

# **Inversion of Airborne Electromagnetic data in 2.5D**

by

Wing Wa Yu

B.Sc., University of Iceland, 2006

M.Sc., University of British Columbia, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF

**Master of Science**

in

THE FACULTY OF GRADUATE STUDIES

(Mathematics)

The University Of British Columbia

(Vancouver)

October 2012

© Wing Wa Yu, 2012

# Abstract

In this work, we implement an inversion algorithm for airborne electromagnetic (AEM) data in the frequency domain by using 2D conductivity models. First, we discretize the 2.5D Maxwell's equations on a staggered grid and test the numerical accuracy of the forward solution. The inverse problem is then solved by regularized minimization approach using the limited memory BFGS variant of the quasi-Newton method. Next, EM responses from a synthetic 2D conductivity model are inverted to validate the algorithm. Finally, we use the algorithm on an AEM field dataset from a RESOLVE survey and compare the inversion results to those obtained from a well-established 1D implementation.

# Table of Contents

<b>Abstract</b>	<b>ii</b>
<b>Table of Contents</b>	<b>iii</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Figures</b>	<b>vi</b>
<b>Acknowledgments</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Inverse problems	1
1.2 AEM system	2
1.3 Motivation and Goal	5
1.4 Outline of the thesis	8
<b>2 Forward Modeling</b>	<b>9</b>
2.1 Maxwell's equations in 2.5D	9
2.2 Fourier transform of the source term	13
2.3 Optimization of $k_y$ and $g$	14
2.4 2D discretization of Maxwell's equations	17
2.4.1 Discretization of the physical quantities	17
2.4.2 Discretization of the differential operators	19
2.4.3 Discretization of the material averaging	25

2.4.4	Matrix representation of Maxwell's equations . . . . .	26
2.5	Solving the adjoint problem . . . . .	28
2.6	Solving for multiple RHS . . . . .	29
<b>3</b>	<b>Inversion framework . . . . .</b>	<b>32</b>
3.1	Objective function . . . . .	33
3.2	Sensitivity matrix . . . . .	34
3.3	Method of optimization . . . . .	36
3.4	Implementation of the optimization . . . . .	38
3.4.1	Data structures . . . . .	38
3.4.2	Constrained optimization . . . . .	38
3.4.3	Regularization parameter selection . . . . .	40
3.4.4	Stopping criteria . . . . .	44
<b>4</b>	<b>Results . . . . .</b>	<b>45</b>
4.1	Testing the forward solution . . . . .	45
4.2	Inversion of synthetic data . . . . .	48
4.3	Inversion of field data . . . . .	50
<b>5</b>	<b>Conclusion and future work . . . . .</b>	<b>59</b>
	<b>Bibliography . . . . .</b>	<b>61</b>

# List of Tables

Table 2.1	Overview of the discretization of the physical quantities and the operators used in the Maxwell's equations . . . . .	20
Table 2.2	Mapping of the four types of gradient operator . . . . .	20
Table 2.3	Substitution of operators for the setup of the adjoint problem	30
Table 4.1	Solution error of the 2.5D forward modeling for different cell sizes. . . . .	47

# List of Figures

Figure 1.1	An AEM system and the magnetic response from a buried conductor. The transmitter (Tx) and receiver (Rx) of the system are housed in a bird which is towed behind and below a helicopter. In the presence of a conductive body, the primary field from the transmitter induces eddy currents in the conductor and a secondary field is produced as a result. The induced field measured at the receiver loop is then used to determine the conductivity of the body. . . .	4
Figure 1.2	A cross-sectional view of the layered earth model. The model is made of layers of earth, each with a thickness of $h_i$ and a conductivity of $\sigma_i$ . . . . .	6
Figure 1.3	A cross-sectional view of the 2D earth model. The model is a function of both position along the flight direction and depth. Each conductor has a conductivity of $\sigma_i$ and its extent in the strike direction is infinite. . . . .	7
Figure 2.1	Discretization of the electromagnetic field and the conductivity model on a 2D staggered grid. Black dots denote the location of the nodes, boxes are the $x$ -edges (defined as edges whose normal points in the $x$ -direction), diamonds are the $z$ -edges and crosses represent the cell-centers. . . .	21

Figure 2.2	An example of a mesh generated for AEM inversion. The color shows the initial guess of the log of the conductivity model and the crosses represent the source positions. . . .	31
Figure 3.1	Tikhonov curve. For large $\beta$ , the model norm $\phi_m$ dominates the objective and the data misfit is underrepresented. The opposite happens when $\beta$ is small. When the noise level is known, the target misfit, $\phi_d(\beta^*, m^*)$ , can be used to determine the regularization parameter. . . . .	42
Figure 3.2	The L-curve. It can be used to select the regularization parameter $\beta$ when the noise level is unknown. In this strategy, $\beta$ is chosen at the point of maximum curvature of the L-curve. . . . .	43
Figure 4.1	Schematic of the mesh and the conductivity model used to test the 2D forward modeling. . . . .	46
Figure 4.2	Comparison of 2D response with 1D analytical solution. a) Computed responses from the synthetic model shown in Fig. 4.1. Circles are the 1D analytical responses and the crosses are the 2D modeled responses. b) Relative error of the 2D responses. . . . .	47
Figure 4.3	Synthetic model used in testing the 2D inversion algorithm. The model is 2D and it consists of a conductive prism and an overburden that is conductive on one end and resistive on the other. Responses were recorded 10 m away from the sources and 40 m above the ground. Locations of the sources are indicated by the red crosses. . . .	50

Figure 4.4	a) Log of the recovered 2D conductivity model for the synthetic data generated by the model shown in 4.3. b) Left: misfit of predicted data normalized by noise standard deviation. Middle and right: The response curves are offset by 1000 ppm for clarity. Crosses represent the synthetic inphase and quadrature responses and solid curves are the predicted data. . . . .	51
Figure 4.5	Satellite image of the Red Dog mine area in Alaska. Red line: outline of the survey area. White line: outline of the waste area. Blue line: outline of the tailing pond. Black lines: ore deposits. Green lines: survey lines inverted. Image from [21]. . . . .	53
Figure 4.6	Predicted data misfit for the inversions on the lines shown in Fig. 4.5: a)-b) Line 10300, c)-d) Line 20590. The curves are offset by 1000 ppm for clarity. Solid lines: predicted response. Crosses: observed data. Observed data are from [21]. . . . .	54
Figure 4.7	Normalized bird altitude and a measured response curve. The two observables are strongly correlated. . . . .	55
Figure 4.8	2D (a),c)) and 1D (b),d)) inverted log conductivity models for the two survey lines highlighted in Fig. 4.5. a)-b) Line 10300, c)-d) Line 20590. . . . .	56



# Acknowledgments

From the beginning to the end of this Master's program, I have received a lot of supports from many good people and I am very thankful to each one of them.

My thesis advisor, Eldad Haber, has always been extremely helpful whenever I encounter difficulties with my research. He is always very supportive of the directions his students would like to take and he always leaves them with plenty of freedom to create and develop their own ideas. For that I am very grateful.

I would like to express my gratitude to Doug Oldenburg, Philip Loewen for being in my thesis committee and for providing invaluable helps in making a successful completion of my degree possible.

Many thanks also go to Joel Jansen and Boris Lum of Teck Resources for sharing the Red Dog field dataset as well as giving me the chance to gain industrial experience through an internship with Teck.

Last but not least, I am very thankful to all my fellow students and colleagues in UBC and in Teck for broadening my understanding in the field of applied mathematics and computer science, as well as in geophysics and geology.

# Chapter 1

## Introduction

### 1.1 Inverse problems

The need to reconstruct a property of a system by using observations made on the system arises in many fields of science and engineering: for example geophysics, medical imaging, machine learning, remote sensing, particle physics, astronomy and many more.

In many cases, despite the relative ease to conduct experiments on a system we are interested in, acquiring enough information such that a property can perfectly be predicted is usually not possible. This problem is most apparent in physical systems, where usually only a fraction of the total number of degrees of freedom is easily accessible, for example in a 3D system where measurements can only be made on the boundaries. This lack of measurable parameters makes the system under-determined: the number of variables in the property that we wish to reconstruct is greater than the number of data points available.

Parameter estimation for an under-determined system is essentially an ill-posed problem, meaning that its solution space is infinitely large and a unique solution does not exist. This type of problem is solved by reducing the solutions to only those that are both good predictors of the observations and

reasonable for the system according to some criteria. Information previously learned about the system, or *a priori* information, is used to devise the criteria to make educated guesses on the solutions.

Studies of these problems, known as inverse problems, have given us a general framework to convert any type of observations on a system to information about any type of properties of the system. But before any data can be inverted, the forward solution has to be known. That is, given a property model, we need to be able to predict the outcome of observations on the model. Once the forward problem is solved, the inverse framework can subsequently be employed to reconstruct the model. The goal of an inverse algorithm is to rebuild the system property by choosing a model that both best-fits to the observed data and satisfies the constraints formulated by *a priori* information on the system. Solving inverse problems therefore involves both solving systems of equations as well as optimizing objective functions with constraints.

## 1.2 AEM system

In recent geophysical exploration projects, airborne electromagnetic (AEM) surveys have become the method of choice for many mining companies to determine the conductivity model under the area of investigation. The popularity of AEM systems comes from their ability to cover a large area in a relatively short time while requiring only a small number of human operators. They are also ideal when the survey area has complicated terrain or is not accessible by ground transportation.

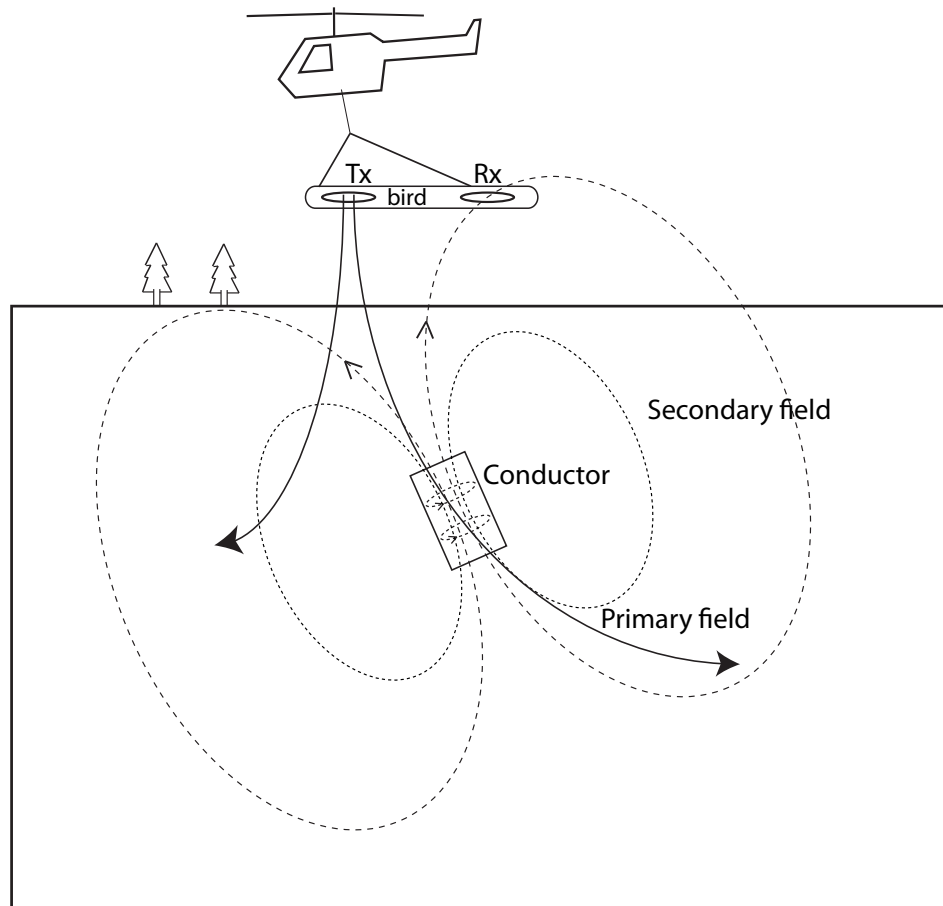
In most AEM systems, horizontal coplanar loops (HCP) are used as transmitters (Tx) and receivers (Rx) to detect conductivity contrasts in the earth. In frequency-domain AEM systems, the loops are typically separated by 10 meters and they are housed in a bird that is towed behind and beneath an aircraft at a height of 30-70 meters above the earth's surface. To determine the conductivity contrast, a time-varying current is passed through the trans-

mitter loop to produce a magnetic dipole field. If a conductive body is buried in the ground, the primary field will induce eddy currents in the conductor and a secondary magnetic field will be generated as a result. This secondary field changes the magnetic flux through the receiver loop and its magnitude can be quantified by measuring the emf induced in the receiver (see Fig 1.1). In this way, the AEM system probes the conductivity distribution next to the loops by measuring the earth's response to changing magnetic fields.

In typical frequency-domain AEM systems, each conductivity depth sounding (mapping of conductivity as a function of depth) consists of magnetic field measurements at five to six frequencies. The frequencies usually span over three orders of magnitude and are spread logarithmically from hundreds of Hz to hundreds of kHz. In this frequency range, the diffusion length of the magnetic field, which is given by the skin depth

$$\delta = \sqrt{\frac{2}{\sigma\omega\mu_0}}, \quad (1.1)$$

is on the order of 1 to 100 meters. Thus, current can be induced from a few meters up to a few hundreds of meters deep into the ground, and the depth of penetration drops as the inverse square root of the frequency. The relation between frequency and penetration depth allows AEM systems to profile the conductivity distribution as a function of depth. To map the horizontal direction, soundings are carried out at regularly spaced intervals along a flight line. Typically, data are sampled every few tenths of a second which is equivalent to a sounding every few meters along the line. To create a pseudo-3D map of the conductivity of the survey area, multiple lines are flown over the area. The line separation can range from tens to hundreds of meters, depending on the size of the geological features under study.



**Figure 1.1:** An AEM system and the magnetic response from a buried conductor. The transmitter (Tx) and receiver (Rx) of the system are housed in a bird which is towed behind and below a helicopter. In the presence of a conductive body, the primary field from the transmitter induces eddy currents in the conductor and a secondary field is produced as a result. The induced field measured at the receiver loop is then used to determine the conductivity of the body.

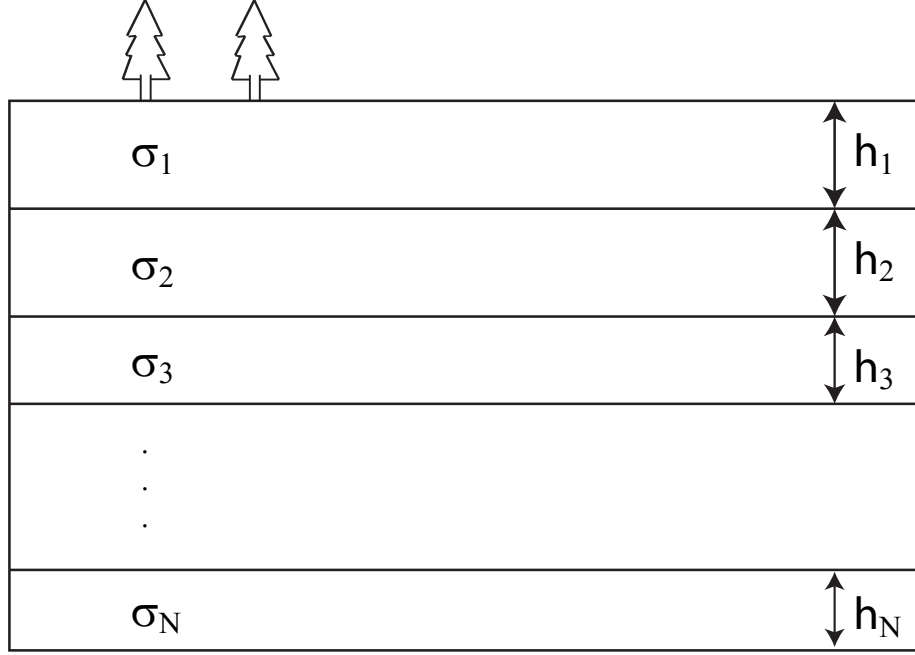
### 1.3 Motivation and Goal

Inverse algorithms that are both reliable and fast enough to process an entire AEM survey dataset are still challenges to many researchers at the moment. The constant movement of the source over a large area of investigation means a vast number of sources has to be solved in each forward modeling. It is not unusual for an airborne survey to have more than hundreds of thousands of different source locations. Compared to ground survey methods, this is about two orders of magnitude more sources.

