf{d}]^T [\nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k]}{[\nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k]^T [\nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k]}. \tag{A.3.4}$$

Approximating the directional derivative of $\mathcal{F}$ with first order finite differences, one has

$$\alpha_k = -\kappa \frac{[\mathcal{F}(\mathbf{m}_k) - \mathbf{d}]^T [\mathcal{F}(\mathbf{m}_k) - \mathcal{F}(\mathbf{m}_k + \kappa \mathbf{p}_k)]}{[\mathcal{F}(\mathbf{m}_k) - \mathcal{F}(\mathbf{m}_k + \kappa \mathbf{p}_k)]^T [\mathcal{F}(\mathbf{m}_k) - \mathcal{F}(\mathbf{m}_k + \kappa \mathbf{p}_k)]}, \tag{A.3.5}$$

where $\kappa \in \mathbb{R}^+$ is a suitably small number.

Since we have a complex valued $\mathcal{F}$ as well as a regularisation term R, the cost function becomes

$$\begin{aligned} \mathcal{J}(\mathbf{m}_k + \alpha \mathbf{p}_k) =& \frac{1}{2} [\overline{\mathcal{F}(\mathbf{m}_k) + \alpha \nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k - \mathbf{d}}]^T \\ & \times [\mathcal{F}(\mathbf{m}_k) + \alpha \nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k - \mathbf{d}] \\ & + R(\mathbf{m}_k) + \alpha \nabla R(\mathbf{m}_k)^T \mathbf{p}_k + \mathcal{O}(|\alpha \mathbf{p}_k|^2). \end{aligned} \tag{A.3.6}$$

Differentiating (A.3.6) with respect to α,

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \alpha} =& \Re \left\{ [\mathcal{F}(\mathbf{m}_k) + \alpha \nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k - \mathbf{d}] \right\}^T \Re \left\{ [\nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k] \right\} \\ & + \Im \left\{ [\mathcal{F}(\mathbf{m}_k) + \alpha \nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k - \mathbf{d}] \right\}^T \Im \left\{ [\nabla \mathcal{F}(\mathbf{m}_k) \mathbf{p}_k] \right\} \\ & + \nabla R(\mathbf{m}_k) \mathbf{p}_k + \mathcal{O}(|\alpha \mathbf{p}_k|^2), \end{aligned} \tag{A.3.7}$$

and so we have the approximate minimiser

$$\begin{aligned} \alpha_k =& -\kappa \Big([\Re \left\{ \mathcal{F}(\mathbf{m}_k) - \mathbf{d} \right\}]^T [\Re \left\{ \mathcal{F}(\mathbf{m}_k) - \mathcal{F}(\mathbf{m}_k + \kappa \mathbf{p}_k) \right\}] \\ & + [\Im \left\{ \mathcal{F}(\mathbf{m}_k) - \mathbf{d} \right\}]^T [\Im \left\{ \mathcal{F}(\mathbf{m}_k) - \mathcal{F}(\mathbf{m}_k + \kappa \mathbf{p}_k) \right\}] + \kappa \nabla R(\mathbf{m}_k)^T \mathbf{p}_k \Big) \\ & \div [\overline{\mathcal{F}(\mathbf{m}_k) - \mathcal{F}(\mathbf{m}_k + \kappa \mathbf{p}_k)}]^T [\mathcal{F}(\mathbf{m}_k) - \mathcal{F}(\mathbf{m}_k + \kappa \mathbf{p}_k)]. \end{aligned} \tag{A.3.8}$$

A.4 Stochastic media model

Jiang et al. [73, 78] give a stochastic media model with an ellipsoidal autocorrelation function, which is able to give a fractal spatial description of the subsurface over a range of length scales, from layer-like to granular medium types. The ellipsoidal autocorrelation function is given by

$$f(x, y, z) = \exp \left[- \left(\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} \right)^{1/(1+r)} \right], \tag{A.4.1}$$

where a, b and c are the autocorrelation lengths, and r is a roughness factor. $r = 1$ corresponds to an exponential ellipsoidal, $r = 0$ a Gaussian ellipsoidal, and $0 < r < 1$ an intermixed ellipsoidal autocorrelation function. Larger values of a, b and b create larger-scale perturbations. Setting $a, b \ll c$ will create a layer-like medium.

To generate the random medium, we first Fourier transform f,

$$F(k_x, k_y, k_z) = \iiint_{-\infty}^{\infty} f(x, y, z) \exp\left[-2\pi\mathrm{i}(k_x x + k_y y + k_z z)\right] \,\mathrm{d}x \,\mathrm{d}y \,\mathrm{d}z. \tag{A.4.2}$$

The power spectrum F is combined with a stochastic field $\theta(k_x, k_y, k_z) \in [0, 2\pi]$ created by a (pseudo) random number generator and inverse Fourier transformed, resulting in the stochastic perturbation

$$\mu(x, y, z) = \iiint_{-\infty}^{\infty} F(k_x, k_y, k_z) \exp(i\theta) \exp\left[-2\pi\mathrm{i}(k_x x + k_y y + k_z z)\right] \,\mathrm{d}k_x \,\mathrm{d}k_y \,\mathrm{d}k_z. \tag{A.4.3}$$

A stochastic perturbation to the background permittivity is then given by

$$\phi = \frac{v}{\sigma}[\mu - \bar{\mu}], \tag{A.4.4}$$

where $\bar{\mu}$ is the mean value of μ, σ is the variance of μ and v the required variance of the stochastic model.