The first generation of inverse algorithms was built on a 1D conductivity model that assumes the earth is made of layers of conductors stacked on top of each other (see Fig. 1.2). An analytical solution to the 1D problem can be obtained for HCP sources [18] and numerical implementations [4, 23] of the solution have been extensively studied and they are well-established. This simple layered earth model has been used in many surveys with good results and on today's processors the 1D algorithm is extremely fast to run. However, the main shortcoming of the model is its simplicity. There are cases where the 1D implementation has troubles resolving correctly 2D or 3D geological structures that exhibit rapid lateral change in the immediate region under the probes (e.g., [5, 14, 25, 27]). To obtain a more accurate description of the earth, inversion techniques based on the full 3D conductivity model are now widely being studied and tested.

At the moment, 3D inversion algorithms based on the finite volume (FV) method [9] and the integral equation approach [3] are available. Although the inversion of an entire AEM dataset is now possible with these algorithms, they still face numerous problems. A full inversion with the FV method can take hours to complete, and the amount of computational resources required is still huge and expensive. Although the integral equation technique is faster, it may not correctly resolve models with high conductivity contrasts.

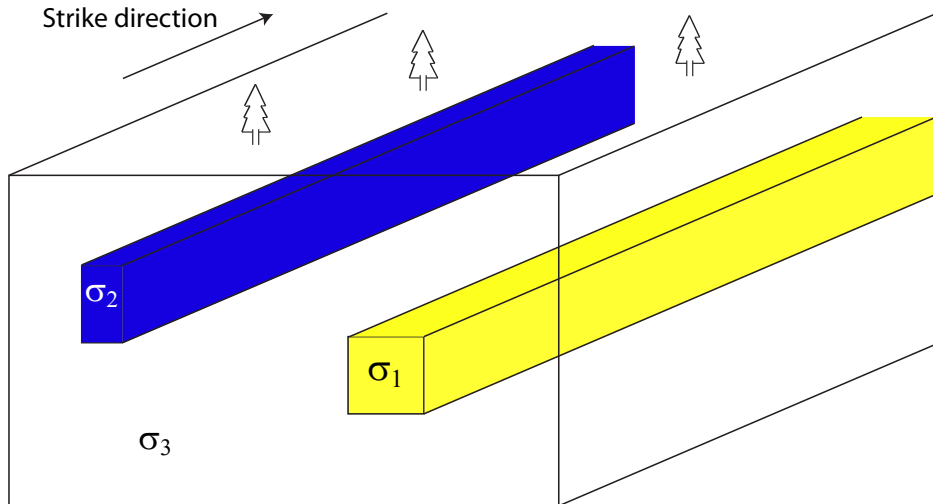
To try to resolve some of the problems that existing 1D and 3D inversions have, we tested and developed an inversion algorithm using a 2D conductivity



**Figure 1.2:** A cross-sectional view of the layered earth model. The model is made of layers of earth, each with a thickness of  $h_i$  and a conductivity of  $\sigma_i$ .

earth model(see Fig. 1.3). In AEM surveys, the spatial separation between survey lines is often large, and the survey is done perpendicular to geological strikes. As a result, each magnetic field measurement does not extend noticeably into the adjacent lines and so each flight line can effectively be inverted independently without too much loss in resolution. This motivates the use of 2D earth models to invert AEM datasets. Compared to 1D algorithms, 2D inversions have the advantage of enabling more precise interpretation of conductive anomalies; and compared to 3D strategies, they require much less computing resources and are relatively straightforward to parallelize for multi-core CPUs.

Over the past decades, the well-known 2.5D formulation of Maxwell's equations has been used to design numerous 2D EM algorithms ([6, 20,



**Figure 1.3:** A cross-sectional view of the 2D earth model. The model is a function of both position along the flight direction and depth. Each conductor has a conductivity of  $\sigma_i$  and its extent in the strike direction is infinite.

22]). These algorithms, which have been applied to various airborne and controlled-source EM problems (e.g., [1, 13, 15, 24]), are either based on finite element (FEM) or finite difference (FD) approaches. But none used the finite volume method, which we have preferred over FEM or FD, due to its simplicity in generating the matrix for the linear system, as well as its ability to handle high conductivity contrast.

In this work, we used the finite volume approach to implement an algorithm for inversion of frequency-domain AEM data. The algorithm, which is developed in Matlab, was first tested on synthetic data generated by a 2D model and later on a 3D AEM field dataset from a Fugro RESOLVE survey. Our 2D inversions on the dataset showed strong agreement with models recovered by a well-established 1D algorithm.



## 1.4 Outline of the thesis

The remainder of this thesis is divided into four chapters. In chapter two, we formulate and solve the forward problem by discretizing the 2.5D Maxwell's equations in the frequency domain on a 2D orthogonal mesh. In chapter three, we present the inversion framework by discussing and constructing the objective function and the sensitivity matrix, followed by a look into the technique used in optimizing the objective function. Finally, in chapter four, we show the accuracy of the forward model and then present the results of our 2D inversions on both synthetic and field data. A conclusion and discussion of future work can be found in the last chapter.

# Chapter 2

## Forward Modeling

In this chapter, we start by presenting the well-studied 2.5D formulation of the Maxwell's equations. By assuming a 2D conductivity model that is invariant in the strike direction, 3D electromagnetic fields are rewritten as the sum of 2D fields in the wavenumber domain by applying Fourier transformations w.r.t. the strike direction. Once the 2.5D Maxwell's equations are solved for an optimized set of wavenumbers, the inverse transformation is used to obtain the 3D solution.

Subsequently, we will present a finite volume discretization of the electromagnetic field on a 2D staggered grid, followed by a discretization for the **CURL** operator and the material averaging. Finally, the discrete system for the 2.5D Maxwell's equations is set up to compute the electromagnetic field.

### 2.1 Maxwell's equations in 2.5D

Conventionally, measurement data in AEM surveys are presented as complex numbers: the real component of the magnetic response, which is called in-phase, and the imaginary part, called quadrature. The response is the ratio between the secondary magnetic field and the primary field produced by the transmitter in free space. For horizontal coplanar loops, the response is given

by

$$Resp = \frac{\vec{H}_z(r_{tr}) - \vec{H}_{p,z}(r_{tr})}{|\vec{H}_{p,z}(r_{tr})|}, \quad \vec{H}_{p,z}(r) = -\frac{m}{4\pi r^3} \quad (2.1)$$

where  $r_{tr}$  is the transmitter to receiver separation,  $m$  is the magnetic moment of the transmitter,  $\vec{H}_{p,z}$  is the  $z$ -component of the primary field and  $\vec{H}_z(r_{tr})$  is the total field measured at the receiver.

To model the magnetic response due to harmonically changing electromagnetic fields,  $\vec{F}(\vec{x}, t) = \vec{F}(\vec{x})e^{-i\omega t}$ , we solve Maxwell's equations in the frequency domain:

$$\nabla \times \vec{E} - i\omega\mu_0\vec{H} = \vec{M}_s \quad (2.2a)$$

$$\nabla \times \vec{H} - \gamma\vec{E} = \vec{J}_s \quad \text{on } \Omega \quad (2.2b)$$

$$\hat{n} \times \vec{H} = 0 \quad \text{on } \partial\Omega \quad (2.2c)$$

where we have imposed a vanishing magnetic field on the boundary of the spatial domain  $\Omega$ , which consists of both the earth and the air above it. In these equations,  $\vec{E}$  and  $\vec{H}$  are the electric and magnetic fields,  $\omega$  is the angular frequency of the fields,  $\mu_0$  the magnetic permeability of free space and  $\vec{J}_s$  and  $\vec{M}_s$  denote the external electric and magnetic sources. The conductivity  $\gamma = \sigma - i\omega\epsilon$  is complex valued, but for most applications in geophysics, where the quasi-static condition  $\omega\epsilon \ll \sigma$  usually applies [23], it can be simplified to the real-valued conductivity  $\sigma$ . From now on, we will use  $\sigma$  for the conductivity.

To solve Maxwell's equations, we decompose the total fields  $\vec{E}, \vec{H}$  into primary and secondary parts:

$$\vec{E} = \vec{E}_0 + \vec{E}_1, \quad \vec{H} = \vec{H}_0 + \vec{H}_1, \quad (2.3)$$

so

$$(\nabla \times \vec{E}_0 - i\omega\mu_0\vec{H}_0) + (\nabla \times \vec{E}_1 - i\omega\mu_0\vec{H}_1) = 0 \quad (2.4a)$$

$$(\nabla \times \vec{H}_0 - \sigma\vec{E}_0) + (\nabla \times \vec{H}_1 - \sigma\vec{E}_1) = \vec{J}_s \quad \text{on } \Omega \quad (2.4b)$$

where we have set  $\vec{M}_s = 0$ , since only electric sources are used in our system. This decomposition separates Maxwell's equations into a primary problem,

$$\nabla \times \vec{E}_0 - i\omega\mu_0\vec{H}_0 = 0 \quad (2.5a)$$

$$\nabla \times \vec{H}_0 = \vec{J}_s \quad (2.5b)$$

and a secondary problem,

$$\nabla \times \vec{E}_1 - i\omega\mu_0\vec{H}_1 = 0 \quad (2.6a)$$

$$\nabla \times \vec{H}_1 - \sigma\vec{E}_1 = \sigma\vec{E}_0 \quad (2.6b)$$

Setting  $\vec{J}_s$  to the current in a HCP, the primary problem describes the field due to the current loop in free space. With the source now replaced by the induced current density evaluated on the entire conductivity model  $\sigma$ , the secondary problem is the Maxwell's equations for the induced magnetic field. This secondary form enables the magnetic response to be computed accurately which is otherwise very hard as the discretization error in computing the total field dominates the induced field strength. From now on, we will work with the secondary form and drop the secondary notation:  $\vec{E}_1, \vec{H}_1 \rightarrow \vec{E}, \vec{H}$ .

In this work, we assume an isotropic 2D conductivity model that has infinite extent in the strike direction  $y$ :  $\sigma = \sigma(x, z)$ . Despite the 2D conductivity, each field component remains 3-dimensional as the source field is a three dimensional function. The advantage of using a 2D model is that now the 3D Maxwell's equations can be reduced to 2D equations by Fourier transforming Eqn. (2.6) with respect to  $y$ . Since our system is invariant under translation in the  $y$ -direction, all field components are either symmetric or antisymmetric in  $y$ . Therefore, we can either use the sine or cosine integral

for the transformation:

$$\tilde{F}(x, k_y, z) = \begin{cases} -i \int_0^\infty \vec{F}(x, y, z) \sin(k_y y) dy & \text{if } \vec{F} \text{ is anti-symmetric in } y \\ \int_0^\infty \vec{F}(x, y, z) \cos(k_y y) dy & \text{if } \vec{F} \text{ is symmetric in } y \end{cases} \quad (2.7)$$

where  $\tilde{F}$  denotes  $\vec{F}$  in the Fourier space which is parameterized by the wavenumber  $k_y$ . For antisymmetric functions, the sine transform is used, while the cosine transform is applied to symmetric functions.

In the wavenumber domain, we have the following 2D Maxwell's equations for each  $k_y$ :

$$\tilde{\nabla}_{k_y} \times \tilde{E} - i\omega\mu_0 \tilde{H} = 0 \quad (2.8a)$$

$$\tilde{\nabla}_{k_y} \times \tilde{H} - \sigma \tilde{E} = \sigma \tilde{E}_0 \quad (2.8b)$$

where  $\tilde{\nabla}_{k_y} \times$  is the 2.5D curl operator in the wavenumber domain and it acts on the field  $\tilde{F}$  as follows:

$$\tilde{\nabla}_{k_y} \times \tilde{F} = \begin{bmatrix} -ik_y \tilde{F}_z - \partial_z \tilde{F}_y \\ \partial_z \tilde{F}_x - \partial_x \tilde{F}_z \\ \partial_x \tilde{F}_y + ik_y \tilde{F}_x \end{bmatrix} \quad (2.9)$$

where  $\tilde{F} = (\tilde{F}_x, \tilde{F}_y, \tilde{F}_z)$ . The Fourier transform of the source field,  $\tilde{E}_0$ , can be approximated by assuming a vertical dipole source and the formulation is shown in the next section. Once a solution to (2.8) is obtained for each  $k_y$ , the 3D solution can be retrieved with the inverse Fourier transform:

$$\vec{F}(x, y, z) = \begin{cases} \frac{2i}{\pi} \int_0^\infty \tilde{F}(x, k_y, z) \sin(k_y y) dk_y & \text{if } \tilde{F} \text{ is anti-symmetric in } k_y \\ \frac{2}{\pi} \int_0^\infty \tilde{F}(x, k_y, z) \cos(k_y y) dk_y & \text{if } \tilde{F} \text{ is symmetric in } k_y \end{cases} \quad (2.10)$$

To evaluate these integrals, we approximate them using the discrete Fourier

transforms:

$$\vec{F}(x, y, z) = \begin{cases} \frac{2i}{\pi} \sum_{j=1}^N \tilde{F}(x, k_{y,j}, z) \sin(k_{y,j}y) g_j & \text{if } \tilde{F} \text{ is anti-symmetric in } k_y \\ \frac{2}{\pi} \sum_{j=1}^N \tilde{F}(x, k_{y,j}, z) \cos(k_{y,j}y) g_j & \text{if } \tilde{F} \text{ is symmetric in } k_y \end{cases} \quad (2.11)$$

where  $g_j$  is a weight associated with the  $j$ -th wavenumber  $k_{y,j}$ . Later in the chapter, we will show how the choice of the wavenumbers and weights can be optimized for our problem in order to minimize the computational cost.

Now, we have seen how Maxwell's equations in 2.5D are formulated. Not only is the computing time reduced because of the smaller size of the problem, the 2.5D implementation also makes the forward calculation easily parallelizable as each 2D equation in (2.8) can be solved independently. Once the 2D conductivity cross-section directly underneath each survey line is recovered, interpolation between adjacent lines can be used to create a pseudo-3D model of the survey area.

## 2.2 Fourier transform of the source term

The primary fields  $\vec{E}_0, \vec{H}_0$  are the free-space electromagnetic fields due to the current density in the source loop,  $\vec{J}_s$ :

$$\nabla \times \vec{E}_0 - i\omega\mu_0\vec{H}_0 = 0 \quad (2.12a)$$

$$\nabla \times \vec{H}_0 = \vec{J}_s \quad (2.12b)$$

Instead of solving (2.12) directly, the primary field  $\vec{E}_0$  can be obtained by modeling the transmitter loop as a vertical magnetic dipole source. For a source centered at  $(x_s, 0, h_s)$  and oscillating at frequency  $\omega$ , the electric field

is given by the vector potential of the dipole,  $\vec{A}$  [11]:

$$\vec{E}_0(x, y, z) = -\partial_t \vec{A} = i\omega \vec{A} = \frac{i\omega m \mu_0}{4\pi r^3} \begin{bmatrix} -y \\ x - x_s \\ 0 \end{bmatrix} \quad (2.13)$$

where  $r = \sqrt{(x - x_s)^2 + y^2 + (z - h_s)^2}$ .

For every source on the same survey line, the Fourier transform of  $\vec{E}_0$  is given by

$$\tilde{E}_0(x, k_y, z) = \frac{i\omega \mu_0 m}{4\pi} \begin{bmatrix} ik_y K_0(|k_y|s) \\ |k_y| K_1(|k_y|s)/s \\ 0 \end{bmatrix} \quad (2.14)$$

where  $s = \sqrt{(x - x_s)^2 + (z - h_s)^2}$  and  $K_n$  is the  $n$ -th order modified Bessel function of the second kind. It is worth mentioning that with this formulation, vertical loops can easily be modeled as well by simply switching the  $x$  and  $z$  coordinates.

## 2.3 Optimization of $k_y$ and $g$

Given a fixed number of wavenumbers to be used to solve (2.8), we seek the optimum  $k_{y,j}$ 's and the associated weights such that the error of approximating the Fourier transform (Eqn. (2.11)) is minimized. In this work, this is done by finding the  $k_{y,j}$ 's and  $g_j$ 's that minimize the error in approximating a vertical dipole field on the surface of a homogeneous half-space. The same optimized values are then used to compute responses due to inhomogeneous half-spaces. Since the receivers in most AEM systems are located in-line with the sources (we will use  $y_{Rx} = y_{Tx} = 0$ ), we only need to be concerned with minimizing the error for the locations on the survey line. For this, we will follow the optimization technique proposed in [17, 26].