Appendix B

Selected Code

B.1 Nuisance parameter linesearch

The file `nuisance_search.m` [194] computes step lengths α_t via a strong Wolfe line search, and $\alpha_n(\alpha_t)$ via a linearised step length, as in Algorithm 6.

```
function [ alpha, phi_alpha, grad_f ] = nuisance_search( x, p, n_indx, data, alpha_1, alpha_max, f,
     g, mdl, params, CFIEp, phi_0, grad_f )
%NUISANCE_SEARCH
%[ alpha, phi_alpha, grad_f ] = nuisance_search( x, p, n_indx, data, alpha_1, alpha_max, f, g, mdl,
     params, CFIEp, phi_0, grad_f )
%   Perform stong Wolfe search in direction p, where parameters p(n_indx)
%   are nuisance parameters and defined implicitly by minimising in
%   direction p(~n_indx).  p(n_indx) is calculated by a linear
%   approximation, which results in either 1 (if this is close to 0 or
%   alpha for p(~n_indx)) or 2 additional forward solves per cost
%   evaluation.
%
% INPUTS:
%   x           -       current paramter model
%   p           -       update direction. Either p or p(~n_indx) must be a
%                       descent direction at x
%   n_indx      -       index of nuisance paramters
%   data        -       data to be fitted
%   alpha_1     -       initial step length
%   alpha_max   -       maximum step length
%   f           -       string or function handle to objective
%   g           -       string of function handle to gradient
%   mdl         -       FEM triangulation model in EIDORS format
%   params      -       parameter struct with forward, inverse and line
%                       search settings
%   CFIEp       -       boundary integral system matrices
%   phi_0       -       objective value at x
%   grad_f      -       gradient at x
%
% OUTPUTS:
%   alpha       -       step lengths alpha=[alpha_t, alpha_n]
%   phi_alpha   -       objective value f(x + alpha p)
%   grad_f      -       gradient g(x + alpha p)
%
%
% (c) 2015 FM Watson

% Check params/set default values
if isfield(params.line,'c1')==0
    params.line.c1 = 1e-3;
end

if isfield(params.line,'c2')==0
    params.line.c2 = 0.9;
end

if isfield(params.line,'max_it')==0
    params.line.max_it = 10;
end


if isfield(params.line, 'alpha_step')
    alpha_step = params.line.alpha_step;
    if ischar(alpha_step)
        switch params.line.alpha_step

            % Calc step so last is alpha_max
            case 'root'
                alpha_step = nthroot(alpha_max/alpha_1,params.line.max_it);
            otherwise
                error('params.line.alpha_step can currently take a numeric (>1) or the string "root"');
        end
    end

else
    alpha_step = 1.5;
end

% min step
if isfield(params.line, 'delta_min')==0
    params.line.delta_min = 1e-3;
    delta_min = params.line.delta_min;
else
    delta_min = 1e-3;
end

% min distance to interpolate
if isfield(params.line, 'nuis_delta')==0
    params.line.nuis_delta = 1e2 * delta_min;
    nuis_delta = 1e2 * delta_min;
else
    nuis_delta = params.line.nuis_delta;
end

% max nuisance step
if isfield(params.line, 'nuis_max_step')==0
    params.line.nuis_max_step = 1.5;
    nuis_c = params.line.nuis_max_step;
else
    nuis_c = params.line.nuis_max_step;
end

% Check F
if ischar(f)
    F = str2func(f);
elseif isa(f,'function_handle')
    F = f;
else
    error('expecting a function for input F')
end

% Check G
if ischar(g)
    G = str2func(g);
elseif isa(f,'function_handle')
    G = g;
else
    error('expecting a function for input g')
end

% nuisance and model parameter update directions
p_mdl = zeros(size(p));
p_nuis = zeros(size(p));

t_indx = setdiff(1:length(x), n_indx);

p_nuis(n_indx) = p(n_indx);
p_mdl(t_indx) = p(t_indx);


% Initialise
alpha_old = 0; alpha_new = alpha_1; k=1;
```

```
% initial value/directional derivative
% [phi_0, ~, grad_f] = F(x, data, mdl, params);

% Descent direction gradient at alpha=0 -- NB will remove contn from
% nuisance params
d_phi_0 = dot(grad_f, p_mdl);
d_phi_0_full = dot(grad_f,p);

phi_old = phi_0;
d_phi_old = d_phi_0;


% Initialise ? seems to need it ?
alpha = alpha_1;
phi_alpha = phi_old;

% initialise interpolation points
ALPHX = zeros(2*params.line.max_it, 2);
Gradpts = zeros(params.line.max_it,2);
Gradpts(1,2) = d_phi_0;

if(d_phi_0 >=0) && (d_phi_0_full >=0) % Check descent direction
    fprintf('grad f . p_mdl = %6.4g, grad_f . p = %6.4g\n ',   d_phi_0, d_phi_0_full);
    disp('Either p or p_mdl must be a descent direction for f at x');
    alpha = 0;
    phi_alpha = phi_0;
    return;
end % if descent

% Indx to ALPHX
k_tot = 1;

% Save first grad
grad_f_0 = grad_f;

% Run optimisation
while k<=params.line.max_it
    str = sprintf('    nuisance_search iteration %i, alpha=%6.4f',k, alpha_new);
    disp(str);
    %

    [phi_new, alpha_nuis, nuis_step, delta_d, Ecell] = nuisance_inner(x + alpha_new * p_mdl);

    if flag>0
        alpha=alpha_old;
        phi_alpha = phi_old;
        break
    end

    % New value does not satisfy sufficient decrease condition
    if phi_new> phi_0 + params.line.c1*alpha_new* d_phi_0 || (phi_new >= phi_old && k>1)

        [alpha, alpha_nuis, ~, phi_alpha, grad_f] = nuisance_zoom(alpha_old, alpha_new,  phi_old,
     d_phi_old, phi_new);

        break
    end % if

    % Calc gradient
    disp('      calculating gradient...')
    [grad_f] = G( x + alpha_new * p_mdl + nuis_step, delta_d, Ecell, mdl, params, CFIEp );
    if flag>0

        alpha=alpha_old;
        phi_alpha = phi_old;

        break
    end

    % Directional derivative
    d_phi_new =  dot(grad_f,p_mdl);

    % Satisfies Wolfe conditions - break
    if abs(d_phi_new) <= -params.line.c2 * d_phi_0

        alpha = alpha_new;
        phi_alpha = phi_new;

        break;
    end % if


    % Increasing function
    if d_phi_new >=0

        [alpha, alpha_nuis, ~,  phi_alpha, grad_f] = nuisance_zoom(alpha_new, alpha_old,phi_new,
 d_phi_new, phi_old);

        break
    end % if

    % If alpha was sufficiently close to alpha_max, approx at max step
    % length so alpha_max is constrained minimum
    if abs(alpha_max - alpha_new) < params.line.delta_min
        alpha = alpha_new;
        phi_alpha = phi_new;

        break

    end

    % update iteration number
    k = k+1;

    % Indx to ALPHX
    k_tot = k_tot + 1;

    % Save gradient interp pts
    Gradpts(k,1) = alpha_new;
    Gradpts(k,2) = phi_alpha;

    % Save old alpha
    alpha_old = alpha_new;

    % Try interp pt
    if k>=3
        % Quadratic interp
        alph_interp_grads = quad_interp_vvv(Gradpts(1+k-3,1), Gradpts(2+k-3,1), Gradpts(3+k-3,1),
 ...
                                            Gradpts(1+k-3,2), Gradpts(2+k-3,2), Gradpts(3+k-3,2));

    else
        % Linear interp
        alph_interp_grads = Gradpts(2,2)*Gradpts(2,1)/(Gradpts(1,2) - Gradpts(2,2));

    end

    % Take the larger step of interpolated and
    alpha_new = max(alpha_step*alpha_new, alph_interp_grads);

    % Check for max step
    alpha_new = min(alpha_new, alpha_max);

    % update old phi
    phi_old = phi_new;

end % while

alpha =  [alpha, alpha_nuis];

% not found
if k==params.line.max_it
    alpha = alpha_new;
    disp('  line_max_it reached without satisfying Wolfe conditions');
end % if


% Check didn't increase
if phi_alpha > phi_0
    phi_alpha = phi_0;
    alpha = [0,0];
    grad_f = grad_f_0;
    disp('  could not reduce function, returning 0 step')
end


% ---------------------------------------------------------------------------
    % Implicitly defined nuisance parameters
    % approximate minimiser along p_nuis in [0, alpha_max]
    function [phi_inner, alpha_inner, nuis_step, delta_d, Ecell] = nuisance_inner(x_trial)
        disp('      solving for implicitly defined variables...')


        % Check if nearby interpolant
        if nuis_c ==0
            [phi_inner, delta_d, ~, Ecell ] = F(x_trial, data, mdl, params, CFIEp);
            alpha_inner = 0;

        elseif any(abs(ALPHX(:,1) - alpha_new) < nuis_delta) && k_tot >=3

            % Unique ordered vals
            ALPHXINT = unique(ALPHX,'rows');

            % Interpolate
            alpha_inner = interp1(ALPHXINT(:,1), ALPHXINT(:,2), alpha_new, 'pchip', 'extrap' );

            % Check in range
            alpha_inner = min([alpha_inner, alpha_max, nuis_c*alpha_new]);
            nuis_step = alpha_inner * p_nuis;

            fprintf('      interpolated trial step alpha_i=%f\n',alpha_inner)

            % Trial
            [phi_inner, delta_d,~,Ecell] = F(x_trial + alpha_inner*p_nuis, data, mdl, params, CFIEp);


            % If increases on phi_old, try 0 step
            if phi_inner > phi_old && alpha_inner >= delta_min
                %
                disp('      increased at interpolation step, trying implicit step alpha_i=0')
                [phi_inner0, delta_d0, ~, Ecell0 ] = F(x_trial, data, mdl, params, CFIEp);

                % Check if better
                if phi_inner0 < phi_inner
                    alpha_inner = 0;
                    phi_inner = phi_inner0;
                    Ecell = Ecell0;
                    delta_d = delta_d0;
                    nuis_step = zeros(size(p_nuis));

                    % Save better value
                    ALPHX(k_tot,1) = alpha_new;
                    ALPHX(k_tot,2) = 0;
                end

            end


        % Standard method
        else

            if nuis_c < 1
                alpha_inner1 = nuis_c * alpha_new;
            else
                alpha_inner1 = alpha_new;
            end

            disp('      trial implicit step alpha_i=0')
            [phi_inner0, delta_d0, ~, Ecell0 ] = F(x_trial, data, mdl, params, CFIEp);

            %
            %         phi_inner_old = phi_inner0;

            % Start with target step
            %         alpha_inner = 2*alpha_new;

            % Test unit step length
            fprintf('      trial implicit step alpha_i=%f\n',alpha_inner1)
            [phi_inner1, delta_d1,~,Ecell1] = F(x_trial + alpha_inner1*p_nuis, data, mdl, params, CFIEp);

            % Regul contribution
            [~, dR0] = Reg(x_trial, params, mdl);
            dR0 = params.reg.lambda * dR0;

            [~, dR1] = Reg(x_trial + alpha_inner1*p_nuis, params, mdl);
            dR1 = params.reg.lambda * dR1;

            % Centred approxin (combine Taylor expansions at 0, alpha))
            dR = (dR0 + dR1)/2;

            % Linearised step length
            alpha_inner = -alpha_inner1*( real(delta_d1(:) - delta_d0(:)).'*real(delta_d0(:))...
                + imag(delta_d1(:) - delta_d0(:)).'*imag(delta_d0(:)) ...
                + alpha_inner1*dot(dR, p_nuis))...
                /(norm(delta_d1(:) - delta_d0(:))^2);



            % Check not too far
            alpha_inner = min([alpha_inner,alpha_max,nuis_c*alpha_inner1]);


            % Safeguard
            % Too short a step, or function increased but interpolation increases
            if alpha_inner <= 0.2 * alpha_inner1 || (alpha_inner >= alpha_new && phi_inner1 >= phi_inner0 )
                alpha_inner = 0;
                phi_inner = phi_inner0;
                Ecell = Ecell0;
                delta_d = delta_d0;
                nuis_step = zeros(size(p_nuis));
```

```

                % Close to alpha_inner1
            elseif alpha_inner >= 0.95*alpha_inner1 && alpha_inner <= 1.1*alpha_inner1
                alpha_inner = alpha_inner1;
                phi_inner = phi_inner1;
                delta_d = delta_d1;
                Ecell = Ecell1;
                nuis_step = alpha_inner1*p_nuis;

            else
                % Test linear step length
                fprintf('      trial implicit step alpha_i=%f\n',alpha_inner)
                [phi_inner, delta_d,~,Ecell] = F(x_trial + alpha_inner*p_nuis, data, mdl, params, CFIEp);
                nuis_step = alpha_inner * p_nuis;

            end

            % Check this didn't increase
            if phi_inner0 < min(phi_inner1,phi_inner)
                alpha_inner = 0;
                phi_inner = phi_inner0;
                Ecell = Ecell0;
                delta_d = delta_d0;
                nuis_step = zeros(size(p_nuis));

            elseif phi_inner1 < min(phi_inner0,phi_inner)
                alpha_inner = alpha_inner1;
                phi_inner = phi_inner1;
                delta_d = delta_d1;
                Ecell = Ecell1;
                nuis_step = alpha_inner1*p_nuis;
            end

            % Save
            ALPHX(k_tot,1) = alpha_new;
            ALPHX(k_tot,2) = alpha_inner;

        end


% ---------------------------------------------------------------------------
    % zoom search in p_mdl
    function [alpha, alpha_nuis, nuis_step,  phi_alpha, grad_f] = nuisance_zoom(alpha_lo, alpha_hi, phi_lo, d_phi_lo, phi_hi)
        strii = sprintf('    zoom called with alpha_lo=%6.4f, alpha_hi=%6.4f', alpha_lo, alpha_hi');
        disp(strii);

        zoom_max_it = 5;

        for kk=1:zoom_max_it
            % Determine range is wide enough for sufficient move /force small step
            % alpha_lo < delta_min corrected outside loop
            if abs(alpha_hi - alpha_lo) < 1.1*delta_min
                alpha_new = alpha_lo;
                phi_alpha = phi_lo;
                % grad_f will either be for alpha_lo, or will be calculated during
                % catch at end of function
                break

            end


            % Interpolate to find min using appropriate cubic or quadratic spline
            if kk==1
            % Have additional gradient info
            if alpha_lo > 0
                alpha_int = cubic_interp_vvgg(alpha_lo, alpha_hi, 0, phi_lo, phi_hi, d_phi_lo,
d_phi_0);

                % Ensure doesn't tend to infinity, interpolation wasn't singular
                % use quadratic if so (further safeguarding follows)
                if alpha_int > max(alpha_lo, alpha_hi) + delta_min || isnan(alpha_int)
                    alpha_int = quad_interp_vvg(alpha_lo, alpha_hi, phi_lo, phi_hi, d_phi_lo);
                end

                % No additional gradient info as phi_lo=0
            else
                % Quadratic interpolation minimum
                alpha_int = quad_interp_vvg(alpha_lo, alpha_hi, phi_lo, phi_hi, d_phi_lo);
            end
        else
            % Have phi_old info
            alpha_int = cubic_interp_vvvg(alpha_lo, alpha_hi, alpha_old, phi_lo, phi_hi,
phi_old, d_phi_lo);

            % Ensure doesn't tend to infinity, interpolation wasn't singular
            % use quadratic if so (further safeguarding follows)
            if alpha_int > max(alpha_lo, alpha_hi) + delta_min || isnan(alpha_int)
                alpha_int = quad_interp_vvg(alpha_lo, alpha_hi, phi_lo, phi_hi, d_phi_lo);
            end

        end

        % ----------------------------------------------------------
        % Safeguard to ensure in range, sufficient move, closer to alpha_lo,
        % isnt NaN
        if alpha_int < min(alpha_lo, alpha_hi) + delta_min || ...
                alpha_int > max(alpha_lo, alpha_hi) - delta_min ||...
                abs(alpha_int - alpha_lo) > abs(alpha_int - alpha_hi)||...
                isnan(alpha_int)

            alpha_new = ( alpha_lo + alpha_hi )/2;

            % Otherwise acceptable
        else
            alpha_new = alpha_int;

        end

        strii = sprintf('    zoom iteration %i, alpha_lo=%6.4f, alpha_hi=%6.4f, alpha=%6.4f',kk
,alpha_lo, alpha_hi, alpha_new);
        disp(strii);
        disp('      calculating cost...');

        % --------------------------------------------------------
        % cost for step length alpha
        [phi_alpha, alpha_nuis, nuis_step, delta_d_in, Ecell_in ] = nuisance_inner(x +
alpha_new * p_mdl);


        % ------------------------------------------------------------
        % Increased, choose step length in (alpha_low, alpha)
        if phi_alpha > phi_0  + params.line.c1 * alpha_new * d_phi_0 || ...
                phi_alpha >= phi_lo
            % Save old
            alpha_old = alpha_hi;
            phi_old = phi_hi;

            % calculate cost
            alpha_hi = alpha_new;

            % ------------------------------------------------------
            % Not increasing, check gradient condition
            else

                %  gradient
                disp('      calculating gradient...');
                [grad_f] = G( x + alpha_new * p_mdl + nuis_step, delta_d_in, Ecell_in, mdl, params, CFIEp );

                d_phi_alpha = dot(grad_f , p_mdl);

                strii = sprintf('      d_phi(alpha)=%6.4f', d_phi_alpha);
                disp(strii);

                % Satisfies Wolfe conditions, return value
                if abs(d_phi_alpha) <= -params.line.c2 * d_phi_0
                    break;

                elseif kk == zoom_max_it
                        disp('    line_max_it reached without satisfying curvature condition');

                end


                % Choose new range (alpha_lo, alpha_hi)
                % Determine which side of alpha min lies
                if d_phi_alpha * (alpha_hi - alpha_lo ) >=0
                    % Save old -- alpha_lo being dropped
                    alpha_old = alpha_hi;
                    phi_old = phi_hi;

                    % update alpha_hi
                    alpha_hi = alpha_lo;


                else
                    % Save old -- alpha_lo being dropped
                    alpha_old = alpha_lo;
                    phi_old = phi_lo;

                end

                %
                phi_lo = phi_alpha;
                d_phi_lo = d_phi_alpha;
                alpha_lo = alpha_new;
            end % if

            % Add to total its (indx to ALPHX)
            k_tot = k_tot + 1;

        end % while
        % For return
        alpha = alpha_new;

        % May not be defined on max its
        if exist('grad_f','var')==0
            [grad_f] = G( x + alpha_new * p_mdl + nuis_step, delta_d_in, Ecell_in, mdl, params, CFIEp );
        end

    end % nuisance_zoom function

end
```

```
% -----------------------------------------------------------------------



% -----------------------------------------------------------------------
% Interpolation minima
% Quadratic, f(x1), f(x2), f'(x1)
function x_min = quad_interp_vvg(x1, x2, v1, v2, g1)
vals = [x1^2, x1, 1; x2^2, x2, 1; 2*x1, 1, 0];
    coefs = vals\[v1; v2; g1];

    % Trial value is minimiser of quadratic
    x_min = -coefs(2)/(2*coefs(1));
end


% Quadratic, f(x1), f(x2), f(x3)
    function x_min = quad_interp_vvv(x1, x2, x3, v1, v2, v3)
        vals = [x1^2, x1, 1; x2^2, x2, 1; x3^2, x3, 1];
        coefs = vals\[v1; v2; v3];

        % Trial value is minimiser of quadratic
        x_min = -coefs(2)/(2*coefs(1));
    end


% Cubic, f(x1), f(x2), f'(x1), f'(x3)
function x_min = cubic_interp_vvgg(x1, x2, x3, v1, v2, g1, g3)
vals = [x1^3, x1^2, x1, 1;...
        x2^3, x2^2, x2, 1;...
        3*x1^2, 2*x1, 1, 0;...
        3*x3^2, 2*x3, 1, 0];
    coefs = vals\[v1; v2; g1; g3];

    % Trial value is minimiser of quadratic
    x_min = (-coefs(2) + sqrt(coefs(2)^2 - 3*coefs(1)*coefs(3)))/(3*coefs(1));
end

% Cubic, f(x1), f(x2), f(x3), f'(x1)
function x_min = cubic_interp_vvvg(x1, x2, x3, v1, v2, v3, g1)
vals = [x1^3, x1^2, x1, 1;...
        x2^3, x2^2, x2, 1;...
        x3^3, x3^2, x3, 1;...
        3*x1^2, 2*x1, 1, 0];
    coefs = vals\[v1; v2; v3; g1];

    % Trial value is minimiser of quadratic
    x_min = (-coefs(2) + sqrt(coefs(2)^2 - 3*coefs(1)*coefs(3)))/(3*coefs(1));
end
```

B.2 Hessian preconditioned trust region

The file Hessian_precon_trust.m [194] calculates an initial update via a trust region model using the first order Taylor approximation of the cost function, including an approximation of the exact remainder (in integral form) using the polarization tensor approximation to the diagonals of the Hessian matrix. See Algorithm 7 for pseudo code. The Hessian components are calculated via Hess_deltas.m and initial_Hessian.m (given in Watson [194]).

```
function [ m0, C, G, H, du2, du_dd ] = Hessian_precon_trust( m0, data, mdl, params, CFIEp, c0, delta_d, g0, Eh0, du2, du_dd)
%HESSIAN_PRECON_NONLIN
% [ m0, C, G, H, du2, du_dd ] = Hessian_precon_trust( m0, data, mdl, params, CFIEp, c0, delta_d, g0, Eh0, du2, du_dd)
%
% Gauss-Newton Hessian type preconditioner for FWI based on polarization
% tensor approximation. Trust region algorithm searches along
% p=-H(0)^{-1}g0, and for a model uses the first order Taylor expansion
% with the exact remainder in integral form (calculated numerically). The
% remainder term has the true Hessian replaced by the tensor-approximated
% diagonals.
%
% INPUTS:
%   m0          -   initial model
%   data        -   GPR data to fit
%   mdl         -   FEM model in EIDORS format
%   params      -   parameter struct
%   CFIEp       -   boundary integral system matrices
% The following are not mandatory:
%   c0          -   initial cost
%   delta_d     -   initial data residuals
%   g0          -   initial gradient
%   Eh0         -   cell of forward fields for m0
%   du2         -   approximate Gauss-Newton diagonal terms
%   du_dd       -   approximate Hessian second derivative diagonals
%
% OUTPUTS:
%   m0          -   updated initial model
%   C           -   updated initial cost
%   G           -   updated initial gradient
%   H           -   initial Hessian approximation
%   du2         -   approximate Gauss-Newton diagonal terms
%   du_dd       -   approximate Hessian second derivative diagonals
%
%
% Currently only accounts for permittivity - conductivity assumed 0,
% permeability assumed 1
% Currently only leading diagonal preconditioner
%
% (c) 2015 FM Watson

% ------------------------------------------------------------------------
% Setup

disp('Hessian precon called')

if isfield(params,'precon')==0
    params.precon = [];
end
```

```
if nargin<9 && nargin >=5
    % Cost, gradient
    disp('  calculating initial cost and gradient...')
    [c0, delta_d,~,Eh0,g0] = objective(m_0, data, mdl,  params, CFIEp);

    % Polarization tensor approximate Hessian terms
    [du2, du_dd] = Hess_deltas(m0, delta_d, Eh0, mdl, params);

%  ----------------------------------------------------------------------------
% Delta terms if not provided
elseif nargin==9
    [du2, du_dd] = Hess_deltas(m0, delta_d, Eh0, mdl, params);

elseif nargin~=11
    error('invalid inputs for Hessian_precon_trust')
end


% -----------------------------------------------------------------------------
% Some setup
% Don't calc for nuisance params?
if isfield(params.precon,'skip_nuisance')
    skip_nuisance = params.precon.skip_nuisance;
else
    skip_nuisance = true;
end

%Assume homog eps0 - should be!
eps0 = m0(1)*params.EM.eps0;

if isfield(params.precon,'lambda')
    lambda = params.precon.lambda;
else
    lambda = 0;
end

% Trust region minimum
if isfield(params.line, 'delta_min')
    delta_min = params.line.delta_min;
else
    delta_min = 1e-6;
end

% -----------------------------------------------------------------------------
% Preconditioning direction
disp('  calculating descent direction and trust-region costs...')
% Hessian at 0
[H01, H02] = initial_Hessian(du2, du_dd, eps0, 0, params);

% Regularisation
H0 = H01 + lambda;

% 2nd deriv if posdef
if all(H0 + H02 > 1e-6)
    H0 = H0 + H02;
end


% Descent direction
p0 = - g0./H0;

% If nuisance, set to 0 for trust region algorithm
if isfield(params.mdl,'nuisance') && skip_nuisance

    p0(params.mdl.nuisance) = 0;
```

```
end


% ---------------------------------------------------------------------------
% Trust region costs

% max step
alpha_max = max_move(m0, p0, mdl, params);
if isfield(params.line,'alpha_max')
    alpha_max = min(alpha_max,params.line.alpha_max);
end

% Disc pts
alpha_ii = [0, logspace(log10(delta_min),log10(alpha_max),101)];
c_ii = zeros(1,length(alpha_ii));
c_ii(1) = c0;

p_H_p = zeros(1,length(alpha_ii));
p_H_p(1) = dot(p0,p0.*H0);

for ii=1:length(alpha_ii)-1
    % Hessian with saturation alpha_ii * p
    [H01, H02] = initial_Hessian(du2, du_dd, eps0, ...
                                    alpha_ii(ii+1)*p0, params);
    H = H01 + lambda;
    % 2nd deriv if posdef
    if all(H + H02 > 1e-6)
        H = H + H02;
    end

    % gradient ctn
    c_ii(ii+1) = c0 + alpha_ii(ii+1) * dot(g0, p0);

    % Next approx Hessian ctn
    p_H_p(ii+1) = dot(p0,p0.*H);

    % Add approximate remainder term
    c_ii(ii+1) = c_ii(ii+1) + ...
        0.5* trapz(linspace(0,1,ii+1),...
                   (1 - (ii:-1:0)/ii) .*alpha_ii(ii+1).^2.*p_H_p(1:ii+1));

end



% ---------------------------------------------------------------------------
% Remove negative as untheasable
if any(c_ii<0)
    ind_neg = find(c_ii<=0,1, 'first');
    alpha_ii = alpha_ii(1:ind_neg-1);
    c_ii = c_ii(1:ind_neg-1);
    alpha_max = alpha_ii(end);
end

% ---------------------------------------------------------------------------


% Approx min
[alpha_min0, c_est0] = trustmin(alpha_ii, c_ii, delta_min, alpha_max);


% Actual cost
fprintf('  evaluating objective at trust region minimum, alpha = %f\n', alpha_min0)

[c_true,~,~,Ecell] = objective(m0 + alpha_min0*p0, data, ...
                                mdl, params, CFIEp);
```

```



% Check agreement
rho = (c0 - c_true)/(c0 - c_est0);

% Re-evaluate if (very) poor agreement or increased cost
kk=1; max_its = 5;
rho_tol = 0.02;

% save for true vals
c_true_old = [];
alpha_old = [];
while rho < rho_tol
    disp('  Reducing trust region...')

    % save old
    c_true_old = [c_true_old, c_true];
    alpha_old = [alpha_old, alpha_min0];

    % reduce region
    alpha_max = 0.25*alpha_max;

    if alpha_max < delta_min
        disp('  trust region failed at alpha_max < alpha_min')
        break
    end

    % Replace nearest nbrs, drop over alpha_max
    % remove more the smaller/more negative rho is
    [~,ind_rm] = find(abs(alpha_ii - alpha_min0) <...
                                0.1*exp(rho_tol)*alpha_min0/exp(rho) ...
                                & alpha_ii < alpha_old(end)...
                                & alpha_ii  > alpha_ii(2));

    ind_rm = unique([ind_rm, find(alpha_ii >= alpha_min0) ]);

    c_ii(ind_rm) = [];
    alpha_ii(ind_rm) = [];

    % new min
    [alpha_interp, interp_indx] = unique([alpha_ii, alpha_old], 'last');
    interp_vals = [c_ii, c_true_old];
    interp_vals = interp_vals(interp_indx);
    [alpha_min0, c_est0] = trustmin(alpha_interp, interp_vals,...
                                        delta_min, alpha_max);

    % Actual cost (check good val for step, else rho stays same and reduce)
    if alpha_min0 < 0.9*min(alpha_old)
        fprintf('  evaluating objective at trust region minimum, alpha = %f\n', alpha_min0)
        [c_true,~,~,Ecell] = objective(m0 + alpha_min0*p0, data,...
                                        mdl, params, CFIEp);
    end

    % If increased on a backstep, break
    if any(c_true > c_true_old) && any(c_true_old < c0)
        [c_true, indx] = min(c_true_old);
        alpha_min0 = alpha_old(indx);
        break
    end

    % Agreement
    rho = (c0 - c_true)/(c0 - c_est0);

    if kk>=max_its
```

```
        disp('  Trust region method failed to converge in max_its')
        break
    end

    kk = kk+1;

end



% Check reduced, if not return approximate Hess and gradient at 0
if c_true>c0
    G = g0;
    C = c0;
    H = H0;
else
    m0 = m0 + alpha_min0 * p0;
    C = c_true;
    G = gradient(m0, delta_d, Ecell, mdl, params, CFIEp);
    [H01, H02] = initial_Hessian(du2, du_dd, eps0, alpha_min0*p0, params);
    H = H01 + lambda;

    % If posdef include 2nd derivs
    if all(H + H02 > 1e-6)
        H = H + H02;
    end

end
end


function [alpha_min, c_est] = trustmin(alpha_ii, c_ii, minstep, maxstep)

if length(alpha_ii)==1
    alpha_min = 0;
    c_est = c_ii(1);

else

    % Linear constraints
    A = [1;-1]; B = [maxstep; minstep];

    % 1st step
    [~,i0] = min(c_ii);

    % Use fmincon and interpolation, starting at lowest val
    [alpha_min, c_est] = fmincon(@(x)interp1(alpha_ii, c_ii, x, ...
        'spline'), alpha_ii(i0), A, B);

end
end
```