We start by Fourier transforming the field due to a homogeneous half-space. For the range of conductivities and frequencies typically found in

AEM surveys, the low-frequency approximation applies, i.e.,  $|kr| \ll 1$ , where  $k = \sqrt{-i\sigma\omega\mu_0}$  and  $r$  is the distance from the center of the source. The quasi-static field of a vertical dipole source on the surface of a half-space with conductivity  $\sigma$  in this regime is given by [19]:

$$\vec{H}_x^l(x, y, 0) = \vec{H}_y^l(x, y, 0) = 0, \quad \vec{H}_z^l(x, y, 0) = \frac{-m}{4\pi r^3} \left(1 - \frac{k^2 r^2}{4}\right) \quad (2.15)$$

where  $r = \sqrt{x^2 + y^2}$  for a source centered at the origin. Substituting (2.15) into the Fourier transform for symmetric functions (2.7) gives

$$\tilde{H}_z^l(x, k_y, 0) = \frac{-m}{4\pi} \left( \frac{|k_y|}{x} K_1(|k_y|x) - \frac{k^2}{4} K_0(|k_y|x) \right) \quad (2.16)$$

Our objective is to minimize the relative error in approximating (2.15) with the discrete Fourier cosine transform (2.11):

$$\vec{H}_z^l(x, 0, 0) \approx \frac{2}{\pi} \sum_j \tilde{H}_z^l(x, k_{y,j}, 0) g_j \quad (2.17)$$

for a range of  $x$ . Given a location  $x_i$ , the relative error  $\epsilon_i$  is:

$$\epsilon_i = \frac{2}{\pi \vec{H}_z^l(x_i, 0, 0)} \sum_j \tilde{H}_z^l(x_i, k_{y,j}, 0) g_j - 1 \quad (2.18)$$

and so our objective function can be written in the matrix form:

$$\Phi = \frac{1}{2} \|\epsilon\|_2^2 = \frac{1}{2} \|\mathbf{1} - \mathbf{M}\mathbf{g}\|_2^2 \quad (2.19)$$

where

$$\mathbf{M}_{ij} = \frac{2\tilde{H}_z^l(x_i, k_{y,j}, 0)}{\pi \vec{H}_z^l(x_i, 0, 0)}, \quad (2.20)$$

and  $\mathbf{g}$  is a vector whose entries are  $g_j$  and  $\mathbf{1}$  is a vector of ones.

Notice that the objective function (2.19) is a function of both  $\mathbf{k}_y$  and  $\mathbf{g}$ .



To reduce it to a function that depends only on  $\mathbf{k}$ , we optimize for  $\mathbf{g}$  using the normal equation of least squares,

$$\mathbf{g} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{1} \quad (2.21)$$

and substitute it back into (2.19) to obtain

$$\Phi = \frac{1}{2} \|\mathbf{1} - \mathbf{M}(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{1}\|_2^2 = \frac{1}{2} \|\mathbf{1} - \mathbf{v}(\mathbf{k}_y)\|_2^2 \quad (2.22)$$

Since the function  $\mathbf{v}(\mathbf{k}_y)$  is nonlinear in  $\mathbf{k}_y$ , we linearize it by expanding it in Taylor series around a point  $\mathbf{k}_y^{(0)}$

$$\mathbf{v}(\mathbf{k}_y) = \mathbf{v}(\mathbf{k}_y^{(0)}) + \frac{\partial \mathbf{v}}{\partial \mathbf{k}_y} \delta \mathbf{k}_y, \quad \delta \mathbf{k}_y = \mathbf{k}_y - \mathbf{k}_y^{(0)}. \quad (2.23)$$

Now the least squares solution for the objective function

$$\Phi = \frac{1}{2} \|\mathbf{1} - \mathbf{v}(\mathbf{k}_y^{(0)}) - \frac{\partial \mathbf{v}}{\partial \mathbf{k}_y} \delta \mathbf{k}_y\|_2^2 \quad (2.24)$$

gives the search direction  $\delta \mathbf{k}_y$ :

$$\delta \mathbf{k}_y = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\mathbf{1} - \mathbf{v}(\mathbf{k}_y^{(0)})) \quad (2.25)$$

where  $\mathbf{J} = \frac{\partial \mathbf{v}}{\partial \mathbf{k}_y}$ . At the  $i$ -th iteration, a new  $\mathbf{k}_y^{(i+1)}$  is thus given by

$$\mathbf{k}_y^{(i+1)} = \mathbf{k}_y^{(i)} + \alpha \delta \mathbf{k}_y^{(i)} \quad (2.26)$$

where  $\alpha$  is a line search parameter. This iterative procedure reduces  $\mathbf{v}(\mathbf{k}_y)$  at each iteration and is repeated until a certain error tolerance is reached. At this point, the corresponding  $\mathbf{k}_y$  is used to compute  $\mathbf{g}$  using (2.20) and (2.21).

It should be noted that since we do not have an analytical expression for

$\frac{\partial \mathbf{v}}{\partial \mathbf{k}_y}$ , it has to be approximated by numerical differentiation:

$$\mathbf{J}^{(i)} = \left( \frac{\partial \mathbf{v}}{\partial \mathbf{k}_y} \right)^{(i)} = \begin{bmatrix} \frac{\Delta \mathbf{v}}{\Delta \mathbf{k}_{y,1}^{(i)}} & \cdots & \frac{\Delta \mathbf{v}}{\Delta \mathbf{k}_{y,N}^{(i)}} \end{bmatrix}, \quad \mathbf{v}(\mathbf{k}_y^{(i)} + \Delta \mathbf{k}_y^{(i)}) - \mathbf{v}(\mathbf{k}_y^{(i)}) \quad (2.27)$$

where we have set  $\Delta \mathbf{k}_y^{(i)} = 0.05 \mathbf{k}_y^{(i)}$ .

## 2.4 2D discretization of Maxwell's equations

In this section, we discretize the 2.5D Maxwell's equations on a 2D staggered grid using a finite volume approach. The linear system to be solved is set up by discretizing the field quantities on either the edges, nodes or cell-centers of the staggered grid and then forming the discrete version of the curl operator and the material averaging for these grid variables. The discretization of the gradient, which appears in the regularization of the inverse problem, will also be presented here.

### 2.4.1 Discretization of the physical quantities

To discretize the physical quantities in Maxwell's equations, we consider an  $m \times n$  staggered grid. Each cell is labeled by the indices  $i$  and  $j$  and has a size of  $h_{x,i}$  by  $h_{z,j}$ . To start, we put the electric fields  $\tilde{E}_x$  and  $\tilde{E}_z$  on the edges of each cell, as shown in Fig. (2.1). This configuration places constraints on the possible locations for the rest of the field quantities. To see this, we expand (2.8) to obtain the system of differential equations for the field components:

$$-ik_y\tilde{E}_z - \partial_z\tilde{E}_y - i\omega\mu_0\tilde{H}_x = 0 \quad (2.28a)$$

$$\partial_z\tilde{E}_x - \partial_x\tilde{E}_z - i\omega\mu_0\tilde{H}_y = 0 \quad (2.28b)$$

$$\partial_x\tilde{E}_y + ik_y\tilde{E}_x - i\omega\mu_0\tilde{H}_z = 0 \quad (2.28c)$$

$$-ik_y\tilde{H}_z - \partial_z\tilde{H}_y - \sigma\tilde{E}_x = \sigma\tilde{E}_{0,x} \quad (2.28d)$$

$$\partial_z\tilde{H}_x - \partial_x\tilde{H}_z - \sigma\tilde{E}_y = \sigma\tilde{E}_{0,y} \quad (2.28e)$$

$$\partial_x\tilde{H}_y + ik_y\tilde{H}_x - \sigma\tilde{E}_z = \sigma\tilde{E}_{0,z} \quad (2.28f)$$

Firstly, Eqns. (2.28a) and (2.28c) imply that the fields  $\tilde{H}_x$  and  $\tilde{H}_z$  should be discretized at the same location as  $\tilde{E}_z$  and  $\tilde{E}_x$ , respectively. Secondly, Eqn. (2.28c) places the  $x$ -directional derivative of  $\tilde{E}_y$  on the  $z$ -edges (edges whose normal points in  $z$ -direction). And since the two-point formula is used for the derivative, this configuration puts  $\tilde{E}_y$  midway between the  $z$ -edges in the  $x$ -direction, namely on the nodes. Following a similar reasoning, but now using the constraint set by Eqn. (2.28f), we can see that  $\tilde{H}_y$  should be discretized at the cell-centers.

Unlike the secondary fields, which can only be approximated numerically, the source field  $\tilde{E}_0$  is continuous and its value is known everywhere. However, although  $\tilde{E}_0$  is an electric field, we will not discretize it like  $\tilde{E}$ . The reason is that  $\tilde{E}_0$  is singular at the center of a source,  $(x, z) = (x_s, h_s)$ . And since we would like to place the center on an  $z$ -edge, so that  $\tilde{H}_z$  can be computed at the receiver center without averaging, we should avoid discretizing  $\tilde{E}_0$  on  $z$ -edges. As a result, we will evaluate  $\tilde{E}_0$  at the cell-centers instead, and then use averaging to bring the current density,  $\sigma\tilde{E}_0$ , to the same mesh as the electric field.

In our modeling, the conductivity is assumed to be piecewise constant and therefore it should be discretized at the cell-centers. This discretization assumes that conductivity is constant throughout a cell but jumps may occur at the boundaries. This way, we can allow for highly varying and highly

discontinuous conductivity which is usually the case in geophysical settings. As required by the Maxwell's equations, the conductivity has to be evaluated at the other grid locations as well. For this, the material averaging matrix, which is discussed later in this chapter, is used to transfer the cell-centered conductivity to the desired locations.

An overview of the discretization of the physical quantities and the discrete operators that act on them is given in Table 2.1.

## 2.4.2 Discretization of the differential operators

To construct the discrete version of the 2.5D curl operator (2.9), **CURL**, and the gradient operator, **GRAD**, we discretize the difference operator  $\partial_x$  (discretization of  $\partial_z$  is then found by symmetry). We begin by realizing that for our 2D staggered grid, there are four types of difference operators. Each of them can be characterized by one of the mappings shown in Table (2.2). Next, we notice that the gradient operators that map edge→cell-center or node→edge use the same 1D nodal difference which maps nodes to cell-centers. And similarly, the edge→node and cell-center→edge mappings follow the structure of the 1D cell-center differences. Moreover, the boundary conditions (BC) given in (2.2c) require a Dirichlet type BC for the magnetic field quantities, while the electric fields follow BC of Neumann type. From this information, we can see that the 1D cell-center differences with Dirichlet BC and the 1D node differences with Neumann BC are needed to build **CURL** and **GRAD**. In this section, we will demonstrate how to discretize the 1D differences and then how to use them to construct the discrete version of the 2D operators  $\partial_x$  and  $\partial_z$ .

To build the matrices for the discrete operators, we will follow the column vector convention. Let  $U$  be a  $M \times N$  matrix whose entries,  $u_{ij}$ , are the values of the function  $u(x, z)$  associated with the cell at  $(x_i, z_j)$ ,  $1 \leq i \leq M, 1 \leq j \leq N$ . Instead of storing  $U$  as a 2D array, we vectorize it and store it in the

**Table 2.1:** Overview of the discretization of the physical quantities and the operators used in the Maxwell's equations

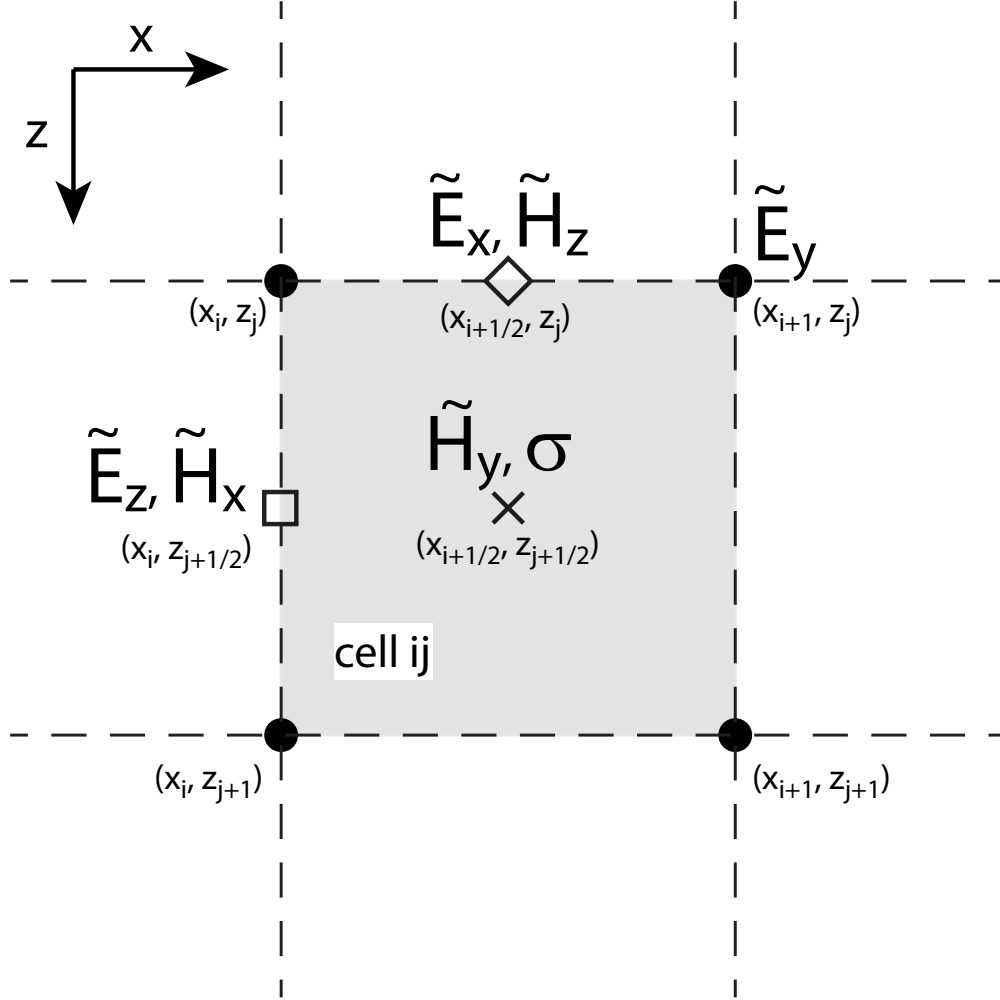
Quantity	Notation	Discretization
electric field	$\begin{bmatrix} \tilde{E}_x \\ \tilde{E}_y \\ \tilde{E}_z \end{bmatrix}$	$\begin{bmatrix} \text{z-edge} \\ \text{node} \\ \text{x-edge} \end{bmatrix}$
magnetic field	$\begin{bmatrix} \tilde{H}_x \\ \tilde{H}_y \\ \tilde{H}_z \end{bmatrix}$	$\begin{bmatrix} \text{x-edge} \\ \text{cell-center} \\ \text{z-edge} \end{bmatrix}$
source electric field	$\begin{bmatrix} \tilde{E}_{0,x} \\ \tilde{E}_{0,y} \\ \tilde{E}_{0,z} \end{bmatrix}$	$\begin{bmatrix} \text{cell-center} \\ \text{cell-center} \\ \text{cell-center} \end{bmatrix}$
conductivity	$\sigma$	cell-center

Operator	Notation	Mapping
curl of electric field	$\mathbf{CURL}_{\tilde{E}}$	$\begin{bmatrix} \text{z-edge} \\ \text{node} \\ \text{x-edge} \end{bmatrix} \rightarrow \begin{bmatrix} \text{x-edge} \\ \text{cell-center} \\ \text{z-edge} \end{bmatrix}$
curl of magnetic field	$\mathbf{CURL}_{\tilde{H}}$	$\begin{bmatrix} \text{x-edge} \\ \text{cell-center} \\ \text{z-edge} \end{bmatrix} \rightarrow \begin{bmatrix} \text{z-edge} \\ \text{node} \\ \text{x-edge} \end{bmatrix}$
x-edge average	$\mathbf{A}_{cc}^{xe}$	cell-center $\rightarrow$ x-edge
nodal average	$\mathbf{A}_{cc}^{nd}$	cell-center $\rightarrow$ node
z-edge average	$\mathbf{A}_{cc}^{ze}$	cell-center $\rightarrow$ z-edge

**Table 2.2:** Mapping of the four types of gradient operator

Acts on	Mapping
$\tilde{E}_x, \tilde{E}_z$	edge $\rightarrow$ cell-center
$\tilde{E}_y$	node $\rightarrow$ edge
$\tilde{H}_x, \tilde{H}_z$	edge $\rightarrow$ node
$\tilde{H}_y$	cell-center $\rightarrow$ edge



**Figure 2.1:** Discretization of the electromagnetic field and the conductivity model on a 2D staggered grid. Black dots denote the location of the nodes, boxes are the  $x$ -edges (defined as edges whose normal points in the  $x$ -direction), diamonds are the  $z$ -edges and crosses represent the cell-centers.

column format such that:

$$U(k) = u_{ij}, \quad k = (N-1) \times i + j \quad (2.29)$$

where  $U(k)$  is the  $k$ -th component of the column form of  $U$ . In this way, the field quantities are stored as:

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{\mathbf{F}}_x \\ \tilde{\mathbf{F}}_y \\ \tilde{\mathbf{F}}_z \end{bmatrix} \quad (2.30)$$

where each 2D field component is in the vector column form.

With this convention, the 2D operator that maps edges to cell-centers and nodes to edges can easily be constructed from the 1D nodal difference operator:

$$\mathbf{D}_{x,\text{node} \rightarrow \text{cell-center}}^{1D} = \begin{bmatrix} \frac{-1}{h_{x,1}} & \frac{1}{h_{x,1}} & & & \\ & \frac{-1}{h_{x,2}} & \frac{1}{h_{x,2}} & & \\ & & \ddots & \ddots & \\ & & & \frac{-1}{h_{x,m}} & \frac{1}{h_{x,m}} \end{bmatrix}, \quad (2.31)$$

and the Kronecker product,

$$\mathbf{A} \otimes \mathbf{B} := \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1,n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n}\mathbf{B} \\ a_{n,1}\mathbf{B} & \dots & a_{n,n-1}\mathbf{B} & a_{nn}\mathbf{B} \end{bmatrix} \quad (2.32)$$

in the following way [10]:

$$\mathbf{D}_{x,\text{x-edge} \rightarrow \text{cell-center}} = \mathbf{I}_n \otimes \mathbf{D}_{x,\text{node} \rightarrow \text{cell-center}}^{1D} \quad (2.33a)$$

$$\mathbf{D}_{x,\text{node} \rightarrow \text{z-edge}} = \mathbf{I}_{n+1} \otimes \mathbf{D}_{x,\text{node} \rightarrow \text{cell-center}}^{1D} \quad (2.33b)$$

where  $\mathbf{I}_n$  is the  $n \times n$  identity matrix and we have used the second-order two-point formula

$$\frac{d}{dx}u(x_{i+\frac{1}{2}}, z_j) = \frac{u(x_{i+1}, z_j) - u(x_i, z_j)}{h_{x,i}} + \mathcal{O}(h_{x,i}^2). \quad (2.34)$$

for the 1D derivative. It is easy to verify that the 2D difference matrices for the  $z$ -direction are:

$$\mathbf{D}_{z,z\text{-edge} \rightarrow \text{cell-center}} = \mathbf{D}_{z,\text{node} \rightarrow \text{cell-center}}^{1D} \otimes \mathbf{I}_m \quad (2.35a)$$

$$\mathbf{D}_{z,\text{node} \rightarrow x\text{-edge}} = \mathbf{D}_{z,\text{node} \rightarrow \text{cell-center}}^{1D} \otimes \mathbf{I}_{m+1} \quad (2.35b)$$

where  $\mathbf{D}_{z,\text{node} \rightarrow \text{cell-center}}^{1D}$  is constructed by replacing the  $h_x$ 's in (2.31) with  $h_z$ 's.

In a similar way, we use the 1D cell-center derivative,

$$\mathbf{D}_{x,\text{cell-center} \rightarrow \text{node}}^{1D} = \begin{bmatrix} \frac{1}{h_{x,1}} & & & & \\ \frac{-2}{(h_{x,1}+h_{x,2})} & \frac{2}{(h_{x,1}+h_{x,2})} & & & \\ & \ddots & \ddots & & \\ & & \frac{-2}{(h_{x,m-1}+h_{x,m})} & \frac{2}{(h_{x,m-1}+h_{x,m})} & \\ & & & \frac{-1}{h_{x,m}} & \end{bmatrix}, \quad (2.36)$$

to build the 2D difference matrices for the mappings edge $\rightarrow$ node and cell-center $\rightarrow$ edge:

$$\mathbf{D}_{x,z\text{-edge} \rightarrow \text{node}} = \mathbf{I}_{n+1} \otimes \mathbf{D}_{x,\text{cell-center} \rightarrow \text{node}}^{1D} \quad (2.37a)$$

$$\mathbf{D}_{x,\text{cell-center} \rightarrow x\text{-edge}} = \mathbf{I}_n \otimes \mathbf{D}_{x,\text{cell-center} \rightarrow \text{node}}^{1D} \quad (2.37b)$$

$$\mathbf{D}_{z,x\text{-edge} \rightarrow \text{node}} = \mathbf{D}_{z,\text{cell-center} \rightarrow \text{node}}^{1D} \otimes \mathbf{I}_{m+1} \quad (2.37c)$$

$$\mathbf{D}_{z,\text{cell-center} \rightarrow z\text{-edge}} = \mathbf{D}_{z,\text{cell-center} \rightarrow \text{node}}^{1D} \otimes \mathbf{I}_m \quad (2.37d)$$

Note that the use of the matrices (2.31) and (2.36) ensures that the Neumann



condition  $\partial_{\hat{n}}u = 0$ , and the Dirichlet condition,  $u(x, z) = 0$ , are imposed automatically at the boundaries and so the boundary condition (2.2c) is already taken care of.

Now, with all four types of derivatives in our arsenal, it is straightforward to construct the curl operator that acts on the electric field  $\tilde{E}$ :

$$\mathbf{CURL}_{\tilde{E}}(k_y) = \begin{bmatrix} 0 & -\mathbf{D}_{\text{nd}}^{\text{xe}} & -ik_y\mathbf{I} \\ \mathbf{D}_{\text{ze}}^{\text{cc}} & 0 & -\mathbf{D}_{\text{xe}}^{\text{cc}} \\ ik_y\mathbf{I} & \mathbf{D}_{\text{nd}}^{\text{ze}} & 0 \end{bmatrix} \quad (2.38)$$

and the curl for the magnetic field  $\tilde{H}$ :

$$\mathbf{CURL}_{\tilde{H}}(k_y) = \begin{bmatrix} 0 & -\mathbf{D}_{\text{cc}}^{\text{ze}} & -ik_y\mathbf{I} \\ \mathbf{D}_{\text{xe}}^{\text{nd}} & 0 & -\mathbf{D}_{\text{ze}}^{\text{nd}} \\ ik_y\mathbf{I} & \mathbf{D}_{\text{cc}}^{\text{xe}} & 0 \end{bmatrix}. \quad (2.39)$$

For clarity, we have simplified  $\mathbf{D}_{A \rightarrow B}$  to  $\mathbf{D}_A^B$  and abbreviated  $x$ -edge,  $z$ -edge, node and cell-center with xe, ze, nd and cc, respectively.

It is worth mentioning that if all the cells are of the same size, then  $\mathbf{D}_{x, \text{nd} \rightarrow \text{cc}}^{1D} = -(\mathbf{D}_{x, \text{cc} \rightarrow \text{nd}}^{1D})^T$ , and the two curl operators become adjoints of each other:

$$\mathbf{CURL}_{\tilde{H}} = \mathbf{CURL}_{\tilde{E}}^T. \quad (2.40)$$

And as a consequence, the resulting matrix of the system is also symmetric.

As we will see later, the regularization of the inverse problem requires the gradient of the conductivity to be computed. To achieve this, we put the cell-center  $\rightarrow$  edge derivatives (2.37b) and (2.37d) together to obtain

$$\mathbf{GRAD}_{\text{cell-center} \rightarrow \text{edge}} = \begin{bmatrix} \mathbf{D}_{\text{cc}}^{\text{xe}} \\ \mathbf{D}_{\text{cc}}^{\text{ze}} \end{bmatrix}. \quad (2.41)$$

### 2.4.3 Discretization of the material averaging

In this section, we construct the material averaging matrix, which averages the cell-centered conductivity and maps it to the edges and nodes where it is evaluated. For this, we use the nearest neighbor extrapolation to obtain the first-order 1D cell-center averaging:

$$\sigma_{i,j+\frac{1}{2}} = \frac{\sigma_{i-\frac{1}{2},j+\frac{1}{2}}h_i + \sigma_{i+\frac{1}{2},j+\frac{1}{2}}h_{i-1}}{h_{i-1} + h_i} + \mathcal{O}(h). \quad (2.42)$$

However, when the conductivity has jumps at the cell boundaries where its derivative is not defined, this approximation is only  $\mathcal{O}(1)$ .

To build the averaging matrix, we note that the 1D averaging has the same sparsity pattern as the 1D cell-center difference. And so to obtain the 2D average, we use the sparsity pattern of Eqn. (2.36) to get

$$\mathbf{A}_{x,\text{cell-center} \rightarrow \text{node}}^{1D} = \begin{bmatrix} \frac{1}{2} & & & & \\ \frac{h_{x,2}}{(h_{x,1}+h_{x,2})} & \frac{h_{x,1}}{(h_{x,1}+h_{x,2})} & & & \\ & \ddots & \ddots & & \\ & & \frac{h_{x,m}}{(h_{x,m-1}+h_{x,m})} & \frac{h_{x,m-1}}{(h_{x,m-1}+h_{x,m})} & \\ & & & \frac{1}{2} & \end{bmatrix}, \quad (2.43)$$

and then use the Kronecker product like before to construct the 2D averaging operator:

$$\mathbf{A}_{\text{cell-center} \rightarrow \text{x-edge}} = \mathbf{I}_n \otimes \mathbf{A}_{x,\text{cell-center} \rightarrow \text{node}}^{1D} \quad (2.44a)$$

$$\mathbf{A}_{\text{cell-center} \rightarrow \text{z-edge}} = \mathbf{A}_{z,\text{cell-center} \rightarrow \text{node}}^{1D} \otimes \mathbf{I}_m \quad (2.44b)$$

$$\mathbf{A}_{\text{cell-center} \rightarrow \text{node}} = \mathbf{A}_{z,\text{cell-center} \rightarrow \text{node}}^{1D} \otimes \mathbf{A}_{x,\text{cell-center} \rightarrow \text{node}}^{1D} \quad (2.44c)$$

#### 2.4.4 Matrix representation of Maxwell's equations

For each combination in the set of parameters  $\{x_s, \omega, k_y\}$ , the following discrete 2.5D Maxwell's equations have to be solved:

$$\mathbf{CURL}_{\tilde{E}} \tilde{\mathbf{E}} - i\omega\mu_0 \tilde{\mathbf{H}} = 0 \quad (2.45a)$$

$$\mathbf{CURL}_{\tilde{H}} \tilde{\mathbf{H}} - \text{diag}(\mathbf{A}_1 \boldsymbol{\sigma}) \tilde{\mathbf{E}} = \mathbf{A}_2(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_0 \quad (2.45b)$$

where

$$\mathbf{A}_1 := \begin{bmatrix} \mathbf{A}_{cc}^{ze} \\ \mathbf{A}_{cc}^{nd} \\ \mathbf{A}_{cc}^{xe} \end{bmatrix} \quad (2.46)$$

and

$$\mathbf{A}_2(\boldsymbol{\sigma}) = \begin{bmatrix} \mathbf{A}_{cc}^{ze} \text{diag}(\boldsymbol{\sigma}) & 0 & 0 \\ 0 & \mathbf{A}_{cc}^{nd} \text{diag}(\boldsymbol{\sigma}) & 0 \\ 0 & 0 & \mathbf{A}_{cc}^{xe} \text{diag}(\boldsymbol{\sigma}) \end{bmatrix}. \quad (2.47)$$

To solve the system, we can isolate for  $\tilde{\mathbf{E}}$  in (2.45) and rearrange the equations to obtain:

$$[\mathbf{CURL}_{\tilde{H}} \mathbf{CURL}_{\tilde{E}} + c \text{diag}(\mathbf{A}_1 \boldsymbol{\sigma})] \tilde{\mathbf{E}} = -c \mathbf{A}_2(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_0 \quad (2.48a)$$

$$\tilde{\mathbf{H}} = \mathbf{CURL}_{\tilde{E}} \tilde{\mathbf{E}} / -c \quad (2.48b)$$

where  $c = -i\omega\mu_0$ . Once the matrix  $\mathbf{A} = \mathbf{CURL}_{\tilde{H}} \mathbf{CURL}_{\tilde{E}} + c \text{diag}(\mathbf{A}_1 \boldsymbol{\sigma})$  is set up, the system can conveniently be solved with the Matlab backslash:  $\tilde{\mathbf{E}} = \mathbf{A} \setminus (-c \mathbf{A}_2(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_0)$ .

Using (2.48) to solve for all field components at once can be computationally expensive when the mesh is dense. Another, simpler, way to a solution is to express (2.45) in terms of the  $y$ -component of the electric and magnetic fields. This approach reduces the size of the linear system to be solved. Once a solution is found, the other field components can be computed straightforwardly using numerical differentiation. To demonstrate, we start by writing

out explicitly the matrices acting on each field component:

$$-ik_y \tilde{\mathbf{E}}_z - \mathbf{D}_{\text{nd}}^{\text{xe}} \tilde{\mathbf{E}}_y - i\omega\mu_0 \tilde{\mathbf{H}}_x = 0 \quad (2.49\text{a})$$

$$\mathbf{D}_{\text{ze}}^{\text{cc}} \tilde{\mathbf{E}}_x - \mathbf{D}_{\text{xe}}^{\text{cc}} \tilde{\mathbf{E}}_z - i\omega\mu_0 \tilde{\mathbf{H}}_y = 0 \quad (2.49\text{b})$$

$$\mathbf{D}_{\text{nd}}^{\text{ze}} \tilde{\mathbf{E}}_y + ik_y \tilde{\mathbf{E}}_x - i\omega\mu_0 \tilde{\mathbf{H}}_z = 0 \quad (2.49\text{c})$$

$$-ik_y \tilde{\mathbf{H}}_z - \mathbf{D}_{\text{cc}}^{\text{ze}} \tilde{\mathbf{H}}_y - \text{diag}(\mathbf{A}_{\text{cc}}^{\text{ze}} \boldsymbol{\sigma}) \tilde{\mathbf{E}}_x = \mathbf{A}_{\text{cc}}^{\text{ze}} \text{diag}(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_{0,x} \quad (2.49\text{d})$$

$$\mathbf{D}_{\text{xe}}^{\text{nd}} \tilde{\mathbf{H}}_x - \mathbf{D}_{\text{ze}}^{\text{nd}} \tilde{\mathbf{H}}_z - \text{diag}(\mathbf{A}_{\text{cc}}^{\text{nd}} \boldsymbol{\sigma}) \tilde{\mathbf{E}}_y = \mathbf{A}_{\text{cc}}^{\text{nd}} \text{diag}(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_{0,y} \quad (2.49\text{e})$$

$$\mathbf{D}_{\text{cc}}^{\text{xe}} \tilde{\mathbf{H}}_y + ik_y \tilde{\mathbf{H}}_x - \text{diag}(\mathbf{A}_{\text{cc}}^{\text{xe}} \boldsymbol{\sigma}) \tilde{\mathbf{E}}_z = \mathbf{A}_{\text{cc}}^{\text{xe}} \text{diag}(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_{0,z} \quad (2.49\text{f})$$

Now, after some algebra, we can eliminate all field components except  $\tilde{\mathbf{E}}_y$  and  $\tilde{\mathbf{H}}_y$  from the equations and express the linear system for these remaining unknowns as

$$\begin{bmatrix} \tilde{\mathbf{E}}_y \\ \tilde{\mathbf{H}}_y \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix}^{-1} \begin{bmatrix} ik_y \mathbf{D}_{\text{ze}}^{\text{nd}} \boldsymbol{\Gamma}_{\text{ze}}^{-1} & \mathbf{I} & ik_y \mathbf{D}_{\text{xe}}^{\text{nd}} \boldsymbol{\Gamma}_{\text{xe}}^{-1} \\ -\mathbf{D}_{\text{ze}}^{\text{cc}} \boldsymbol{\Gamma}_{\text{ze}}^{-1} & 0 & \mathbf{D}_{\text{xe}}^{\text{cc}} \boldsymbol{\Gamma}_{\text{xe}}^{-1} \end{bmatrix} \mathbf{A}_2(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_0 \quad (2.50)$$