B.3 Modified l-BFGS

The code `lbfgs.mat` [194] solves the full-wave inversion problem via l-BFGS, and includes modifications for parameter refinement, re-weighting frequency components, changing regularization parameters, dampening of BFGS update and nuisance parameters. See Algorithms 1 and 2 for pseudo-code for the l-BFGS optimisation scheme.

```
function [ m_k, resvec, g, stepl, Flag, params, M, S, Y, e_vec ] = lbfgs( d, f, m_0, mem, mdl, params, CFIEp, C0, G0, m_true )
%LBFGS Performs L-BFGS optimisation for cost function f
% [ m_k, resvec, g, stepl, Flag, params, M, S, Y, e_vec ] = lbfgs2(d, f, m_k, mem, params, C0, G0, m_true)
%   Inputs:
%       d       -   data
%       f       -   function handle or string
%       m_k     -   initial estimate
%       mem     -   number or previous iterations to store
%       params  -   struct containing domain and collection manifold
%                   information, max it/tolerance and regularisation
%       C0, G0  -   initial cost and gradient (both must be supplied or
%                   will be ignored).  Allows user to specify a modified
%                   initial descent direction G0, but this may cause a
%                   Wolfe linesearch to fail
%       m_true  -   the true value to test convergence against.  Must only
%                   be used if e_vec is an output
%
%   Initial Hessian may be supplied in params.outer.initial_Hess. May be
%   the leading diagonal, a sparse matrix, or a struct containing .H0, .S
%   and .Y defining an lBFGS Hessian matrix. Note these are not currently
%   refined with other parameters, so if the grid is refined initial_Hess
%   will be ignored
%
%   Reweighting/reparameterising occurs when an exit condition is broken.
%   For this purpose, all exit conditions are rescaled by
%   params.outer.refine_tol and .reweight_tol respectivel, with defaults 1
%
%   Outputs:
%       m_k     -   Solution returned
%       resvec  -   residual value
%       Flag    -   Returns 0 if optimisation converged, 1 if optimisation
%                   stagnated, 2 if max_its was reached before convergence,
%                   3 if invalid descent direction returned (contains any
%                   INFs of NaNs)
%       k       -   Iteration number returned,
%       M       -   Solution history
%       e_vec   -   2-norm of error against true solution x_true (if given)
%
%   Exit conditions:
%       Residual/r1 < params.outer.r_tol (relative)         -   flag = 0
%       Gradient norm < params.outer.g_tol (absolute)       -   flag = 0
%       Decrease in residual < p.o.decrease_tol (absolute)  -   flag = 1
%       Change in solution < p.o.stag  (relative)           -   flag = 1
%       Max its reached                                     -   flag = 2
%
% (c) 2015 FM Watson


% ---------------------------------------------------------------------------
% Check inputs, fields

m_k = m_0;