Here, the blocks of  $\mathbf{S}$  are defined as

$$\mathbf{S}_{11} = \mathbf{D}_{\text{xe}}^{\text{nd}} \boldsymbol{\Sigma}_{\text{xe}} \mathbf{D}_{\text{nd}}^{\text{xe}} + \mathbf{D}_{\text{ze}}^{\text{nd}} \boldsymbol{\Sigma}_{\text{ze}} \mathbf{D}_{\text{nd}}^{\text{ze}} - \text{diag}(\mathbf{A}_{\text{cc}}^{\text{nd}} \boldsymbol{\sigma}) \quad (2.51\text{a})$$

$$\mathbf{S}_{12} = ik_y \mathbf{D}_{\text{xe}}^{\text{nd}} \boldsymbol{\Gamma}_{\text{xe}}^{-1} \mathbf{D}_{\text{cc}}^{\text{xe}} - ik_y \mathbf{D}_{\text{ze}}^{\text{nd}} \boldsymbol{\Gamma}_{\text{ze}}^{-1} \mathbf{D}_{\text{cc}}^{\text{ze}} \quad (2.51\text{b})$$

$$\mathbf{S}_{21} = -\frac{ik_y}{c} \mathbf{D}_{\text{xe}}^{\text{cc}} \boldsymbol{\Gamma}_{\text{xe}}^{-1} \mathbf{D}_{\text{nd}}^{\text{xe}} + \frac{ik_y}{c} \mathbf{D}_{\text{ze}}^{\text{cc}} \boldsymbol{\Gamma}_{\text{ze}}^{-1} \mathbf{D}_{\text{nd}}^{\text{ze}} \quad (2.51\text{c})$$

$$\mathbf{S}_{22} = \mathbf{D}_{\text{xe}}^{\text{cc}} \boldsymbol{\Gamma}_{\text{xe}}^{-1} \mathbf{D}_{\text{cc}}^{\text{xe}} + \mathbf{D}_{\text{ze}}^{\text{cc}} \boldsymbol{\Gamma}_{\text{ze}}^{-1} \mathbf{D}_{\text{cc}}^{\text{ze}} - \mathbf{I}_{m \times n} \quad (2.51\text{d})$$

where  $\boldsymbol{\Gamma}_{\text{xe}} = \text{diag}(\mathbf{A}_{\text{cc}}^{\text{xe}} \boldsymbol{\gamma})$ ,  $\boldsymbol{\Gamma}_{\text{ze}} = \text{diag}(\mathbf{A}_{\text{cc}}^{\text{ze}} \boldsymbol{\gamma})$ ,  $\boldsymbol{\Sigma}_{\text{xe}} = \text{diag}(\mathbf{A}_{\text{cc}}^{\text{xe}} \boldsymbol{\sigma}) \boldsymbol{\Gamma}_{\text{xe}}^{-1}$ ,  $\boldsymbol{\Sigma}_{\text{ze}} = \text{diag}(\mathbf{A}_{\text{cc}}^{\text{ze}} \boldsymbol{\sigma}) \boldsymbol{\Gamma}_{\text{ze}}^{-1}$  and  $\boldsymbol{\gamma} = k_y^2 \mathbf{1} - c \boldsymbol{\sigma}$  where  $\mathbf{1}$  is a vector of ones. Once the

Eqn. (2.50) is solved, the remaining field components can be recovered from

$$\begin{bmatrix} \tilde{\mathbf{E}}_x \\ \tilde{\mathbf{E}}_z \end{bmatrix} = \begin{bmatrix} \mathbf{\Gamma}_{ze}^{-1} & 0 \\ 0 & \mathbf{\Gamma}_{xe}^{-1} \end{bmatrix} \begin{bmatrix} ik_y \mathbf{D}_{nd}^{ze} & c \mathbf{D}_{cc}^{ze} \\ ik_y \mathbf{D}_{nd}^{xe} & -c \mathbf{D}_{cc}^{xe} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{E}}_y \\ \tilde{\mathbf{H}}_y \end{bmatrix} + c \begin{bmatrix} \mathbf{\Gamma}_{ze}^{-1} & 0 & 0 \\ 0 & 0 & \mathbf{\Gamma}_{xe}^{-1} \end{bmatrix} \mathbf{A}_2(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_0, \quad (2.52)$$

$$\begin{bmatrix} \tilde{\mathbf{H}}_x \\ \tilde{\mathbf{H}}_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_{xe} \mathbf{D}_{nd}^{xe} & ik_y \mathbf{\Gamma}_{xe}^{-1} \mathbf{D}_{cc}^{xe} \\ -\boldsymbol{\Sigma}_{ze} \mathbf{D}_{nd}^{ze} & ik_y \mathbf{\Gamma}_{ze}^{-1} \mathbf{D}_{cc}^{ze} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{E}}_y \\ \tilde{\mathbf{H}}_y \end{bmatrix} + ik_y \begin{bmatrix} 0 & 0 & -\mathbf{\Gamma}_{xe}^{-1} \\ \mathbf{\Gamma}_{ze}^{-1} & 0 & 0 \end{bmatrix} \mathbf{A}_2(\boldsymbol{\sigma}) \tilde{\mathbf{E}}_0. \quad (2.53)$$

Compared to solving for all components simultaneously, this method only requires solving for two field components. Cutting the number of variables by a third significantly improves the computation time as the number of matrix entries is now shrunk to half of that in the full system. Additionally, performing the elimination after the system is discretized preserves the order of accuracy of the original approximation, as rearranging the matrices and variables with algebra does not change the order of accuracy. On the contrary, if the system is first reduced and then discretized, the order of accuracy of the remaining field components is not guaranteed to be the same as that of  $\tilde{\mathbf{E}}_y$  and  $\tilde{\mathbf{H}}_y$ .

In this work, we have used Matlab's backslash direct solver for the solution of the matrix system. For many right hand sides like we have here, it is more effective to use direct solving methods. However, iterative methods like conjugate gradient (CG) and multigrid become necessary when dealing with larger systems, for example when the fields are 3D functions.

## 2.5 Solving the adjoint problem

As we will see later, solving the inverse problem requires solving the adjoint problem, which has the form:

$$\mathbf{A}^T \mathbf{x} = \mathbf{rhs}. \quad (2.54)$$

This problem is relatively easy if matrix  $\mathbf{A}$  is small and Matlab's backslash is used to obtain the numerical solution, as the adjoint matrix  $\mathbf{A}^T$  can be formed easily. However, if the reduced system is used for solution, the setup of the problem becomes a bit more involved.

To set up the adjoint problem for the reduced system, we start by writing out  $\mathbf{A}^T$  explicitly,

$$\begin{aligned}
\mathbf{A}^T &= \mathbf{CURL}_{\tilde{E}}^T \mathbf{CURL}_{\tilde{H}}^T - \text{cdiag}(\mathbf{A}_1 \boldsymbol{\sigma}) \\
&= \begin{bmatrix} 0 & -\mathbf{D}_{\text{nd}}^{\text{xe}} & -ik_y \\ \mathbf{D}_{\text{ze}}^{\text{cc}} & 0 & -\mathbf{D}_{\text{xe}}^{\text{cc}} \\ ik_y & \mathbf{D}_{\text{nd}}^{\text{ze}} & 0 \end{bmatrix}^T \begin{bmatrix} 0 & -\mathbf{D}_{\text{cc}}^{\text{ze}} & -ik_y \\ \mathbf{D}_{\text{xe}}^{\text{nd}} & 0 & -\mathbf{D}_{\text{ze}}^{\text{nd}} \\ ik_y & \mathbf{D}_{\text{cc}}^{\text{xe}} & 0 \end{bmatrix}^T - \text{cdiag}(\mathbf{A}_1 \boldsymbol{\sigma}) \\
&= \begin{bmatrix} 0 & (\mathbf{D}_{\text{ze}}^{\text{cc}})^T & -ik_y \\ -(\mathbf{D}_{\text{nd}}^{\text{xe}})^T & 0 & (\mathbf{D}_{\text{nd}}^{\text{ze}})^T \\ ik_y & -(\mathbf{D}_{\text{xe}}^{\text{cc}})^T & 0 \end{bmatrix} \begin{bmatrix} 0 & (\mathbf{D}_{\text{xe}}^{\text{nd}})^T & -ik_y \\ -(\mathbf{D}_{\text{cc}}^{\text{ze}})^T & 0 & (\mathbf{D}_{\text{cc}}^{\text{xe}})^T \\ ik_y & -(\mathbf{D}_{\text{ze}}^{\text{nd}})^T & 0 \end{bmatrix} \\
&\quad - \text{cdiag}(\mathbf{A}_1 \boldsymbol{\sigma}) \tag{2.55}
\end{aligned}$$

And now by noticing that the blocks of  $\mathbf{CURL}_{\tilde{E}}^T$  and  $\mathbf{CURL}_{\tilde{H}}^T$  have the same sparsity pattern, as well as those of  $\mathbf{CURL}_{\tilde{H}}^T$  and  $\mathbf{CURL}_{\tilde{E}}^T$ , we can now solve the adjoint system by making the substitutions indicated in Table (2.3) to the operators in Eqns. (2.50)-(2.53). If the grid is uniform in size, then  $\mathbf{D}_{\text{A}}^{\text{B}} = -(\mathbf{D}_{\text{B}}^{\text{A}})^T$  for any type of mapping, and only the last substitution is needed.

## 2.6 Solving for multiple RHS

Compared to other survey methods, the number of data points acquired for each source is much smaller in AEM systems. Each source in a frequency domain AEM survey is identified by its location and frequency. And for each source, only one EM sounding measurement is made. If an iterative method is used, this one to one ratio between the number of sources and

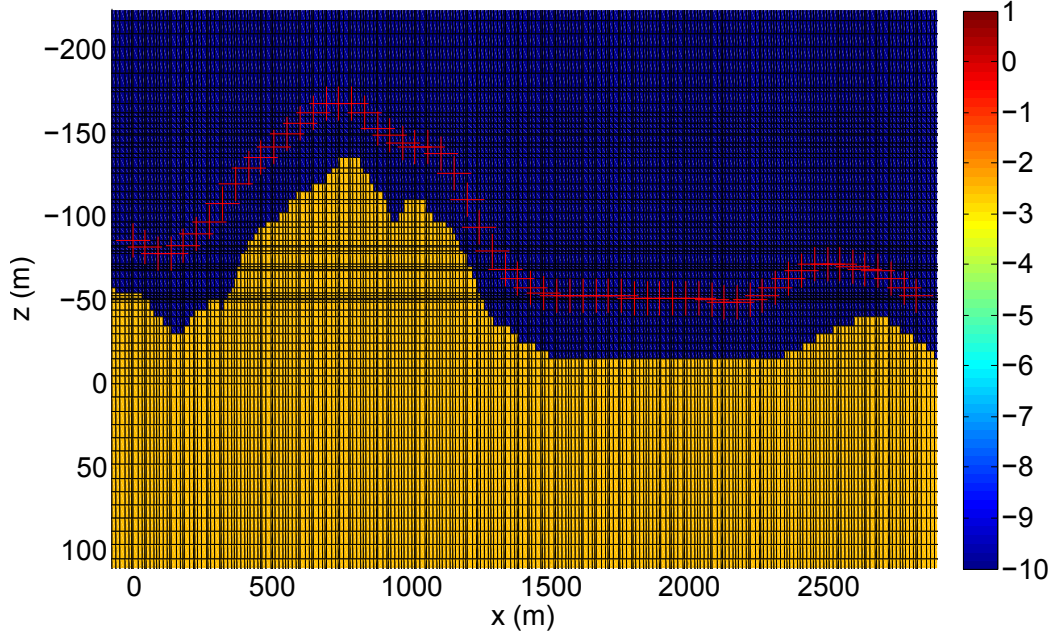
**Table 2.3:** Substitution of operators for the setup of the adjoint problem

$\mathbf{A}$	$\rightarrow$	$\mathbf{A}^T$
$\mathbf{D}_{cc}^{ze}$	$\rightarrow$	$-(\mathbf{D}_{ze}^{cc})^T$
$\mathbf{D}_{xe}^{nd}$	$\rightarrow$	$-(\mathbf{D}_{nd}^{xe})^T$
$\mathbf{D}_{ze}^{nd}$	$\rightarrow$	$-(\mathbf{D}_{nd}^{ze})^T$
$\mathbf{D}_{cc}^{xe}$	$\rightarrow$	$-(\mathbf{D}_{xe}^{cc})^T$
$\mathbf{D}_{nd}^{xe}$	$\rightarrow$	$-(\mathbf{D}_{xe}^{nd})^T$
$\mathbf{D}_{ze}^{cc}$	$\rightarrow$	$-(\mathbf{D}_{cc}^{ze})^T$
$\mathbf{D}_{xe}^{cc}$	$\rightarrow$	$-(\mathbf{D}_{cc}^{xe})^T$
$\mathbf{D}_{nd}^{ze}$	$\rightarrow$	$-(\mathbf{D}_{ze}^{nd})^T$
$c$	$\rightarrow$	$-c$

the number of receivers can make the computation quite costly, as the total number of forward solves ( $\mathbf{x} = \mathbf{A}^{-1}\mathbf{rhs}$ ) or transposed solves ( $\mathbf{x} = \mathbf{A}^{-T}\mathbf{rhs}$ ) scales with the number of data points. However, using direct solver, multiple sources can be solved at about the same time as solving for a single source. In this case, the majority of the computational effort goes into LU factorizing the forward matrix and the cost of solving for each source with forward/backward-substitution is only a fraction of the cost of the factorization. However, to take advantage of this, the sources have to share the same linear system. Namely, the mesh needs to be set up such that it contains all source locations. With such a mesh, the number of forward solves required does not scale linearly with the number of sources, but only with the number of sounding frequencies  $N_\omega$  and the number of optimum wavenumbers  $N_{k_y}$ .

In Matlab, multiple right hand sides can be solved by concatenating the different RHS column vectors into a RHS matrix. The matrix is passed to backslash and the outputs are stored in a matrix whose columns are the solutions of the corresponding columns in the RHS matrix.

A typical mesh generated for multiple source locations can be seen in Fig-



**Figure 2.2:** An example of a mesh generated for AEM inversion. The color shows the initial guess of the log of the conductivity model and the crosses represent the source positions.

ure (2.2). The coarser padding cells, which are used to extend the core mesh boundary so to reduce boundary effects, are not shown in the figure. In the mesh, we can see that in order to incorporate the source altitudes accurately, cells around the sources have a  $h_z$  less than 1 meter. This high tolerance is necessary, as a difference of a meter in the altitude of the source can affect the computed response by a couple of percent, creating discretization errors that could dominate other sources of error in the data. If the discretization error is not carefully controlled, serious error underestimation can occur in the inversion process. This can lead to unsuccessful inversions.

Now that we know how to predict the magnetic response from any given conductivity distribution, we can proceed to apply the standard inversion framework to build an algorithm to invert AEM data to conductivity models.



# Chapter 3

## Inversion framework

The objective of an inversion algorithm is to convert observed data into a model of a system property such that the data can be explained by the model. But since most inverse problems are ill-posed, meaning that they are not well-posed in the sense defined by Jacques Hadamard, it is necessary to restrict the possible recovered solutions to the most probable and most reasonable ones. What counts as a good recovered model depends on the application. For example, in geophysical settings where physical properties of the earth are recovered, a good model should show few features on a length scale longer than the expected size of the anomalies, while the recovered anomalies should have a size typical for structures created by similar geological formations. By contrast, in image deblurring, a good reconstruction adds sharper contrasts to the features of the image and makes it richer in details. In inverse problems, the different application dependent requirements on the solution can be achieved by adding the appropriate regularizations to the data-fitting via mathematical constraints.

### 3.1 Objective function

Given the magnetic field response,  $\mathbf{d}^{\text{obs}}$ , observed above a survey line, the goal of an EM inversion is to recover a smooth conductivity model that minimizes the data misfit. Similar to other EM inverse problems, the log of the conductivity  $m = \log_{10}(\sigma)$  is used and the data fitting is regularized by incorporating *a priori* information into the objective function. Here, we regularize with the most commonly used constraints in geophysical inversions: we require that the model is smooth and close to a reference model. To accomplish this, we add the Tikhonov regularization to the data misfit:

$$\min_m \phi(m) = \min_m \phi_d(m) + \beta \phi_m(m) \quad (3.1)$$

where

$$\phi_d(m) = \frac{1}{2} \sum_i \left\| \frac{d_i^{\text{pred}}(m) - d_i^{\text{obs}}}{\sigma_i^{\text{std}}} \right\|_2^2 \quad (3.2)$$

is the data misfit and  $i$  is an index for all combinations of  $\omega$  and  $\mathbf{x}_s$ .

$$\phi_m(m) = \frac{1}{2} (\alpha_s \|m - m_{\text{ref}}\|_2^2 + \alpha_x \|\partial_x(m - m_{\text{ref}})\|_2^2 + \alpha_z \|\partial_z(m - m_{\text{ref}})\|_2^2) \quad (3.3)$$

is the model misfit and  $\beta$  is the regularization parameter. The predicted data,  $d^{\text{pred}}$ , is computed by extracting the secondary field due to  $\sigma = e^m$  at the receivers using the measurement matrices  $\mathbf{P}_i$ :

$$d_i^{\text{pred}} = \mathbf{P}_i \vec{\mathbf{H}}(x_s, \omega) = \mathbf{P}_i \sum_j^{N_{k_y}} g_j \tilde{\mathbf{H}}(x_s, \omega, k_{y,j}) \quad (3.4)$$

The data misfit term is the squared sum of the misfit of the data points normalized by the noise standard deviation  $\sigma_i^{\text{std}}$ . The second term is the smallness regularization and the corresponding parameter,  $\alpha_s$ , controls the length scale of the details in the inverted model. The smoothness in  $x$  and  $z$ -directions is controlled by the parameters  $\alpha_x$  and  $\alpha_z$  in the third and forth

terms, respectively and minimizing them has the effect of reducing sharp jumps in the solution so that a smooth model is produced.

## 3.2 Sensitivity matrix

As we will discuss later, the objective function is minimized by a quasi-Newton optimization technique. In this section, we derive the gradient of the objective function needed to compute the search step for the optimization.

To start, we discretize the objective function and obtain

$$\begin{aligned}\phi(\mathbf{m}) &= \frac{1}{2} \|\mathbf{Q}^{-1}(\mathbf{d}^{\text{pred}}(\mathbf{m}) - \mathbf{d}^{\text{obs}})\|_2^2 \\ &+ \frac{\beta}{2} (\alpha_s \|\mathbf{m} - \mathbf{m}_{\text{ref}}\|_2^2 + \alpha_x \|\mathbf{D}_x(\mathbf{m} - \mathbf{m}_{\text{ref}})\|_2^2 + \alpha_z \|\mathbf{D}_z(\mathbf{m} - \mathbf{m}_{\text{ref}})\|_2^2) .\end{aligned}\tag{3.5}$$

where  $\mathbf{Q} = \text{diag}(\boldsymbol{\sigma}_{\text{std}})$ , and  $\mathbf{D}_x = \mathbf{D}_{\text{cc}}^{\text{xe}}$  and  $\mathbf{D}_z = \mathbf{D}_{\text{cc}}^{\text{ze}}$  as shown in Chapter 2.4.2. Next, we differentiate the objective function with matrix calculus to find the gradient:

$$\begin{aligned}\nabla \phi(\mathbf{m}) &= \left( \frac{\partial \mathbf{d}^{\text{pred}}}{\partial \mathbf{m}} \right)^T (\mathbf{Q}^{-2}(\mathbf{d}^{\text{pred}} - \mathbf{d}^{\text{obs}})) \\ &+ \beta (\alpha_s (\mathbf{m} - \mathbf{m}_{\text{ref}}) + \alpha_x \mathbf{D}_x^T \mathbf{D}_x (\mathbf{m} - \mathbf{m}_{\text{ref}}) + \alpha_z \mathbf{D}_z^T \mathbf{D}_z (\mathbf{m} - \mathbf{m}_{\text{ref}})) \\ &= \sum_i \left( \sum_j^{N_{ky}} g_j \mathbf{J}_{i,j}^T \mathbf{P}_i^T \mathbf{Q}_i^{-2} (\mathbf{d}_i^{\text{pred}} - \mathbf{d}_i^{\text{obs}}) \right) \\ &+ \beta (\alpha_s (\mathbf{m} - \mathbf{m}_{\text{ref}}) + \alpha_x \mathbf{D}_x^T \mathbf{D}_x (\mathbf{m} - \mathbf{m}_{\text{ref}}) + \alpha_z \mathbf{D}_z^T \mathbf{D}_z (\mathbf{m} - \mathbf{m}_{\text{ref}}))\end{aligned}\tag{3.6}$$

In this equation, the sensitivity matrix  $\mathbf{J}_{i,j}$  is given by the derivative of the magnetic field with respect to the model:

$$\mathbf{J}_{i,j} = \frac{\partial \tilde{\mathbf{H}}}{\partial \mathbf{m}} = \frac{\partial \tilde{\mathbf{H}}}{\partial \boldsymbol{\sigma}} \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{m}} = \frac{\text{CURL}_{\tilde{E}}}{-c} \frac{\partial \tilde{\mathbf{E}}}{\partial \boldsymbol{\sigma}} \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{m}}\tag{3.7}$$

where  $\tilde{\mathbf{E}} = \tilde{\mathbf{E}}(x_s, \omega, k_y)$  is given by the forward solution:

$$\tilde{\mathbf{E}} = \mathbf{A}^{-1}(-c\mathbf{A}_2(\boldsymbol{\sigma})\tilde{\mathbf{E}}_0) \quad (3.8)$$

To find its derivative w.r.t to the conductivity model, we apply implicit differentiation to (2.48):

$$\frac{\partial}{\partial \boldsymbol{\sigma}} \left( [\mathbf{CURL}_{\tilde{H}} \mathbf{CURL}_{\tilde{E}} + c \text{diag}(\mathbf{A}_1 \boldsymbol{\sigma})] \tilde{\mathbf{E}} \right) = \frac{\partial}{\partial \boldsymbol{\sigma}} \left( -c\mathbf{A}_2(\boldsymbol{\sigma})\tilde{\mathbf{E}}_0 \right) \quad (3.9)$$

$$\mathbf{A} \frac{\partial \tilde{\mathbf{E}}}{\partial \boldsymbol{\sigma}} + c \frac{\partial}{\partial \boldsymbol{\sigma}} \left( \text{diag}(\mathbf{A}_1 \boldsymbol{\sigma}) \tilde{\mathbf{E}} \right) = \frac{\partial}{\partial \boldsymbol{\sigma}} \left( -c\mathbf{A}_2(\boldsymbol{\sigma})\tilde{\mathbf{E}}_0 \right) \quad (3.10)$$

Now by applying the property  $\text{diag}(\mathbf{a})\mathbf{b} = \text{diag}(\mathbf{b})\mathbf{a}$  twice, once on each side of (3.10), we obtain:

$$\mathbf{A} \frac{\partial \tilde{\mathbf{E}}}{\partial \boldsymbol{\sigma}} + c \frac{\partial}{\partial \boldsymbol{\sigma}} \left( \text{diag}(\tilde{\mathbf{E}}) \mathbf{A}_1 \boldsymbol{\sigma} \right) = -c \frac{\partial}{\partial \boldsymbol{\sigma}} \left( \begin{bmatrix} \mathbf{A}_{cc}^{ze} \text{diag}(\tilde{\mathbf{E}}_{0,x}) \\ \mathbf{A}_{cc}^{nd} \text{diag}(\tilde{\mathbf{E}}_{0,y}) \\ \mathbf{A}_{cc}^{xe} \text{diag}(\tilde{\mathbf{E}}_{0,z}) \end{bmatrix} \boldsymbol{\sigma} \right) \quad (3.11)$$

$$\mathbf{A} \frac{\partial \tilde{\mathbf{E}}}{\partial \boldsymbol{\sigma}} + c \text{diag}(\tilde{\mathbf{E}}) \mathbf{A}_1 = -c \mathbf{A}_3(\tilde{\mathbf{E}}_0) \quad (3.12)$$

$$\frac{\partial \tilde{\mathbf{E}}}{\partial \boldsymbol{\sigma}} = -c \mathbf{A}^{-1} \left( \mathbf{A}_3(\tilde{\mathbf{E}}_0) + \text{diag}(\tilde{\mathbf{E}}) \mathbf{A}_1 \right) \quad (3.13)$$

where we have defined

$$\mathbf{A}_3(\tilde{\mathbf{E}}_0) := \begin{bmatrix} \mathbf{A}_{cc}^{ze} \text{diag}(\tilde{\mathbf{E}}_{0,x}) \\ \mathbf{A}_{cc}^{nd} \text{diag}(\tilde{\mathbf{E}}_{0,y}) \\ \mathbf{A}_{cc}^{xe} \text{diag}(\tilde{\mathbf{E}}_{0,z}) \end{bmatrix}. \quad (3.14)$$

Now, by going back to Eqn. (3.7), we find that:

$$\mathbf{J}_{i,j} = \frac{\partial \tilde{\mathbf{H}}}{\partial \mathbf{m}} = \mathbf{CURL}_{\tilde{E}} \mathbf{A}^{-1} \left( \mathbf{A}_3(\tilde{\mathbf{E}}_0) + \text{diag}(\tilde{\mathbf{E}}) \mathbf{A}_1 \right) \text{diag}(\ln(10)10^{\mathbf{m}}) \quad (3.15)$$

At last, we take the transpose of the sensitivity to obtain

$$\mathbf{J}_{i,j}^T = \left( \frac{\partial \tilde{\mathbf{H}}}{\partial \mathbf{m}} \right)^T = \text{diag}(\ln(10)10^{\mathbf{m}}) \left( \mathbf{A}_3^T(\tilde{\mathbf{E}}_0) + \mathbf{A}_1^T \text{diag}(\overline{\tilde{\mathbf{E}}}) \right) (\mathbf{A}^T)^{-1} \mathbf{CURL}_{\tilde{\mathbf{E}}}^T \quad (3.16)$$

where  $\overline{\tilde{\mathbf{E}}}$  represents the complex conjugate of  $\tilde{\mathbf{E}}$ . We plug this result back into (3.6) to get the gradient of our optimization objective.

From Eqns. (3.6) and (3.16), we can see that each time the gradient is evaluated, two forward solves are needed: one computes the secondary electric field  $\tilde{\mathbf{E}}$  by solving the regular forward problem given by Eqn. (3.8), and the other solves the adjoint problem given by:

$$\mathbf{x} = (\mathbf{A}^T)^{-1} \mathbf{rhs} \quad (3.17)$$

$$\mathbf{rhs} = \mathbf{CURL}_{\tilde{\mathbf{E}}}^T \mathbf{P}_i^T \frac{d_i^{\text{pred}} - d_i^{\text{obs}}}{\sigma_{i,\text{std}}^2} \quad (3.18)$$

Together with the evaluation of the misfit function in each line search iteration, which we will discuss in the next section, the total number of solves involving  $\mathbf{A}$  and  $\mathbf{A}^T$  is  $3N_\omega N_{k_y}$  if one assumes that the misfit function only needs to be checked once for each new search direction.

### 3.3 Method of optimization

In this work, the objective function is minimized using the quasi-Newton iterative method L-BFGS ([16]). This method finds the search direction by approximating the inverse Hessian using updates of the argument and the gradient. As in all iterative method, the optimization is initialized with a starting guess  $\mathbf{m}_0$ . Then, in each subsequent iteration, a new model is computed using

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \alpha_\ell \delta \mathbf{m}_n \quad (3.19)$$

where the line search parameter  $\alpha_\ell$  decreases exponentially with the line search iteration number  $\ell$ :  $\alpha_\ell = \text{const}^\ell$ ,  $\ell = 0, 1, \dots$ , and  $\delta \mathbf{m}_n$  is the search direction given by the quasi-Newton approximate Hessian  $\mathbf{H}_n$  and the gradient:

$$\delta \mathbf{m}_n = -\mathbf{H}_n^{-1} \nabla \phi(\mathbf{m}_n). \quad (3.20)$$

Using the L-BFGS algorithm to approximate the inverse Hessian has the advantage that the sensitivity does not have to be computed explicitly but instead only matrix-vector products involving the Jacobian:

$$\mathbf{J}_{i,j}^T \mathbf{x}, \quad (3.21)$$

are needed [8]. In situations where the full Hessian is too costly to be evaluated, L-BFGS offers a cheap way to compute a good approximation to the inverse Hessian. At each L-BFGS iteration, the search step  $\delta \mathbf{m}_n$ , is computed from an initial guess of the Hessian, for which we use the Hessian of the regularization term

$$\mathbf{H}_m = \alpha_s \mathbf{I} + \alpha_x \mathbf{D}_x^T \mathbf{D}_x + \alpha_z \mathbf{D}_z^T \mathbf{D}_z, \quad (3.22)$$

and the model and gradient updates

$$\mathbf{s}_l = \mathbf{m}_{l+1} - \mathbf{m}_l \quad (3.23)$$

$$\mathbf{y}_l = \nabla \phi(\mathbf{m}_{l+1}) - \nabla \phi(\mathbf{m}_l) \quad (3.24)$$

collected from the previous  $k$  iterations  $n-1, \dots, n-k$ . A history of only  $k = 10$  recent update vectors usually does a good job in obtaining a descent search direction. For those interested, details on implementing the L-BFGS algorithm can be found in [12].

## 3.4 Implementation of the optimization

### 3.4.1 Data structures

In our inversion scheme, we seek to recover the 2D conductivity distribution of the earth directly underneath a survey line by simultaneously inverting soundings made along the line. Good data structures for storing the observed data and the electromagnetic fields are important to making the code readable and easy to run and maintain.

The observed data are characterized by the source position,  $x_s$  and the frequency  $\omega$ . As an example, the data point measured with a source located at position number  $s$  at  $w$ -th frequency can be retrieved in Matlab with: `dataObs(:, s, w)`.

The fields, which are stored as column vectors, are functions of the wavenumbers  $k_y$  as well. The most convenient way is to adapt a four dimensional array structure for the storage. The first dimension will have the length of the stored variable: 1 for the observed data and the total number of edges/nodes/cell-centers for the fields. The second dimension is used to differentiate the sources, the third dimension separates the frequencies and the fourth dimension the different wavenumbers. To extract the field  $\tilde{E}_x$  computed for source number  $s$ , frequency number  $w$  and  $k$ -th wavenumber, we type the command: `Ex{w, k}(:, s)` (the cell structure is used to allow for storage of sparse matrices as Matlab does not support sparse matrices with more than two dimensions).

### 3.4.2 Constrained optimization

By incorporating additional constraints into the optimization, the solution space can be further confined and the recovered solution will resemble more to a realistic real-world model. As mentioned earlier, constraints are derived from *a priori* information and they can come in many different types.

Most commonly, geophysical constraints come from borehole measurements, physical properties of the survey area extracted from previous geological or geophysical studies, and inversion results from other surveys.

In geological settings, the range of observed conductivity/resistivity can span over eight orders of magnitude. Putting restrictions on the value of the conductivity during the optimization is therefore very helpful in producing a realistic model. In the algorithm, the restriction is implemented by the projected gradient method [2]. The method confines the conductivity in each cell of the recovered model to a given interval by applying a projection function to the updated model at each iteration:

$$\mathbf{m}_{n+1} = P_{\Theta}(\mathbf{m}_n + \alpha_{\ell} \delta \mathbf{m}_n), \Theta = \{\mathbf{m} \in \mathbb{R}^n : \mathbf{m}_{\min} \leq \mathbf{m} \leq \mathbf{m}_{\max}\} \quad (3.25)$$

where

$$P_{\Theta}(m(x_i, z_j)) = \begin{cases} m_{\max}(x_i, z_j) & \text{if } m(x_i, z_j) > m_{\max}(x_i, z_j) \\ m_{\min}(x_i, z_j) & \text{if } m(x_i, z_j) < m_{\min}(x_i, z_j) \end{cases} \quad (3.26)$$

with  $m_{\min}$  and  $m_{\max}$  being the user-defined lower and upper bound model, respectively. By bounding each cell individually, one can define the conductivity in those areas that one has confidence in. For example, conductivity data from a drillhole sample can be used to narrow down the conductivity distribution in the region immediately around the borehole core, soil/rock samples on the surface can be used to make better inversion of the overburden and regional geophysical surveys can be used to place a global bound on the entire survey area.

In addition to the ability to restrict the conductivity to be within a range of values, we can also define spatial regions in the model that can be updated in each iteration. This option is implemented in our algorithm by first defining an active region and then modifying the computed gradient as



follows:

$$\nabla\phi(\mathbf{m}_n)_{\text{mod}}^k = \begin{cases} \nabla\phi(\mathbf{m}_n)_k & \text{if } k \leftrightarrow (i, j) \text{ and cell } ij \text{ is in the active area} \\ 0 & \text{otherwise.} \end{cases} \quad (3.27)$$

By doing this, parameters outside the active region are not updated during the optimization and are left the same as in the initial guess. The use of active areas allows one to, for example, control the location in which a potential target can be found. In addition, it also provides an easy way to test the effect of placing a target in different locations on the data fitting.

### 3.4.3 Regularization parameter selection

Choosing the appropriate regularization parameter,  $\beta$ , is essential for a good solution to the regularized minimization problem (3.1). Each strategy of reaching a solution to the problem produces a different  $\beta$ . In geophysical inversions, the most commonly used strategies are the discrepancy principle, the L-curve and generalized cross-validation (GCV). The first approach is usually used when the noise is Gaussian and its standard deviation is known. When the noise level cannot easily be determined, as is often the case, one can turn to the L-curve or GCV. In our algorithm, only the first two approaches are implemented. (The GCV approach requires knowing the full Hessian which is too computationally expensive to obtain for the size of our problem.)