```

```
% Check F
if ischar(f)
    F = str2func(f);
elseif isa(f,'function_handle')
    F = f;
else
    error('expecting a function for input F')
end

disp('Beginning LBFGS optimisation')

if isfield(params.outer, 'damping')
    damping = params.outer.damping;
    if strcmp(damping,'auto')==0 && strcmp(damping, 'on')==0 && strcmp(damping, 'off')
        error('params.outer.damping can be strings "auto", "on", or "off"')
    end
else
    damping = 'auto';
end

% Nuisance params
if isfield(params.mdl, 'nuisance') && strcmp(params.line.solver, 'nuisance')
    nuis_indx = params.mdl.nuisance;
    targ_indx = setdiff(1:length(m_k), nuis_indx);
else
    nuis_indx = [];
    targ_indx = 1:length(m_k);
end

% Stopping delay
if isfield(params.outer, 'stop_delay')
    stop_delay = params.outer.stop_delay;
else
    stop_delay = 0;
end
delayed = 0;


% Weight changes
if isfield(params.outer, 'omega_weight')
    omega_weight = params.outer.omega_weight;
    params.reg.omega_weight = omega_weight(1,:);


    weight_ii = 1;
    if isfield(params.outer,'reweight_tol')
        reweight_tol = params.outer.reweight_tol;
    else
        reweight_tol = 1e0;
    end

    if isfield(params.outer,'reweight_its')
        reweight_its = params.outer.reweight_its;
    else
        reweight_its = params.outer.max_its+2;
    end


else
    reweight_tol = 1;
    weight_ii = 1;
    omega_weight = 1;
    reweight_its = params.outer.max_its + 2;


end
```

```
% Refinement
if isfield(params.outer, 'refine_pts')
    refinement = true;
    refine_pts = params.outer.refine_pts;
    refine_it = 1;
    max_refine = length(refine_pts);

    % When to refine
    if isfield(params.outer, 'refine_tol')
        refine_tol = params.outer.refine_tol;
    else
        refine_tol = 1e0;
    end

    if isfield(params.outer,'refine_its')
            refine_its = params.outer.refine_its;
    else
        refine_its = params.outer.max_its+2;
    end


else
    refinement = false;
    refine_tol = 1;
    refine_its = params.outer.max_its + 2;

end

% ----------------------------------------------------------------------
% Initialise variables/parameters

% Last m differences between x_i and x_i-1
S = zeros(numel(m_k), mem);

% Last m differences between grad x_i+1 and grad x_i
Y = zeros(numel(m_k), mem);

% Indexing for S/Y
m_ind = repmat(1:mem, 1, ceil((params.outer.max_its + 2)/2));

% Initialise 2-loop parameter
alpha_i = zeros(1,mem);

% Initialise res vector
resvec = zeros(params.outer.max_its + 1,1);

% Initialise norm of gradient
g = resvec;

d_phi_0 = 0;

% Initial cost and gradient
if nargin>=9
    if isempty(C0) || isempty(G0)
        disp('  calculating initial cost and gradient...');
        [resvec(1), ~,~,~, G_new] = F(m_k, d,mdl,  params, CFIEp);
        G_old = G_new;
        g(1) = norm(G_new);
    else
        resvec(1) = C0;
        g(1) = norm(G0);
        G_old = G0;
        G_new = G0;
    end
else
```

```
    disp('  calculating initial cost and gradient...');
    [resvec(1), ~,~,~, G_new] = F(m_k, d,mdl,  params, CFIEp);
    G_old = G_new;
    g(1) = norm(G_new);
end

fprintf('  initial cost %6.4f and norm of gradient %6.4f\n',resvec(1), norm(G_new))

% Flag
Flag = 0;

% Iteration number
k = 1;
dont_exit = false;

% Actual error output
if nargout>=6;
    M = zeros(length(m_k), params.outer.max_its +1);
    M(:,1) = m_k;

    if nargout == 10
        e_vec = zeros(params.outer.max_its + 1,1);
        if  nargin==10
            [epsk, sigk, muk] = coarse2fine(m_k, params);
            if all(size(epsk)==size(m_true{1}))
                e_vec(1) = e_vec(1) + norm(epsk - m_true{1});
            end
            if all(size(sigk)==size(m_true{2}))
                e_vec(1) = e_vec(1) + norm(sigk - m_true{2});
            end
            if all(size(muk)==size(m_true{3}))
                e_vec(1) = e_vec(1) + norm(muk - m_true{3});
            end

        end

    end
end

% Step-lengths
if nnz(nuis_indx) > 0 && strcmp(params.line.solver, 'nuisance')
    stepl = zeros(2, params.outer.max_its);
else
    stepl = zeros(1,params.outer.max_its);
end

% Memory for current it (loses 1 if bad step)
mem_k = mem;

% ----------------------------------------------------------------------------
% Perform L-BFGS optimisation procedure
% ----------------------------------------------------------------------------

% Stop if either gradient or residual reach predefined tol
% Do while (||k==1) to force always one iteration
while ( g(k) > params.outer.g_tol  && resvec(k)/resvec(1) > params.outer.r_tol ) || k==1 || dont_exit
    fprintf('Beginning iteration %i:\n',k)

    % ------------------------------------------------------------------------
    % Implicit Hessian approximation
    disp('  calculating descent direction...')

    if k==1
        if isfield(params.outer, 'gamma0')
            gamma_k = params.outer.gamma0;
        else
            gamma_k = 1.;
        end
    else
        gamma_k = ( S(:,m_ind(k-1)).' * Y(:,m_ind(k-1)) )/ ...
            ( Y(:,m_ind(k-1)).' * Y(:,m_ind(k-1)) );
    end


    % 2-loop recursions for implicit Hessian approximation
    q = G_new;

    % ---------------
    % Loop 1
    for i=k-1:-1:max(k-mem_k,1)
        alpha_i(i) = S(:,m_ind(i)).' * q/( Y(:,m_ind(i)).'*S(:,m_ind(i)) );
        q = q - alpha_i(i) * Y(:,m_ind(i));
    end

    % Initial Hessian ----------------------
    % Output p_k approximates H * G_new
    if isfield(params.outer,'initial_Hess') && k <= mem_k
        % Initial Hess approxn
        H0_sup = params.outer.initial_Hess;

        % supplied as lBFGS params
        if isstruct(H0_sup)
            % Calculate with 2-loop reccursion
            p_k = inv_Hess_vec_prod(H0_sup.S, H0_sup.Y, q, ...
                spdiags(1./H0_sup.H0, 0, length(m_k), length(m_k)));

            H0_sup = spdiags(H0_sup.H0, 0, length(m_k), length(m_k));

        elseif isnumeric(H0_sup)

            % Supplied as matrix or vector
            if size(H0_sup,1)==length(m_k) && size(H0_sup,2)==length(m_k)
                % Initial Hess approxn
                p_k = cgs(H0_sup, q);


            elseif size(H0_sup,1)==length(m_k) && size(H0_sup,2)==1
                % Supplied as leading diagonal
                p_k = q./H0_sup;
                H0_sup = spdiags(H0_sup, 0, length(m_k), length(m_k));

            end

        else
            warning('supplied initial Hessian size does not match')
            p_k = gamma_k*q;
        end

        % Ensure satisfies weak quasi-Newton in later iterates
        % i.e. s_{k-1}^T H0k s_{k-1} = s_{k-1}^T y_{k-1}
        if k > 1
            p_k = p_k * ( S(:,m_ind(k-1)).' * Y(:,m_ind(k-1)) )/ ...
            ( S(:,m_ind(k-1)).' *H0_sup* S(:,m_ind(k-1)) );
        end


        % Scale with gamma_k (assumes initial Hessian becomes less accurate
        % with each iteration)
        scale = (mem_k + 1 - k)/mem_k;
        p_k = scale * p_k + (1-scale) * gamma_k * q;
    else
        p_k = gamma_k * q;
    end

    % ---------------------
    % Loop 2
    for i=max(1,k-mem_k) : k-1
        beta = Y(:,m_ind(i)) .' * p_k/( Y(:,m_ind(i)).' * S(:,m_ind(i)) );
        p_k = p_k + S(:,m_ind(i)) * (alpha_i(i) - beta);
    end

    % Invalid descent direction/inf or NaN found
    if nnz(nuis_indx)>0
        p_targ = zeros(size(p_k));
        p_targ(targ_indx) = p_k(targ_indx);
    else
        p_targ = p_k;
    end

    if any(isnan(p_k)) || any(isinf(p_k)) || ...
            (dot(G_new, p_k) <=0 && dot(G_new, p_targ) <=0)

        fprintf('Invalid lBFGS descent direction, iteration %i. Dropping last (s, y) pair...\n', k)
     ;

        p_k_try = inv_Hess_vec_prod(S(:,m_ind(min(1, k-mem):k-1)),...
            Y(:,m_ind(max(1, k-mem):k-1)), G_new, gamma_k);

        % Choose direction
        if dot(G_new, p_k_try) <=0

            Flag = 3;
            fprintf('Invalid lBFGS descent direction, iteration %i. Using steepest descent\n', k);
            p_k = G_new;

        else
            p_k = p_k_try;
        end

    end



    % ---------------------------------------------------------------------
    % Perform line search
    disp('  performing linesearch...')



    % Initial and max step lengths
    d_phi_0_old = d_phi_0;
    d_phi_0 = dot(G_new, -p_k);

    if k==1
        alpha_0 = -resvec(1)/(1.2*d_phi_0);

    else

        % Choose max of:
        %  Interpolated quadratic between F_k-1, F_k and grad F_k^T p_k;
        %  O(1) change in cost same as previous step
        alpha_0 = max(stepl(1,k-1) * d_phi_0_old/d_phi_0,...
                      2*(resvec(k) - resvec(k-1))/d_phi_0);

    end

    % Safeguard a minimum step
    if isfield(params.line, 'delta_min')
        if alpha_0<params.line.delta_min
            alpha_0 = 10*params.line.delta_min;
        end
    elseif alpha_0 <= 0.001
        alpha_0 = 0.01;
    end

    % force 1 if possible
    alpha_0 = min(1, 1.05*alpha_0);


    % max step: min of  suff. decrease line intersects 0, and user defined
    if isfield(params.line, 'c1')
        alpha_max = -resvec(k)/(1.2*params.line.c1*d_phi_0);
    else
        alpha_max = -resvec(k)/(1.2*1e-3 * d_phi_0);
    end
    if isfield(params.line, 'alpha_max')
        alpha_max = min(alpha_max, params.line.alpha_max);
    end

    % Check box constraints
    [alpha_max_phys, m_fixed] = max_move(m_k, -p_k, mdl, params);
    p_k(m_fixed) = 0;
    alpha_max = min(alpha_max, alpha_max_phys);

    % Ensure first step less than alpha_max
    if alpha_max < alpha_0
        alpha_0 = alpha_max/2;
    end

    % Perform linesearch, pass new cost and gradient
    [alpha, resvec(k+1), G_new]  = lsearch(m_k, -p_k, d, alpha_0, ...
        alpha_max, F, resvec(k), G_new, mdl, params, CFIEp);

    stepl(:,k) = alpha.';

    % ----------------------------------------------------------------------
    % Update solution
    % No nuisance params
    if length(alpha)==1
        m_k = m_k - alpha * p_k;
        S(:,m_ind(k)) = -alpha * p_k;

    % Nuisance params
    elseif length(alpha)==2
        m_k(targ_indx) = m_k(targ_indx) - alpha(1)*p_k(targ_indx);
        m_k(nuis_indx) = m_k(nuis_indx) - alpha(2)*p_k(nuis_indx);

        S(targ_indx, m_ind(k)) = - alpha(1)*p_k(targ_indx);
        S(nuis_indx, m_ind(k)) = - alpha(2)*p_k(nuis_indx);

    else
        error('line search returned a non scalar step length (may return 1 or 2 scalars in case of
     nuisance parameters)');
    end



    % ----------------------------------------------------------------------
    % Change parameter weighting
    % Either reweight tol or other exit conditions met
    if (norm(S(:,m_ind(k)))/norm(m_0) < reweight_tol*params.outer.stag ||...
            resvec(k) - resvec(k+1) < reweight_tol*params.outer.decrease_tol  ||  ...
            resvec(k)/resvec(1) <= reweight_tol*params.outer.r_tol ||...
            ~rem(k, reweight_its) )...
           && weight_ii < size(omega_weight,1)

        weight_ii = weight_ii + 1;
        params.reg.omega_weight = omega_weight(weight_ii,:);

        %
        disp('  Frequencies reweighted, recalculating cost and gradient...')
        [resvec(k+1), ~,~,~, G_new] = F(m_k, d,mdl,  params, CFIEp);
        g(k+1) = norm(G_new);

        %
        dont_exit = true;

    end

    % ------------------------------------------------------------
    % Reparameterise
    % Occurs now so that next descent direction calcn is on refined grid.
    % Next descent direction will be the same for this parameterisation
    if refinement
        if (norm(S(:,m_ind(k)))/norm(m_0) < ...
                                refine_tol*params.outer.stag ||...
                resvec(k) - resvec(k+1) < ...
                                refine_tol*params.outer.decrease_tol || ...
                resvec(k+1)/resvec(1) <= ...
                                refine_tol*params.outer.r_tol ||...
                ~rem(k,refine_its)) && refine_it <= max_refine

            disp('Refinement tolerance met, refining coarse inversion grid...')

            % New disc pts
            params_new = parameterise_model('nearest', mdl, params, ...
                                            params.mdl.n_param(2), ...
                                            params.mdl.n_param(3),...
                                      params.outer.refine_pts{refine_it});


            % Nuisance params
            if isfield(params.mdl, 'nuisance')
                nuis = zeros(length(m_k),1);
                nuis(nuis_indx) = 1;
                nuis_indx = find(reparameterise(nuis, params, params_new));
                params_new.mdl.nuisance = nuis_indx;

            end

            % Reparameterise
            p_k = reparameterise(p_k, params, params_new);
            m_k = reparameterise(m_k, params, params_new);
            S = reparameterise(S, params, params_new);
            Y = reparameterise(Y, params, params_new);
            m_0 = reparameterise(m_0, params, params_new);


            % Recalc cost and gradient on new grid
            [resvec(k+1), ~,~,~, G_new] = F(m_k, d,mdl,  params_new, CFIEp);
            g(k+1) = norm(G_new);

            if isfield(params.mdl, 'nuisance')
                targ_indx = setdiff(1:length(m_k) , nuis_indx);
            end


            if exist('G_old','var')
```

```
                G_old = reparameterise(G_old, params, params_new);
            end

            if exist('M', 'var')
                M = reparameterise(M, params, params_new);
            end


            % Lose old params
            params = params_new;

            % Force next iteration
            dont_exit = true;

            %
            refine_it = refine_it + 1;

        end

    end


    % ---------------------------------------------------------------------------
    % Actual error for known true solution
    if nargout>=6
        % Actual error
        if nargin==10 && nargout ==10
            if  nargin==10
                [epsk, sigk, muk] = coarse2fine(m_k, params);
                if all(size(epsk)==size(m_true{1}))
                    e_vec(k+1) = e_vec(k+1) + norm(epsk - m_true{1});
                end
                if all(size(sigk)==size(m_true{2}))
                    e_vec(k+1) = e_vec(k+1) + norm(sigk - m_true{2});
                end
                if all(size(muk)==size(m_true{3}))
                    e_vec(k+1) = e_vec(k+1) + norm(muk - m_true{3});
                end

            end
        end

        % Solution history
        M(:,k+1) = m_k;

    end

    g(k+1) = norm(G_new);

    fprintf('  Completed iteration %i, normalised residual value %6.4f, norm of gradient %6.4f\n',k,resvec(k+1)/resvec(1),norm(G_new))


    % --------------------------------------------------------------------------
    % Check if require damping, update differences in gradient
    d_phi_alpha = dot(G_new, p_k);
    if (abs(d_phi_alpha) > -params.line.c2*d_phi_0 && strcmp(damping, 'auto')) ||...
        strcmp(damping, 'on')

        % If nuisance params exist, damp seperately
        if length(alpha)==2
            % Requires damping, calc replacement y_k
            p_k_tn = zeros(size(p_k));
            p_k_tn(targ_indx) = alpha(1)*p_k(targ_indx);
            p_k_tn(nuis_indx) = alpha(2)*p_k(nuis_indx);

            H_sk = Hess_vec_prod(S(:,m_ind(max(1,k - mem_k - 1): k-1)),...
            Y(:,m_ind(max(1,k - mem_k - 1): k-1)),...
            -p_k_tn, gamma_k);


            % Target parameters
           sk_yk = dot(-alpha(1)*p_k(targ_indx), G_new(targ_indx) - G_old(targ_indx));

           sk_H_sk = dot(-alpha(1)*p_k(targ_indx), H_sk(targ_indx));

           % Damping factor (1 is no damping)
            if sk_yk >= 0.2*sk_H_sk
                theta_k_t = 1;

            else
                theta_k_t = 0.8*sk_H_sk/(sk_H_sk - sk_yk);

            end

            % Nuisance parameters
            sk_yk = dot(-alpha(2)*p_k(nuis_indx), G_new(nuis_indx) - G_old(nuis_indx));
            sk_H_sk = dot(-alpha(2)*p_k(nuis_indx), H_sk(nuis_indx));


           % Damping factor (1 is no damping)
            if sk_yk >= 0.2*sk_H_sk - eps % -eps in case of 0 step
                theta_k_n = 1;

            else
                theta_k_n = 0.8*sk_H_sk/(sk_H_sk - sk_yk);

            end

            % Apply damped update seperately
            Y(nuis_indx, m_ind(k)) = theta_k_n*(G_new(nuis_indx) - G_old(nuis_indx)) + (1 - theta_k_n)*H_sk(nuis_indx);
            Y(targ_indx, m_ind(k)) = theta_k_t*(G_new(targ_indx) - G_old(targ_indx)) + (1 - theta_k_t)*H_sk(targ_indx);

        else
            % Requires damping, calc replacement y_k
            H_sk = Hess_vec_prod(S(:,m_ind(max(1,k - mem_k - 1): k-1)),...
            Y(:,m_ind(max(1,k - mem_k - 1): k-1)),...
            -alpha*p_k, gamma_k);

            sk_H_sk = dot(-alpha*p_k, H_sk);

            % Single parameter set
            sk_yk = dot(-alpha*p_k, G_new - G_old);

            % Damping factor (1 is no damping)
            if sk_yk >= 0.2*sk_H_sk
                theta_k = 1;

            else
                theta_k = 0.8*sk_H_sk/(sk_H_sk - sk_yk);

            end

            % Apply damped update
            Y(:,m_ind(k)) = theta_k*(G_new - G_old) + (1 - theta_k)*H_sk;

        end


    else
```

```
        % No damping
        Y(:,m_ind(k)) = G_new - G_old;

    end

    % ----------------------------------------------------------------------
    % Stop conditions (minimum found is the while condition)


    % ----------------------------------------------------------------------
    % Stagnation (x_k - x_k-1 = -alpha * p_k)
    if norm(S(:,m_ind(k)))/norm(m_0) < params.outer.stag && ~dont_exit
        disp('No sufficient change in solution');
        Flag = 1;

        % Stop or delay
        if delayed >= stop_delay
            break
        else
            delayed = delayed + 1;
        end


    % ----------------------------------------------------------------------
    % Max its
    elseif k==params.outer.max_its
        Flag = 2;

        break

    % ----------------------------------------------------------------------
    % Insufficient decrease
    elseif resvec(k) - resvec(k+1) < params.outer.decrease_tol && ~dont_exit
        disp('No sufficient decrease')
        Flag = 1;

        % Reset to previous (in case of increase in cost)
        if resvec(k+1) > resvec(k)
            m_k = m_k - S(:,m_ind(k));
            disp('Cost function increased, returning previous iterate')
            break
        end

        if delayed >= stop_delay
            break
        else
            delayed = delayed + 1;

        end


    % ----------------------------------------------------------------------
    % Else update old values
    else

        % Update k
        k = k+1;
        G_old = G_new;

        dont_exit = false;
        delayed = 0;

    end

    % Discard current iteration if no change to solution
    % Only occurs if another iteration forced (e.g. reparameterised)
    if norm(S(:,m_ind(k))) < 1e1 * eps
        mem_k = mem_k-1;
    else
        mem_k = mem;
    end


    % ----------------------------------------------------------------------
    % Reduce/increase epsilon for TV norm
    if isfield(params.reg,'epsilon') && isfield(params.reg,'continuation')
        if delayed<=0 && params.reg.epsilon > 10*eps
            params.reg.epsilon = params.reg.epsilon * params.reg.continuation;
            params.reg.continuation = 0.99*params.reg.continuation;
        else
            params.reg.epsilon = params.reg.epsilon * 2;
        end
    end

    % Increase nuisance max step if is field
    if isfield(params.line, 'nuis_max_step') && isfield(params.line, 'nuis_step_mod')
        params.line.nuis_max_step = params.line.nuis_max_step * params.line.nuis_step_mod;
    end


end


if nargout>=6
    M = M(:,1:k+1);
    if nargout == 10
            if all(size(m_k)==size(m_true))
                [epsk, sigk, muk] = coarse2fine(m_k, params);
                if all(size(epsk)==size(m_true{1}))
                    e_vec(k+1) = e_vec(k+1) + norm(epsk - m_true{1});
                end
                if all(size(sigk)==size(m_true{2}))
                    e_vec(k+1) = e_vec(k+1) + norm(sigk - m_true{2});
                end
                if all(size(muk)==size(m_true{3}))
                    e_vec(k+1) = e_vec(k+1) + norm(muk - m_true{3});
                end

            end
    end

resvec = resvec(1:k+1);
g = g(1:k+1);

end
fprintf('LBFGS optimisation completed with final residual %6.3f and flag %i\n',resvec(k),Flag)



end
```

Bibliography

[1] A. Adler, WRB. Lionheart, and F. Watson. Shape sensitivity of EIT and GPR. Working title.

[2] Yogi Ahmad. *A robust and efficient iterative method for the numerical solution of the Helmholtz equation*. PhD thesis, Technishe Universiteit Delft, December 2005.

[3] Mark Ainsworth. Discrete dispersion relation for hp-version finite element approximation at high wave number. *SIAM Journal on Numerical Analysis*, 42(2):553–575, 2004.

[4] Mark Ainsworth and Joe Coyle. Hierarchic hp-edge element families for maxwell's equations on hybrid quadrilateral/triangular meshes. *Computer Methods in Applied Mechanics and Engineering*, 190(49-50):6709–6733, 2001.

[5] Mehiddin Al-Baali and Lucio Grandinetti. On the bahaviour of damped quasi-Newton methods for unconstrained optimization. *Iranian Journal of Operations Research*, 3(1):1–10, 2012.