The discrepancy principle recovers a model by selecting the  $\beta$  that converges to a target data misfit, *tmisfit*, equal to

$$tmisfit := \phi_d(\beta^*, m^*) = N_{data} \quad (3.28)$$

where  $N_{data}$  is the number of measurement points and the star  $*$  denotes the optimum value of the variable. Constraining the data misfit in this way can be justified by the assumption of Gaussian noise and the fact that  $\phi_d(m^*)$

follows a  $\chi^2$  distribution and its expected value is

$$E(\phi_d(m^*)) = N_{data}. \quad (3.29)$$

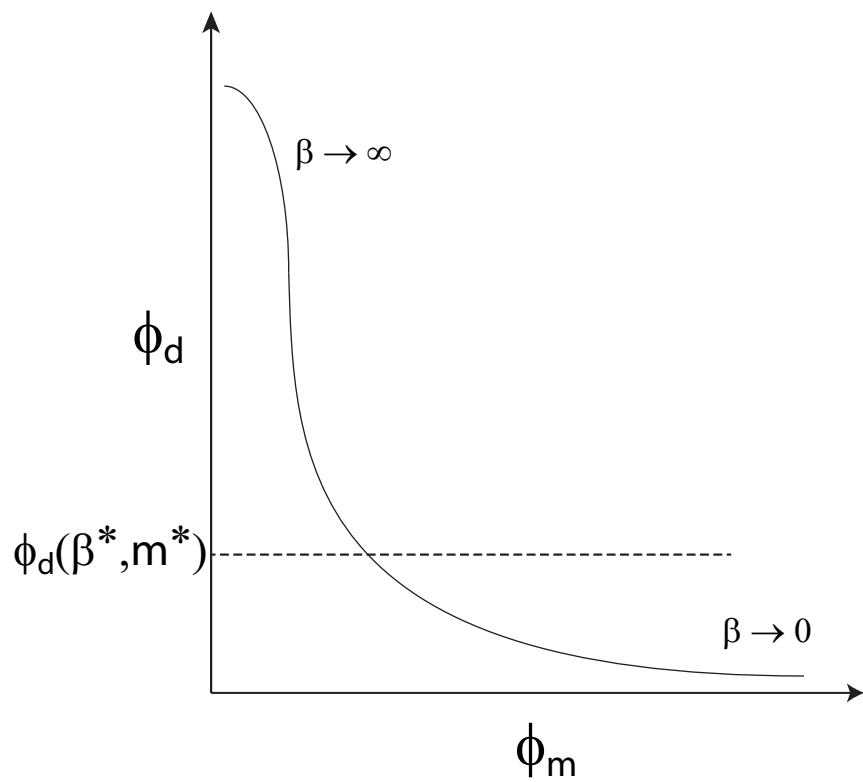
If  $\beta$  is too small, the minimization will try to reduce  $\phi_d$  further, resulting in a solution that fits the noise, i.e., over-fitting. On the other end, if  $\beta$  is too large the optimization renders an under-fitted and featureless model as only the model misfit is minimized and the recovery does not make much use of the observed information. The effect of the regularization parameter on the two misfits at convergence is illustrated by the well-known Tikhonov curve (Fig. 3.1). To find the  $\beta$  that gives *tmisfit*, it is advised to start with a large guess for  $\beta$  and then decrease its value in subsequent optimization trials until the desired target misfit is reached within a defined tolerance. This is because more iterations are required to reach convergence when  $\phi_d$  is the dominant term. Also, since  $\phi_m$  changes logarithmically with  $\beta$ , it is beneficial to generate a series of solutions for logarithmically spaced  $\beta$ 's. The search for  $\beta$  can be automated and the time it takes reduced by using a cooling strategy [7]. However, in the current version of the algorithm,  $\beta$  can only be chosen manually.

In geophysical inversions, the errors come from uncertainties in the measurements as well as error in the forward solution. Estimating the error precisely can therefore be a tough task. In the cases where the noise level is uncertain, one can account for the uncertainty in the estimation by multiplying the target misfit by a factor, *chifact*, such that the *tmisfit* is now

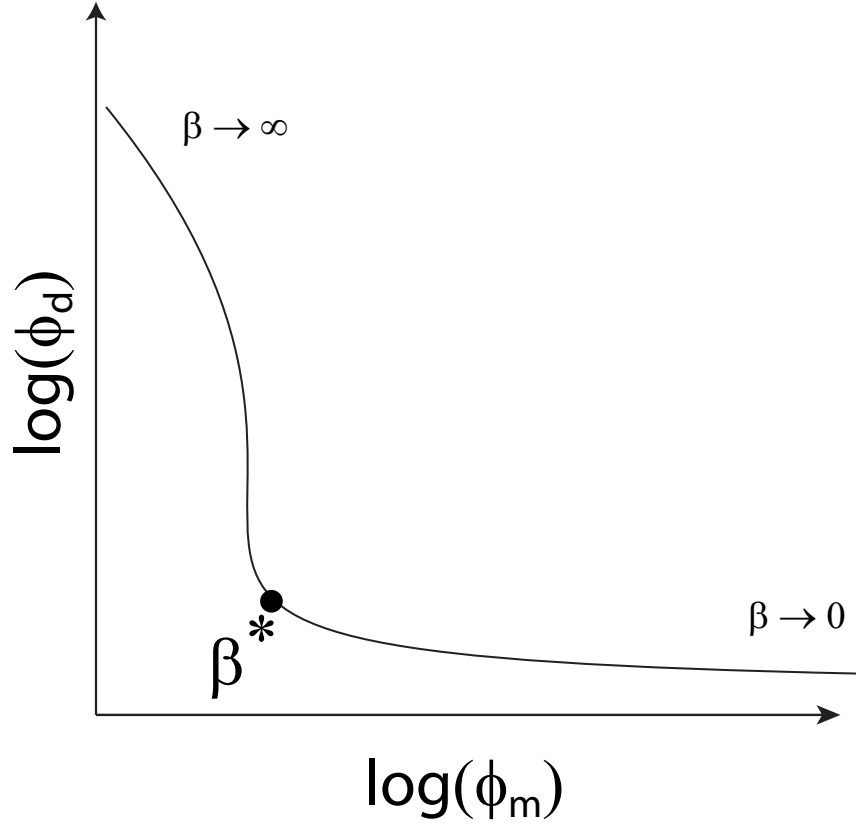
$$tmisfit = chifact \times N_{data}. \quad (3.30)$$

Setting the *chifact* to above (below) 1 allows us to choose another model that fits to a larger (smaller) error. Also, the same strategy can be used to pick a model that underfits or overfits the current noise level.

In the occasions when the noise level is completely unknown or the un-



**Figure 3.1:** Tikhonov curve. For large  $\beta$ , the model norm  $\phi_m$  dominates the objective and the data misfit is underrepresented. The opposite happens when  $\beta$  is small. When the noise level is known, the target misfit,  $\phi_d(\beta^*, m^*)$ , can be used to determine the regularization parameter.



**Figure 3.2:** The L-curve. It can be used to select the regularization parameter  $\beta$  when the noise level is unknown. In this strategy,  $\beta$  is chosen at the point of maximum curvature of the L-curve.

certainly in determining it is too high,  $\beta$  is generally chosen at the point of maximum curvature of the L-curve (see Fig. 3.2). The idea behind this is that in the absence of a good estimate of the noise, we prefer a model that both fits the data well and has a small model norm. In the current implementation of the inversion, this method can only be used by manually generating a series of solutions for a range of  $\beta$  like in the discrepancy strategy.

### 3.4.4 Stopping criteria

Once the quasi-Newton optimization is started, the inverted model is updated at each iteration until a stopping criteria or a convergence condition is reached. In our algorithm, the iteration is terminated when one of the following conditions is met:

- the maximum number of quasi-Newton iterations,  $maxit$ , is exceeded
- the size of the search step is small enough;  $\max(\text{abs}(\alpha_\ell \delta \mathbf{m}_n)) \leq tol\_step$
- the change in the objective function is small enough;  $|\phi(\mathbf{m}_{n+1}) - \phi(\mathbf{m}_n)| / |\phi(\mathbf{m}_n)| \leq tol_\phi$
- the norm of the gradient is small enough;  $\|\nabla \phi(\mathbf{m}_{n+1})\| / \|\nabla \phi(\mathbf{m}_0)\| \leq tol$

# Chapter 4

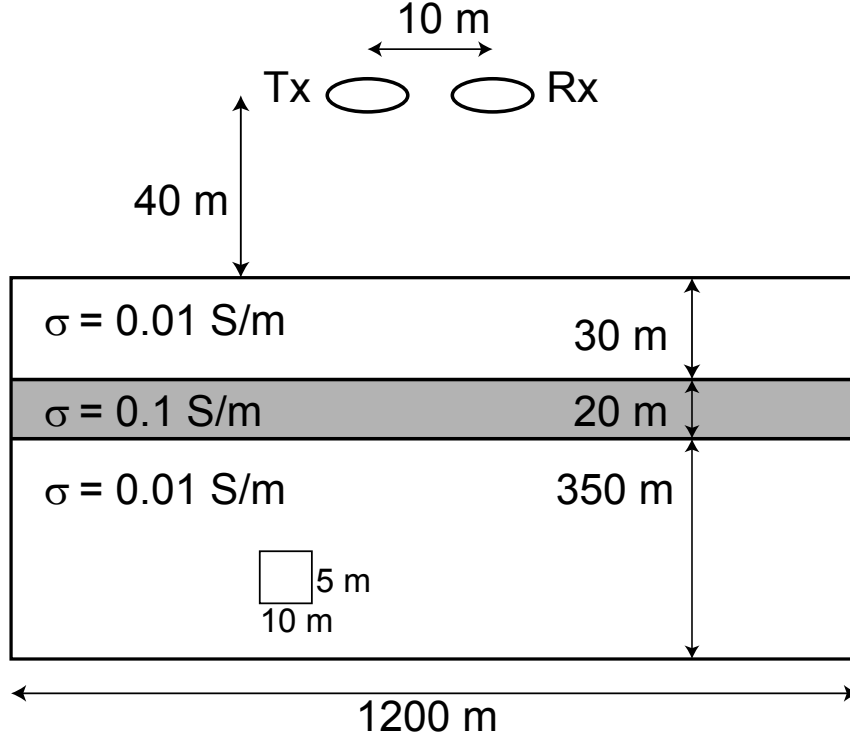
## Results

To evaluate the numerical accuracy of our forward modeling, a known 1D analytical solution was used to benchmark the 2.5D implementation. To assess the reliability of our inversion, we tested the algorithm on synthetic data and field data from a RESOLVE AEM survey. The inverted models were compared to the synthetic 2D models and also to models from a well-established 1D inversion supplied by Fugro.

### 4.1 Testing the forward solution

The numerical accuracy of our algorithm was tested by computing the response from a synthetic layered earth model and then comparing the results against the 1D analytical solution [4, 23]. The model is shown in Fig. 4.1; it consists of a 20 m thick conductive layer of  $10^{-1}$  S/m buried 30 m under the surface of a  $10^{-2}$  S/m half-space. Figure 4.2 shows the 2D inphase and quadrature response. The frequencies used in the computation spanned a range that is typically used in AEM systems and the response was computed 10 m away from the transmitter and 40 m above the ground.

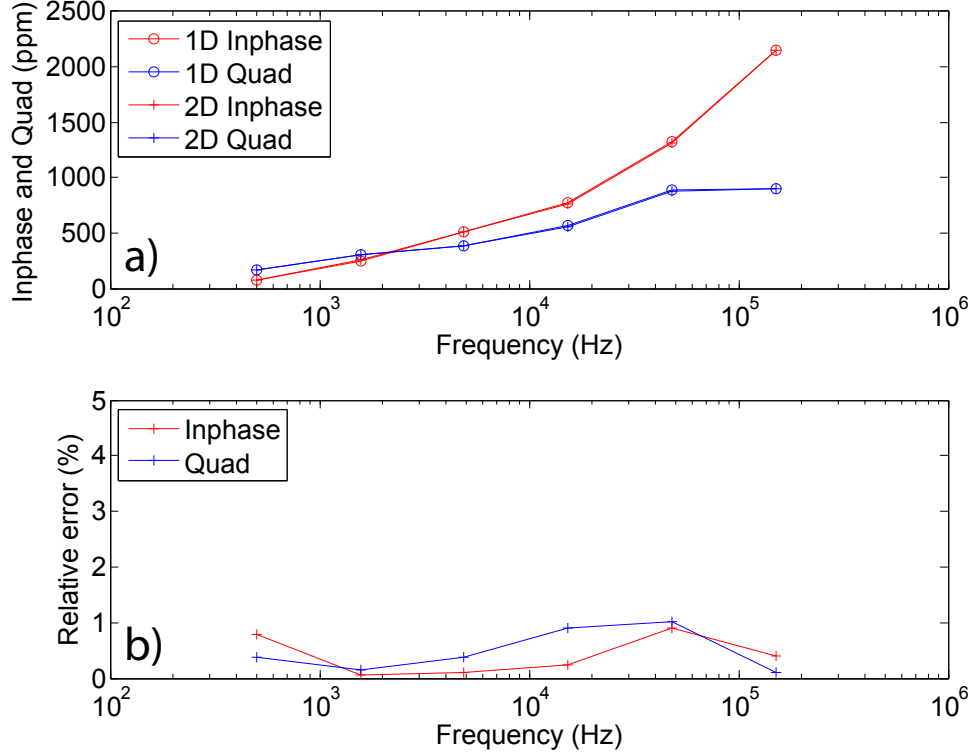
Our results show that with only six optimized wavenumbers and a cell size of  $10\text{ m} \times 5\text{ m}$ , the error in computing the 2D response can be reduced



**Figure 4.1:** Schematic of the mesh and the conductivity model used to test the 2D forward modeling.

to within 1% of the 1D analytical value. The response was also computed for different cell sizes and the discretization errors are listed in Table 4.1. For cell sizes under 10 m $\times$ 10 m, the relative error averaged over all inphase and quadrature responses was found to be less than 1%. Since the measurement uncertainties in AEM systems are usually well above this level, these solution errors are sufficiently small for the purpose of inversion.

Now that the forward problem can be accurately solved, we can proceed to test the inversion algorithm.



**Figure 4.2:** Comparison of 2D response with 1D analytical solution. a) Computed responses from the synthetic model shown in Fig. 4.1. Circles are the 1D analytical responses and the crosses are the 2D modeled responses. b) Relative error of the 2D responses.

**Table 4.1:** Solution error of the 2.5D forward modeling for different cell sizes.

Cell size	Max. Inphase error(PPM)	Max. Quad error(PPM)	Mean rel. error(%)
$10 \times 10$	38.35(2.89%)	20.71(3.10%)	1.00
$10 \times 5$	12.28(0.92%)	8.88(1.00%)	0.37
$5 \times 5$	10.76(0.81%)	6.68(0.76%)	0.31
$2.5 \times 2.5$	3.75(0.28%)	1.61(0.18%)	0.11



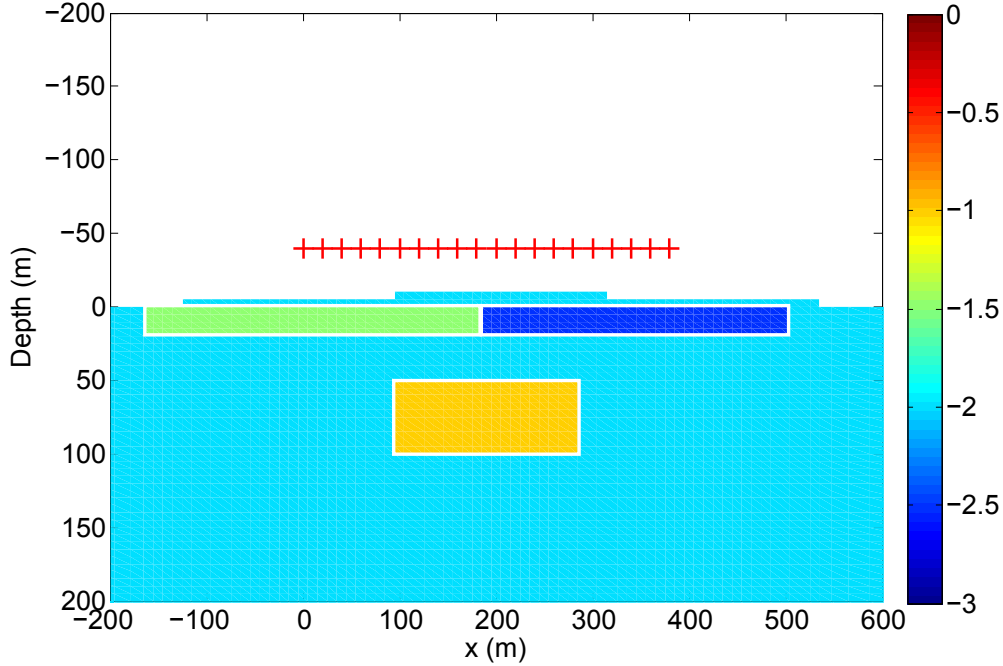
## 4.2 Inversion of synthetic data

We next test our inversion code on synthetic data. Before inversions are performed on actual geophysical datasets, it is always a good idea to test the method on known solutions first. Using synthetic models that replicate complex geological structures and topology can help in finding limitations and shortcomings of the algorithm. Here, we tested our algorithm on a 200 m wide and 50 m deep conductive 2D prism that is buried inside a 0.01 S/m half space (see Fig. 4.3). The prism has a conductivity of 0.1 S/m and is buried 30 m under a 20 m thick overburden that is resistive (0.003 S/m) on one half and conductive (0.03 S/m) on the other. Two small inclines were added to the topology to mimic variations in the height of earth’s surface. In order to avoid committing the “inversion crime”, the synthetic response was generated using a twice denser mesh with cell size of 5 m×5 m. The response 10 m away from the transmitters was computed for 20 horizontal coplanar sources and 5 logarithmically spaced frequencies. Each source-receiver pair is separated by 20 m and is placed 40 m above the zero depth level. Gaussian noise with a standard deviation of 5% of the response value plus a floor of  $10^{-10}$  A/m was used to contaminate the data before the inversion.