[6] Mehiddin Al-Baali, Lucio Grandinetti, and Ornella Pisacane. Damped techniques for limited memory BFGS method for large-scale optimization. *Journal of Optimisation Theory and Applications*, 161:688–699, May 2014.

[7] Habib Ammari, Ekaterina Iakovleva, Dominique Lesselier, and Ga´ele Perusson. MUSIC-type electromagnetic imaging of a collection of small three-dimensional bounded inclusions. *SIAM Journal on Scientific Computing*, 29(2):674–709, 2007.

[8] Habib Ammari and Hyeonbae Kang. High-order terms in the asymptotic expansions of the steady-state voltage potentials in the presence of conductivity inhomogeneities of small diameter. *SIAM Journal on Mathematical Analysis*, 34(5):1152–1166, 2003.

[9] Habib Ammari and Hyeonbae Kang. *Polarization and Moment Tensors, with Applications to Inverse Problems and Effective Medium Theory.* Applied Mathematical Sciences. Springer-Verlag, New York, 2007.

[10] Habib Ammari, Hyeonbae Kang, and Kyoungsun Kim. Polarization tensors and effective properties of anisotropic composite materials. *Journal of Differential Equations*, 215(2):401 – 428, 2005.

[11] Habib Ammari, Hyeonbae Kang, and Mikyoung Lim. Polarization tensors and their applications. *Journal of Physics: Conference Series*, 12(1):13, 2005.

[12] Habib Ammari, Hyeonbae Kang, Gen Nakamura, and Kazumi Tanuma. Complete asymptotic expansions of solutions of the system of elastostatics in the presence of an inclusion of small diameter and detection of an inclusion. *Journal of Elasticity and the Physical Science of Solids*, 67(2):97–129, 2002.

[13] Habib Ammari and Jin Keun Seo. An accurate formula for the reconstruction of conductivity inhomogeneities. *Advances in Applied Mathematics*, 30(4):679 – 705, 2003.

[14] Habib Ammari, Michael S Vogelius, and Darko Volkov. Asymptotic formulas for perturbations in the electromagnetic fields due to the presence of inhomogeneities of small diameter II. the full Maxwell equations. *Journal de Mathmatiques Pures et Appliques*, 80(8):769 – 814, 2001.

[15] Amsalu Y Anagaw and Mauricio D Sacchi. Edge-preserving seismic imaging using the total variation method. *Journal of Geophysics and Engineering*, 9:138–146, 2012.

[16] Aleksandr Y Aravkin and Tristan van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Problems*, 28, 2012.

[17] S R Arridge. Optical tomography in medical imaging. *Inverse Problems*, 15(2):R41, 1999.

[18] Amir Asnaashari, Romain Brossier, Stphane Garambois, Franois Audebert, Pierre Thore, and Jean Virieux. Regularized seismic full waveform inversion with prior model information. *Geophysics*, 78(2):R25–R36, 2013.

[19] Saman Babaie-Kafaki, Masoud Fatemi, and Nezam Mahdavi-Amiri. Two effective hybrid conjugate gradient algorithms based on modified bfgs updates. *Numerical Algorithms*, 58(3):315–331, 2011.

[20] C.E. Baum. Identification of hidden mines. *IEEE*, pages 692–695, Sept 2007.

[21] Alvin Bayliss, Max Gunzburger, and Eli Turkel. Boundary conditions for the numerical solution of elliptic equations in exterior regions. *SIAM Journal of Applied Maths*, 42, April 1982.

[22] Bradley M. Bell and James V. Burke. Algorithmic differentiation of implicit functions and optimal values. In Christian H. Bischof, H. Martin Bücker, Paul Hovland, Uwe Naumann, and Jean Utke, editors, *Advances in Automatic Differentiation*, volume 64 of *Lecture Notes in Computational Science and Engineering*, pages 67–77. Springer-Verlag Berlin Heidelberg, 2008.

[23] Hafedh Ben-Hadj-Ali, Stéphane Operto, and Jean Virieux. 3D frequency-domain full-waveform tomography based on a domain decomposition forward problem. In *78th Annual International Meeting*, pages 1945–1949. SEG, Las Vegas, 2008.

[24] Jean-Pierre Berrenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114:185–200, 1994.

[25] Mario Bertero and Patrizia Boccacci. *Introduction to Inverse Problems in Imaging.* Institiute of Physics Publishing, Bristol and Philadelphia, 1998.

[26] G. Beylkin. Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized radon transform. *Journal of Mathematical Physics*, 26(1):99–108, January 1985.

[27] Liliana Borcea. Electrical impedance tomography. *Inverse Problems*, 18(6):R99, 2002.

[28] Andrea Borsic. *Regularisation methods for imaging from electrical measurements.* Doctor of philosophy, School of Engineering, Oxford Brookes University, July 2002.

[29] Romain Brossier, Stèphane Operto, and Jean Virieux. Which data residual norm for robust elastic frequency-domain waveform inversion? *Geophysics*, 75:R37–R46, May-June 2010.

[30] Sebastian Busch, Jan van der Kruk, Jutta Bikowski, and Harry Vereecken. Combined effective wavelet estimation and full-waveform inversion of GPR data. In *Proceedings of the 2011 6th International Workshoop on Advanced Ground Penetrating Radar (IWAGPR)*, pages 1–9. IEEE, 2011.

[31] Sebastian Busch, Jan van derk Kruk, and Harry Vereecken. Improved characterization of fine-textured soils using on-ground GPR full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 52(7):3947–3958, July 2014.

[32] S. Butzer, A. Kurzmann, and T. Bohlen. 3D elastic full-waveform inversion of small-scale heterogeneities in transmission geometry. *Geophysical Prospecting*, 61:1238–1251, 2013.

[33] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, September 1995.

[34] D J Cedio-Fengya, S Moskow, and M S Vogelius. Identification of conductivity imperfections of small diameter by boundary measurements. continuous dependence and computational reconstruction. *Inverse Problems*, 14(3):553, 1998.

[35] Chasles. Nouvelle solution du probléme de l'attraction d'un ellipsoid hétérogène sur un point extérieur. *Liouville's Journal*, December 1840.

[36] M. Cheney and B. Borden. *Fundamentals of Radar Imaging.* Society for Industrial and Applied Mathematics, 2009.

[37] W. C. Chew and J. M. Jin. Perfectly matched layers in the discretized space: An analysis and optimization. *Electromagnetics*, 16:325–340, 1996.

[38] Weng Cho Chew and William H. Weedon. A 3d perfectly matched medium from modified maxwell's equations with stretched coordinates. *Microwave and Optical Technology Letters*, 7(13):599–604, September 1994.

[39] J.F. Clearbout. *Imaging the Earth's Interior.* Blackwell Scientific Publications, London, 1985.

[40] Dennis A. Cooke and William A. Schneider. Generalized linear inversion of reflection seismic data. *Geophysics*, 48:665–676, June 1983.

[41] David Daniels. *Ground Penetrating Radar.* The Institution of Electrical Engineers, 2nd edition, 2004.

[42] David Daniels. Current technology capability of gpr landmine detection. Seminar to Find A Better Way researchers, June 2015.

[43] David Daniels, J´urgen Braunstein, and Michael Nevard. Using MINEHOUND in Cambodia and Afghanistan. *The Journal of ERW and Mine Action*, 2014.

[44] David J. Daniels. *Ground Penetrating Radar for Baried Landmine and IED Detection*, pages 89–111. NATO Science for Peace and Security Series B: Physics and Biophysics. Springer Netherlands, 2008.

[45] D.J. Daniels, D.J. Gunton, and H.F. Scott. Introduction to subsurface radar. *Radar and Signal Processing, IEE Proceedings F*, 135(4):278–320, Aug 1988.

[46] George Dassios and Ralph Kleinman. *Low Frequency Scattering.* Oxford University Press, 2000.

[47] M V de Hoop, H Smith, G Uhlmann, and R D van der Hilst. Seismic imaging with the generalized radon transform: a curvelet transform perspective. *Inverse Problems*, 25:1–21, 2009.

[48] Maarten V. de Hoop and Norman Bleistein. Generalized radon transform inversions for reflectivity in anisotropic elastic medica. *Inverse Problems*, 13:669–690, 1997.

[49] Michael Dell, editor. *Jane's Mines and Mine Clearance.* Colin King, 2011-2012.

[50] Robert C. Doheny, Sean Burke, and Roger Cresci. Handheld standoff mine detection system (HSTAMIDS) field evaluation in thailand. In *Proceedings of Detection and Remediation Technologies for Mines and Minelike Targets X*. SPIE, June 2005.

[51] Amanda L. Higginbotham Duque, William Lee Perry, and Christine M. Anderson-Cook. Complex microwave permittivity of secondary high explosives. *Propellants, Explosives, Pyrotechnics*, 39:275–283, 2014.

[52] Maire E. and Withers P. J. Quantitative x-ray tomography. *International Materials Reviews*, 59(1):1–43, 2014.

[53] Howard C. Elman, David J. Silvester, and Andrew J. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics.* Oxford University Press, 2005.

[54] Y. A . Erlangga, C. Vuik, and C. W. Oosterlee. On a class of preconditioners for solving the discrete helmholtz equation. *Applied Numerical Mathematics*, pages 409 – 425, 2004.

[55] Jacques R. Ernst, Alan G. Green, Hansruedi Maurer, and Klaus Holliger. Application of a new 2D time-domain full-waveform inversion scheme to crosshole radar data. *Geophysics*, 72(5):J53–J64, September-October 2007.

[56] Jacques R. Ernst, Hansruedi Maurer, Alan G. Green, and Klaus Holliger. Full-waveform inversion of crosshole radar data based on 2-D finite difference time-domain solutions of maxwell's equations. *IEEE Transactions on Geoscience and Remote Sensing*, 45(9):2807–2828, September 2007.

[57] O.G. Ernst and M.J. Gander. Why it is difficult to solve helmholtz problems with classical iterative methods. In Ivan G. Graham, Thomas Y. Hou, Omar Lakkis, and Robert Scheichl, editors, *Numerical Analysis of Multiscale Problems*, volume 83 of *Lecture Notes in Computational Science and Engineering*, pages 325–363. Springer Berlin Heidelberg, 2012.

[58] Xuan Feng and Motoyuki Sato. Pre-stack migration applied to GPR for landmine detection. *Inverse Problems*, 20:99–115, 2004.

[59] Avner Friedman and Michael Vogelius. Identification of small inhomogeneities of extreme conductivity by boundary measurements: a theorem on continuous dependence. *Archive for Rational Mechanics and Analysis*, 105(4):299–326, 1989.

[60] Katsushisa Furuta and Jun Ishikawa, editors. *Anti-personnel Landmine Detection for Humanitarian Demining: The Current Situation and Future Direction for Japanese Research an Development.* Springer London, 2009.

[61] Charankumar Godavarthi, Ting Zhang, Guillaume Maire, Patrick C. Chaumet, Hugues Giovannini, Anne Talneau, Kamal Belkebir, and Anne Sentenac. Super-resolution with full-polarized tomographic diffractive microsopy. *Journal of the Optical Society of America*, 32(2):287–292, February 2015.

[62] Tom Goldstein and Stanley Osher. The split bregman method for L1-regularized problems. *SIAM Journal of Imaging Sciences*, 2(2):323–343, 2009.

[63] Roberto D. Graglia. On the numerical integration of the linear shape functions times the 3-D Green's function or its gradient on a plane triangle. *IEEE Transactions on Antennas and Propagation*, 41(10):1448–1455, October 1993.

[64] Anne Greenbaum. *Iterative Methods for Solving Linear Systems.* SIAM, 1997.

[65] Mines Advisory Group. Humanitarian demining. Seminar to Find A Better Way researchers, June 2012. MIRAN Landmines Workshop.

[66] A.V. Guglielmi. On the leontovich boundary condition in geoelectromagnetism. *Physics of the Solid Earth*, 45(9), 2009.

[67] Jacques Hadamard. Sur les problèmes aux dérivés partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.

[68] Martin Hanke. Limitations of the l-curve method in ill-posed problems. *BIT Numerical Mathematics*, 36(2):287–301, 1996.

[69] P.C. Hansen. Computation of the singular value expansion. *Computing*, 40(3):185–199, 1988.

[70] Per Christian Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34(4):561–580, December 1992.

[71] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems.* SIAM, 1998.

[72] Per Christian Hansen and Dianne Prost OLeary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computing*, 14(6):1487–1503, 1993.

[73] Zimeng Hiang, Zhaofa Zeng, Jing Li, Fengshan Liu, and Fengshou Wu. Simulation and analysis of GPR signal based on stochastic media model. In *14th International Conference on Ground Penetrating Radar (GPR)*, June 2012.

[74] B. Hofmann. *Regularization for Applied Inverse and Ill-Posed Problems.* Teubner, Stuttgart, 1986.

[75] Marianne Houbiers, Edward Wiarda, Joachim Mispel, Dmitry Nikolenko, Denes Vigh, Bjørn-Egil Knudsen, Mark Thompson, and David Hill. 3D full-waveform inversion at mariner – a shallow north sea reservoir. In *83rd Annual International Meeting.* SEG, Las Vegas, 2013.

[76] Wenyi Hu, Aria Abubakar, and Tarek M. Habashy. Simultaneous multifrequency inversion of full-waveform seismic data. *Geophysics*, 74(2), March–April 2009.

[77] Yunqing Huang, Jichun Li, Chao Wu, and Wei Yang. Superconvergence analysis for linear tetrahedral edge elements. *Journal of Scientific Computing*, 62(1):122–145, 2015.

[78] Zimeng Jiang, Zhaofa Zeng, Jing Li, Fengshan Liu, and Wenben Li. Simulation and analysis of GPR signal based on stochastic media model with an ellipsoidal autocorrelation function. *Journal of Applied Geophysics*, 99:91–97, 2013.

[79] Jian-Ming Jin and John L. Volakis. A hybrid finite element method for scattering and radiation by microstrip patch antennas and arrays residing in a cavity. *IEEE Transactions on Antennas and Propagation*, 39(11), November 1991.

[80] Jianming Jin. *The Finite Element Method in Electromagnetics.* John Wiley and Sons, second edition edition, 2002.

[81] Jari Kaipio and Erkki Somersalo. Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198(2):493 – 504, 2007. Special Issue: Applied Computational Inverse Problems.

[82] Mirza Karamehmedović. Bounds on the stably recoverable information for the inverse source problem. Pre-print, 2014.

[83] R. E. Kleinman and T. B. A. Senior. Rayleigh scattering. In *Low and high frequency asymptotics*, volume 2 of *Acoustic electromagnetic and elastic wave scattering*, chapter 1, pages 1–70. Noth-Holland, 1986.

[84] R.E. Kleinman. The Rayleigh region. *Proceedings of the IEEE*, 53(8):848–856, Aug 1965.

[85] A. Klotzsche, J. van der Kruk, N. Linde, and J. Doetsch. 3D characterization of an aquifer using full-waveform inversion and amplitude analysis. In *Proceedings of the 2013 7th International Workshop on Advanced Ground Penetrating Radar (IWAGPR).* IEEE, 2013.

[86] A. Klotzsche, J. van der Kruk, G. A. Meles, J. A. Doetsch, H. Maurer, and N. Linde. Full-waveform inversion of crosshole ground penetrating radar data to characterize a gravel aquifer close to the Thur River, Switzerland. In *13^{th} International Conference on Ground Penetrating Radar*, 2010.

[87] Kim Knudsen, Matti Lassas, Jennifer L. Mueller, and Samuli Siltanen. Dbar method for electrical impedance tomography with discontinuous conductivities. *SIAM Journal on Applied Mathematics*, 67(3):893–913, 2007.

[88] Seiichiro Kuroda, Mutsuo Takeuchi, and Hee Joon Kim. Full-waveform inversion algorithm for interpreting crosshole radar data: a theoretical approach. *Geosciences Journal*, 11(3):211–217, September 2007.

[89] V. Lauer. New approach to optical diffraction tomography yielding a vector equation of diffraction tomography and a novel tomographic microscope. *Journal of Microscopy*, 205(2):165–176, 2002.

[90] F. Lavoué, R. Brossier, S. Garambois, and J. Virieux. Permittivity and conductivity reconstruction by full waveform inversion of GPR data using the L-BFGS-B algorithm. In *Proceedings of the 18th European Meeting of Environmental and Engineering Geophysics, Paris, France*. Near Surface Geoscience, September 2012.

[91] F. Lavoué, R. Brossier, S. Garambois, J. Virieux, and L. Métivier. 2D full waveform inversion of GPR surface data: permittivity and conductivity imaging. In *Proceedings of the 2013 7th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*. IEEE, 2013.

[92] François Lavoué. *2D full waveform inversion of ground penetrating radar data, towards multiparameter imaging from surface data*. Sciences dela terre, de l'univers et de l'environement, Université de Grenoble, 2014.

[93] Paul Ledger. Different forms of p-s tensor. Private communication, January 2015.

[94] Paul D. Ledger and William R. B. Lionheart. Characterizing the shape and material properties of hidden targets from magnetic induction data. *IMA Journal of Applied Mathematics*, 2015.

[95] Paul D. Ledger and William R. B. Lionheart. The perturbation of electromagnetic fields at distances that are large compared with the object's size. *IMA Journal of Applied Mathematics*, 80(3):865–892, 2015.

[96] Paul David Ledger. *An hp-Adaptive Finite Element Procedure for Electromagnetic Scattering Problems*. Doctor of philosophy, Department of Civil Engineering, University of Wales, Swansea, November 2001.

[97] Wah June Leong and Chuei Yee Chen. A class of diagonal preconditioners for limited memory bfgs method. *Optimization Methods and Software*, 28(2):379–392, 2013.

[98] MA Leontovich. Investigations on radiowave propagation, part ii. *Printing House Academy of Science*, 1948.

[99] Na Li, Yousef Saad, and Edmond Chow. Crout versions of ILU for general sparse matrices. *SIAM Journal on Scientific Computing*, 25(2):716–728, 2003.

[100] Jian Liu and Jian-Ming Jin. A highly effective preconditioner for solving the finite element–boundary integral matrix equation of 3-d scattering. *IEEE Transactions on Antennas and Propagation*, 50(9):1212–1221, September 2002.

[101] Olga Lopera, Evert C. Slob, Nada Milisavljević, and Sébastien Lambot. Filtering soil surface and antenna effects from GPR data to enhance landmine detection. *IEEE Transactions on Geoscience and Remote Sensing*, 45, March 2007.

[102] A. E. H. Love. The integration of the equations of propagation of electric waves. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 197:pp. 1–45, 1901.

[103] P. Lutz, S. Garambois, and H. Perroud. Influence of antenna configurations for gpr survey: information from polarization and amplitude versus offset measurements. *Geological Society, London, Special Publications*, 211(1):299–313, 2003.

[104] Yong Ma and Dave Hale. Cwp-679 a projected hessian for full waveform inversion.

[105] Yong Ma and Dave Hale. Quasi-newton full-waveform inversion with a projected hessian matrix. *Geophysics*, 77(5):R207–R216, 2012.

[106] Jacquiline MacDonald. *Alternatives for Landmine Detection.* RAND, 2003.

[107] Liam A Marsh, Christos Ktistis, Ari Jrvi, David W Armitage, and Anthony J Peyton. Three-dimensional object location and inversion of the magnetic polarizability tensor at a single frequency using a walk-through metal detector. *Measurement Science and Technology*, 24(4):045102, 2013.

[108] James Clerk Maxwell. On physical lines of force. *The London, Edinburgh and Dubplin Philosophical Magazine and Journal of Science*, March 1861.

[109] James Clerk Maxwell. *A Treatise on Electricity and Magnetism*, volume 2. Clarendon Press, 1873.

[110] Giovani Meles, Jan Van der Kruk, Steward A. Greenhalgh, Jacques R. Ernst, Hansruedi Maurer, and Alan G. Green. A new vector waveform inversion algorithm for simultaneous updating of conductivity and permittivity parameters from combination crosshole/borehole-to-surface GPR data. *IEEE Transactions on geoscience and remote sensing*, 48(9):3391–3407, September 2010.

[111] Giovanni Meles, Stewart Greenhalgh, Alan Green, Hansruedi Maurer, and Jan van der Kruk. GPR full-waveform sensitivity and resolution analysis using an FDTD adjoint method. *IEEE Transactions on Geoscience and Remote Sensing*, 50:1881–1896, 2012.

[112] Giovanni Meles, Stewart Greenhalgh, Jan van der Kruk, Alan Green, and Hansruedi Maurer. Taming the non-linearity problem in GPR full-waveform inversion for high contrast media. *Journal of Applied Geophysics*, 73:174–186, 2011.

[113] Giovanni Angelo Meles. *New Developments in Full Waveform Inversion of GPR Data.* Doctor of sciences, Eidgenössische Technische Hochschule, Eth Zurich, 2011.

[114] L. Métivier, F. Bretaudeau, R. Brossier, S. Operto, and J. Virieux. Full waveform inversion and the truncated Newton method: quantitative imaging of complex surface structures. Submitted to Geophysical Prospecting, September 2013.

[115] L. Métivier, R. Brossier, J. Virieux, and S. Operto. Full waveform inversion and the truncated newton method. Pre-print, November 2012.

[116] L. Métivier, R. Brossier, J. Virieux, and S. Operto. The truncated newton method for full waveform inversion. In *2nd Int. Workshop on New Computational Methods for Inverse Problems (NCMIP 2012), Journal of Physics: Conference Series 286*, pages 1–6, 2012.

[117] D. Miller, M.Oristaglio, and G. Beylkin. A new slant on seismic imaging: Migration and integral geometry. *Geophysics*, 52, July 1987.

[118] Peter Monk. A finite element method for approximating the time-harmonic Maxwell equations. *Numerische Mathematik*, 1992.

[119] Peter Monk. *Finite Element Methods for Maxwell's Equations*. Numerical Mathematics and Scientific Computation. Oxford Science Publications, 2003.

[120] Jorge J. Moré and David J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, 20(3):286–307, September 1994.

[121] J. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012.

[122] J.C. Nédélec. A new family of mixed finite elements in 3. *Numerische Mathematik*, 50(1):57–81, 1986.

[123] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, second edition, 2000.

[124] M. Oberröhrmann, A. Klotzsche, H. Vereecken, and J. van der Kruk. Optimization of acquisition setup for cross-hole GPR full-waveform inversion using checkerboard analysis. *Near Surface Geophysics*, 11:197–209, 2013.

[125] Maria Elizabeth G. Ong. Uniform refinement of a tetrahedron. *SIAM Journal of Scientific Computing*, 15(5):1134–1144, September 1994.

[126] Monk P. On the p- and hp-extension of ndlec's curl-conforming elements. *Journal of Computational and Applied Mathematics*, 53(1):117–137, 1994. cited By 29.

[127] V P Palamodov. A uniform reconstruction formula in integral geometry. *Inverse Problems*, 28(6):065014, 2012.

[128] Victor P. Palamodov. A new reconstruction method in integral geometry. Preprint, September 2011.

[129] Victor P. Palamodov. A uniform reconstruction formula in integral geometry. Preprint, November 2011.

[130] Wen peng Wang, Bo Zhao, Xiao jun Liu, and Guang you Fang. Total-variation improved split bregman method for ground penetrating radar image restoration. *Journal of Applied Geophysics*, 99:146–153, 2013.

[131] Andrew F. Peterson. The "interior resonance" problem associated with surface integral equations of electromagnetics: Numerical consequences and a survey of remedies. *Electromagnetics*, 10(3):293–312, 1990.

[132] A. Pica, J. P. Diet, and A. Tarantola. Nonlinear inversion of seismic reflection data in a laterally invariant medium. *Geophysics*, 55(3):284–292, 1990.

[133] R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167:495–503, September 2006.

[134] A.J. Poggio and E.K. Miller. *Integral Equation Solutions of Three-dimensional Scattering Problems*. MBA technical memo: MBAssociates. MB Assoc., 1970.

[135] Pólya and Szegö. *Isopermetric Inequalities in Mathematical Physics*. Annals of Mathematical Studies Number 27. Princeton University Press, Princeston, NJ, 1951.

[136] Nick Polydorides. Electromagnetic inverse problems for nematic liquid crystals and capacitance imaging. *Manchester Institute for Mathematical Sciences sprint*, March 2007.

[137] R. Gerhard Pratt, Changsoo Shin, and G.J. Hicks. Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion. *Geophysical Journal International*, 133(2):341–262, 1998.

[138] Sadsiva M. Rao, Donald R. Wilton, and Allen W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Transactions on Antennas and Propagation*, AP-30(3), May 1982.

[139] C. Ravaut, S. Operto, L. Improta, J. Virieux, A. Herrero, and P. Dell'Aversana. Multiscale imaging of complex structures from multifold wide-aperture seismic data by frequency-domain full-waveform tomography: application to a thrust belt. *Geophysics*, pages 1032–1056, December 2004.

[140] Lord rayleigh. On the incidence of aerial and electric waves upon small obstacles in the form of ellipsoids or elliptic cylinders, and on the passage of electric waves through a circular aperture in a conducting screen. *Philosophical Magazine*, 44:28–52, 1897.

[141] R.E.-Plessix. A helmholtz iterative solver for 3d seismic-imaging problems. *Geophysics*, 72, 2007.

[142] Roger L. Roberts and Jeffrey J. Daniels. Analysis of gpr polarization phenomena. *Journal of Environmental and Engineering Geophysics*, 1(2):139–157, 1996.

[143] Zachary S. Sacks, David M. Kingsland, Robert Lee, and Jin-Fa Lee. A perfectly matched anisotropic absorber for use as an absorbing boundary condition. *IEEE Transactions on Antennas and Propagation*, 43(12):1460–1463, December 1995.

[144] H Sahli, E Nyssen, L van Kempen, and J Cornelis. Feature extraction and classification methods for ultra-sonic and radar mine detection. *IEEE CESA '98 conference*, 4:82–87, April 1998.

[145] Motoyuki Sato and Xuan Feng. GPR migration algorithm for landmines buried in inhomogeneous soil. In *Antennas and Propagation Society International Symposium*, pages 206–209. IEEE, 2005.

[146] Motoyuki Sato, Jun Fujiwara, Zuan Feng, Zheng-Shu Zhou, and Takao Zobayashi. Evaluation of a hand-held GPR MD sensor system (ALIS). Technical report, Tohoku University, 2005.

[147] Timofey Grigorievich Savelyev, Luc van Kempen, Hichem Sahli, Juergen Sachs, and Motoyuku Sato. Investigation of time-frequency features for GPR landmine discrimination. *IEEE Transactions on Geoscience and Remote Sensing*, 45:118–129, January 2007.

[148] M. Schiffer and G. Szegö. Virtual mass and polarization. *Transactions of the American Mathematical Society*, 1949.

[149] Joachim Schöberl. Netgen mesh generator 4.4. Johannes Kepler Universty Linz.

[150] Joachim Schöberl and Sabine Zaglmayr. High order nédélec elements with local complete sequence properties. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, 24(2):374–384, 2005.

[151] T.B.A. Senior. Impedance boundary conditions for imperfectly conducting surfaces. *Applied Scientific Research, Section B*, 8(1):418–436, 1960.

[152] Xin-Qing Sheng, Jian-Ming Jin, Jiming Song, Weng Cho Chew, and Cai-Cheng Lu. Solution of combined-field integral equation using multilevel fast multipole algorithm for scattering by homogeneous bodies. *IEEE Transactions on Anetannas and Propagation*, 46(11):1718–1726, November 1998.

[153] Xin-Qing Sheng, Jian-Ming Jin, Jiming Song, Cai-Cheng Lu, and Weng Cho Chew. On the formulation of hybrid finite-element and boundary integral methods for 3D scattering. *IEEE Transactions on Antennas and Propagation*, 46(3):303–311, March 1998.

[154] Changsoo Shin, Seonghyung Jang, and Dong-Yoo Min. Improved amplitude preservation for prestack depth migration by inverse scattering theory. *Geophysical Prospecting*, 49:5, March 2001.

[155] Wonseok Shin and Shanhui Fan. Choice of the perfectly matched layer boundary condition for frequency-domain maxwell's equations solvers. *Journal of Computational Physics*, 231:3406–3431, 2012.

[156] Shanker Man Shrestha and Ikuo Arai. High resolution image reconstruction by GPR using MUSIC and SAR processing method for landmine detection. Technical report, Electronics Engineering Department, The University of Electrocommunications, 1-5-1 Chofugaoka, Chofu, Tokyo, 2003.

[157] Ilya Silvestrov and Vladimir Tcheverda. SVD analysis in application to full waveform inversion of multicomponent seismic data. In *International Conference on Inverse Problems 2010, Journal of Physics: Conference Series 290*, pages 1–8, 2010.

[158] Katherine M. Simonson. Statistical considerations in designing tests of mine detection systems: I-measures related to the probability of detection. Technical report, Sandia National Laboratories, 1998.

[159] Laurent Sirgue and R. Gerhard Pratt. Efficient waveform inversion and imaging: a strategy for selecting temporal frequencies. *Geophysics*, 69:231–248, January-February 2004.

[160] Erkki Somersalo, David Isaacson, and Margaret Cheney. A linearized inverse boundary value problem for maxwell's equations. *Journal of Computational and Applied Mathematics*, 42:123–136, 1992.

[161] Jiming Song and Weng Cho Chew. Multilevel fast-multipole alforithm for solving combined field integral equations of electromagnetic scattering. *Microwave and Optical Technology Letters*, 10(1):14–19, September 1995.

[162] Juan Song, Qing Huo Lui, Pete Torrione, and Leslie Collins. Two-dimensional and three-dimensional NUFFT migration method for landmine detection using ground-penetrating radar. *IEEE Transactions on Geoscience and Remote Sensing*, 44:1462–1469, June 2006.

[163] J.P. Suárez, P. Abad, A. Plaza, and M.A. Padrón. Computational aspects of the refinement of 3D tetrahedral meshes. *Journal of Computational Methods in Sciences and Engineering*, 5:215–224, 2005.

[164] Anders Sullivan, Raju Damarla, Norbert Geng, Yanting Dong, and Lawrence Carin. Ultrawide-band synthetic aperture radar for detection of unexploded ordnance: Modeling and measurements. *IEEE Transactions on Antennas and Propagation*, 48(9), September 2000.

[165] Y. Sun and J. Li. Time-frequency analysis for plastic landmine detection via forward-looking ground penetrating radar. In *Proceedings of Radar, Sonar and Navigation*, volume 150. IEEE, August 2003.

[166] Chen-To Tai. *Dyadic Green Functions in Electromagnetic Theory.* IEEE Press, 1993.

[167] A. Tarantola. Linearized inversion of seismic reflection data. *Geophysical Prospecting*, 32:998–1015, 1984.

[168] William Carlisle Thacker. The role of the hessian matrix in fitting models to measurements. *Journal of Geophysical Research*, 94(C5):6177–6196, May 1989.

[169] Inc. The Mathworks. Matlab mldivide algorithm for sparse inputs. http://uk.mathworks.com/help/matlab/ref/mldivide.html. Natick, Massachusetts, United States.

[170] William Thomson and Peter Guthrie Tait. *Treatise on Natural Philosophy: Volume 2*. Cambridge University Press, 2 (7 sept. 2010) edition, 1879.

[171] Find a better way. http://www.findabetterway.org.uk.

[172] What is demining? http://school.mech.uwa.edu.au/~jamest/demining/info/what-is.html.

[173] Convention on the prohibition of the use, stockpiling, production and transfer of anti-personnel mines and on their desctruction. www.un.org/Fepts/mine/UNDOcs/ban_trty.htm, September 1997.

[174] Landmine monitor major findings. www.the-monitor.org, 2014.

[175] G Uhlmann. Electrical impedance tomography and caldern's problem. *Inverse Problems*, 25(12):123011, 2009.

[176] H.C. van de Hulst. *Light scattering by small particles*. Dover Publications, Inc. New York, 1957.

[177] H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[178] L. van Kempen, E. Nyssen, H. Sahli, and J Cornelis. Pattern recognition experiments for ultra-sonic and radar ap-mine detection. Technical report, ETRO Department-IRIS research group, Vrije Universiteit Brussel, September 1997.

[179] Luc van Kempen. *Ground Penetrating Radar for Anti-Personnel Landmine Detection.* Doctor in engineering sciences, Vrije Universitieit Brussel, Faculty of Engineering Sciences, Dept. of Electronics and Informatics, September 2006.

[180] Denes Vigh and E. William Starr. 3D prestack plane-wave, full-waveform inversion. *Geophysics*, 73(5):VE135–VE144, 2008.

[181] J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74:WCC1–WCC26, November - December 2009.

[182] C R Vogel. Non-convergence of the L-curve regularization parameter selection method. *Inverse Problems*, 12, March 1996.

[183] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, January 1996.

[184] Curtis R. Vogel. *Computional Methods for Inverse Problems.* SIAM, 2002.

[185] Vogelius, Michael S., Volkov, and Darko. Asymptotic formulas for perturbations in the electromagnetic fields due to the presence of inhomogeneities of small diameter. *ESAIM: Mathematical Modelling and Numerical Analysis*, 34(4):723–748, 3 2000.

[186] John L. Volakis, Arindam Chatterjee, and Leo C. Kempel. *Finite Element Method for Electromagnetics.* IEEE Press, 1998.

[187] John L. Volakis and Kubilay Sertel. *Integral Equation Methods for Electromagnetics.* SciTech Publishing, Inc., 2012.

[188] John L. Volakis, Kubilay Sertel, and Brian C. Usner. *Frequency Domain Hybrid Finite Element Methods in Electromagnetics.* Morgan & Claypool, 2006.

[189] C. Vuik, Y. A. Erlangga, and C. W. Oosterlee. Shifted Laplace preconditioner for the helmholtz equations. Technical report, Dept. Appl. Math. Anal., Deflt Univ. Tech., 2003.

[190] Maue A. W. On the formulation of a general scattering problem by means of an integral equation. *Z. Phys.*, 126:601, 1949.

[191] Michael Warner, Andrew Ratcliffe, Tenice Nangoo, Joanna Morgan, Adrian Umpleby, Nikhil Shah, Vetle Vinje, Ivan Štekl, Lluís Guasch, Caroline Win, Graham Conroy, and Alexandre Bertrand. Anisotropic 3D full-waveform inversion. *Geophysics*, 78(2):R58–R80, March-April 2013.

[192] F Watson and WRB Lionheart. SVD analysis of GPR full-wave inversion. In *Ground Penetrating Radar (GPR), 2014 5th International Conference on*, pages 484–490, 2014.

[193] Francis Watson. A full-wave inversion strategy for surface GPR measurements to determine 3D targets in an area of interest. Working title.

[194] Francis Watson. GPR FWI Matlab code. University of Manchester, Manchester, United Kingdom. 10.5281/zenodo.31437.

[195] Francis Watson. A novel preconditioner for the GPR full-wave inversion problem in 3D.

[196] Francis Watson. Data collection report. IAA 074 Feasibility of Full-Wave Inversion, May 2015.

[197] Hassler Whitney. *Geometric Integration Theory.* Princeton University Press, 1957.

[198] David Wong and Lawrence Carin. Analysis and processing of ultra wide-band SAR imagery for buried landmine detection. *IEEE Transactions on Antennas and Propagation*, 46:1747–1748, November 1998.

[199] Shi-Liang Wu, Ting-Zhu Huang, Liang Li, and Liang-Lin Xiong. Positive stable preconditioners for symmetric indefinite linear systems arising from Helmholtz equations. *Elsevier*, 2009.

[200] Pasi Ylä-Oijala and Matti Taskinen. Calculation of CFIE impedance matrix elements with RWG and *IEEE Transactions on Antennas and Propagation*, 51(8), August 2003.

[201] Zhaofa Zeng, Jing Li, Ling Huang, and Xuan Feng. Improving target detection accuracy based on multipolarization MIMO GPR. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):15–24, January 2015.

[202] Yu Zhu and Anreas Cangellaris. *Multigrid Finite Element Methods for Electromagnetic Field Modeling*. John Wiley & Sons, 2006.

[203] Massoud Zolgharni, H. Griffiths, and D.S. Holder. Imaging haemorrhagic cerebral stroke by frequency-difference magnetic induction tomography: numerical modelling. In Jos Vander Sloten, Pascal Verdonck, Marc Nyssen, and Jens Haueisen, editors, *4th European Conference of the International Federation for Medical and Biological Engineering*, volume 22 of *IFMBE Proceedings*, pages 2464–2467. Springer Berlin Heidelberg, 2009.