In the inversion, the data were assigned a standard deviation of  $8\% + 10^{-10}$  A/m and the mesh cell size of 10 m×10 m generates a forward system of about 8000 variables. The other inversion parameters were  $N_{k_y} = 6$ ,  $\alpha_s = 10^{-4}$ ,  $\alpha_x = \alpha_z = 4 \times 10^{-2}$ . The reference and initial model were set to a 0.01 S/m half-space. The inversion was run using the Parallel Computing Toolbox in Matlab with 8 parallel processes and a laptop equipped with a 2.30GHz Intel i7-2820QM CPU. For a  $\beta$  of 1000 and a *chifact* of 1, the target misfit of 100 was reached in 23 iterations and 20 minutes.

The results of inversion on the synthetic data are shown in Fig. 4.4. The data misfit showed that the optimization converged to a model that predicted the majority of the observed response data to within the noise level, except for several points near the first source which can be explained by the

smaller data per parameter density on the edges. The recovered model revealed three distinct regions of conductivity contrast in accordance with the synthetic model. The relative position of the regions are consistent with the true model, but the sizes and conductivities displayed some discrepancies. A slightly better recovery was observed for the resistive half of the overburden than the conductive half, while in total only about 60-70% of the area of the true overburden was reconstructed with the correct property value. This is actually a decent result considering that only about a third of the length of the overburden was covered by the measurements. This finding is evidence that the range of detection for conductors of similar conductivity does not extend beyond 100 m at the frequencies tested, or in other words the footprint of the AEM system setup is at most 100 m. As for the inverted target, its conductivity plateaued at a value that is 6% less than the true value. From the peak value, the conductivity dropped continuously until it reached the reference value about two cells away from the edge of the true target, which puts the uncertainty on the position of the edge at about 20 m. This uncertainty is the effect of regularizing with the mean-squared error (Eqn. 3.3) which produces models with smooth edges by heavily penalizing sharp jumps in the model. Although it is possible to reduce it with a different regularizing function (such as the total variation), the uncertainty does not pose major problems for mineral targeting purposes, as the dimension of many economically viable ore deposits is often an order of magnitude larger. With these results, we are now more confident that the algorithm is functioning like a proper inversion is supposed to. However, before we can label it as a working code, we need to test the inversion once more, but now on a 3D field dataset. Details and results of this test are described in the next section.

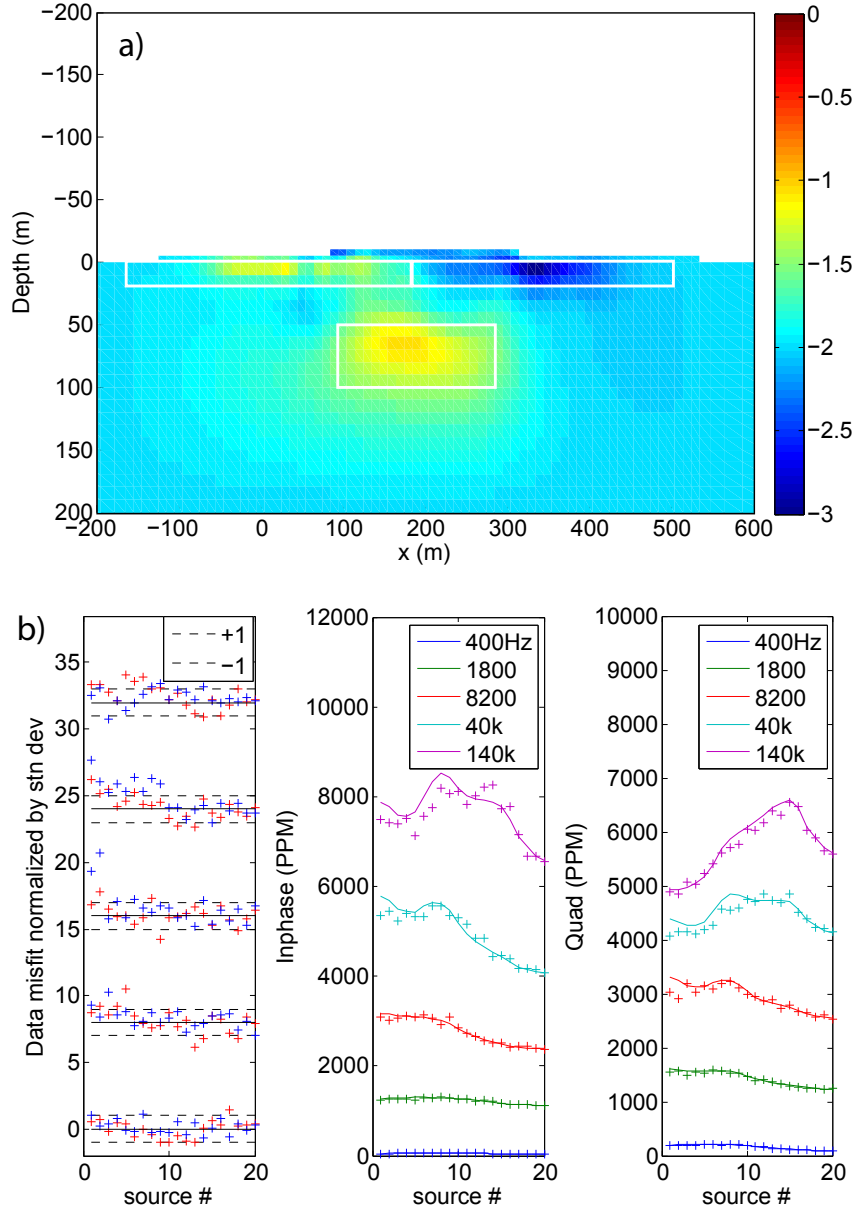


**Figure 4.3:** Synthetic model used in testing the 2D inversion algorithm. The model is 2D and it consists of a conductive prism and an overburden that is conductive on one end and resistive on the other. Responses were recorded 10 m away from the sources and 40 m above the ground. Locations of the sources are indicated by the red crosses.

### 4.3 Inversion of field data

To check the effectiveness of our algorithm in inverting field data, data from a Fugro RESOLVE survey were inverted. Since a unique solution does not exist, we can only verify the algorithm by comparing to results obtained with other well tested inversions. For this purpose, we compared our results to models from an established 1D inversion tool supplied by Fugro.

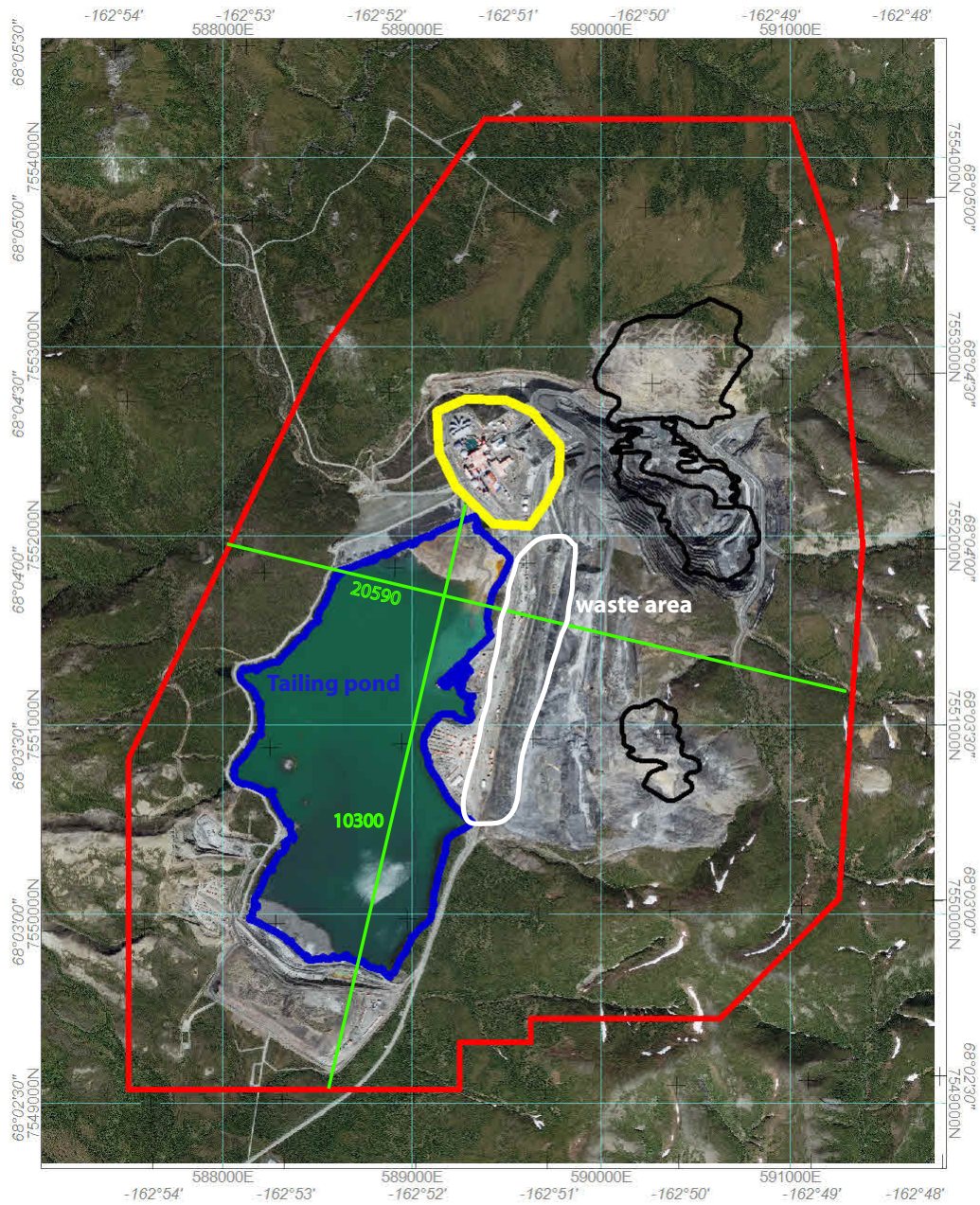
In the survey, a total of 684 line kilometers were flown over the Red Dog mine site in Alaska. The survey lines were spaced 50 m apart and they



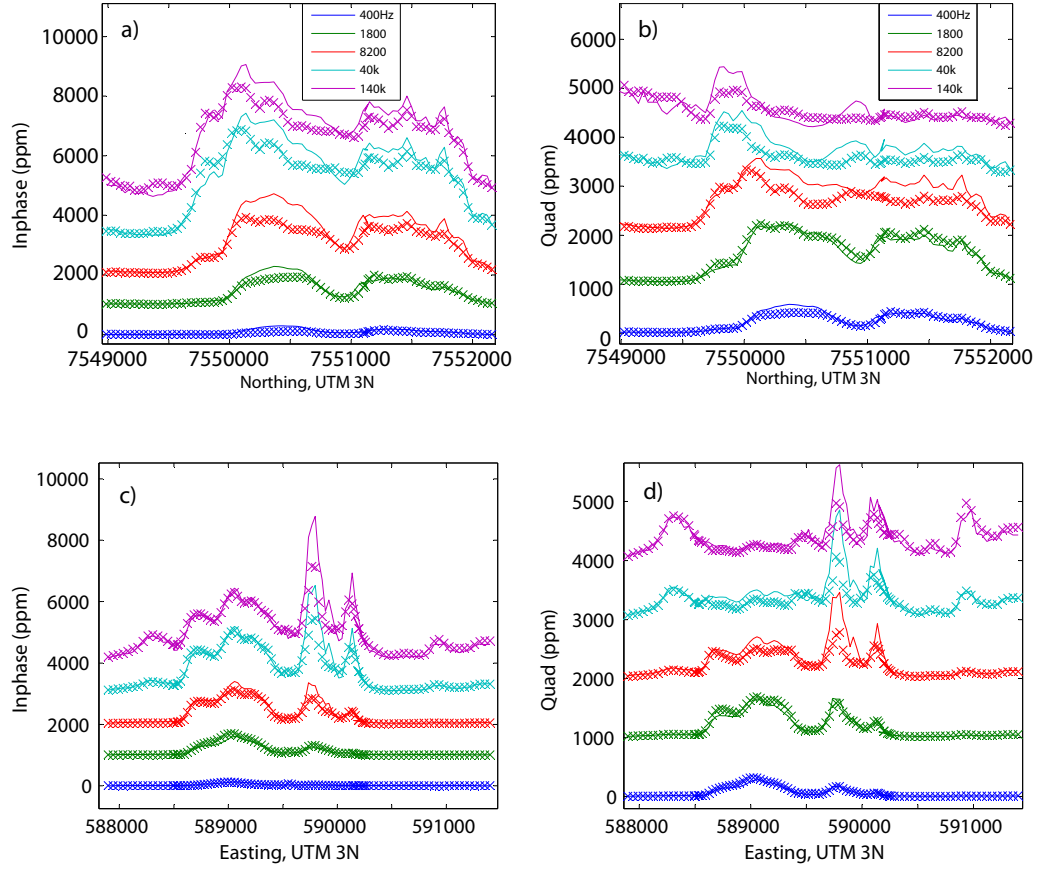
**Figure 4.4:** a) Log of the recovered 2D conductivity model for the synthetic data generated by the model shown in 4.3. b) Left: misfit of predicted data normalized by noise standard deviation. Middle and right: The response curves are offset by 1000 ppm for clarity. Crosses represent the synthetic inphase and quadrature responses and solid curves are the predicted data.

covered the site in two orthogonal directions:  $13^\circ$  and  $103^\circ$  azimuth. To map the conductivity contrasts of the survey area, the AEM system measured the magnetic response of the area in five frequencies at heights of 40-60 m above the ground and 7.9 m away from the sources. The objective of the survey was to use the conductivity detected in and around the mine’s tailing pond to identify any possible leakage of tailings from the waste area located to the east into the pond.

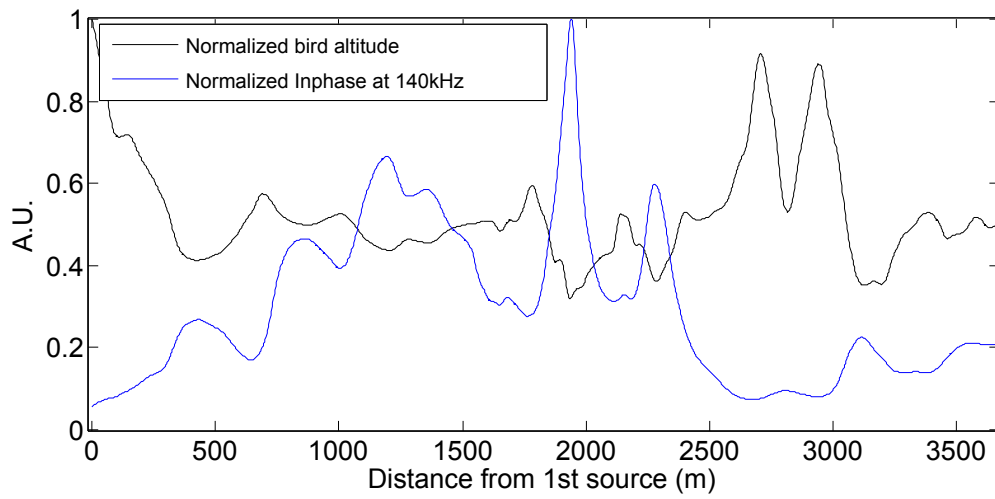
Here, instead of running a full inversion using every data point of each line, the responses were resampled at 40 m intervals. Equivalently, the data were decimated to 7% of the original sampling rate. The area of inversion had a 200 m depth extent and it was discretized with a mesh of  $8\text{ m} \times 8\text{ m}$  cells. The paddings added 300 m to both ends of the line and 300 m to the depth of the mesh. The data were assigned a noise standard deviation of  $10\% \times \mathbf{d}^{\text{obs}} + 10^{-9}$  A/m and a half-space of  $10^{-3}$  S/m was used as the reference model. By using a  $\beta$  determined from L-curve strategy and a convergence criteria of  $\text{tol\_step} = 0.02$ , convergence was usually reached within 50 iterations of L-BFGS with a history of 20 recent updates. Each L-BFGS iteration took about 3 mins on a cluster node outfitted with two 6-core Intel Xeon X5660 CPUs at 2.8 GHz and 64 GB of memory.



**Figure 4.5:** Satellite image of the Red Dog mine area in Alaska. Red line: outline of the survey area. White line: outline of the waste area. Blue line: outline of the tailing pond. Black lines: ore deposits. Green lines: survey lines inverted. Image from [21].

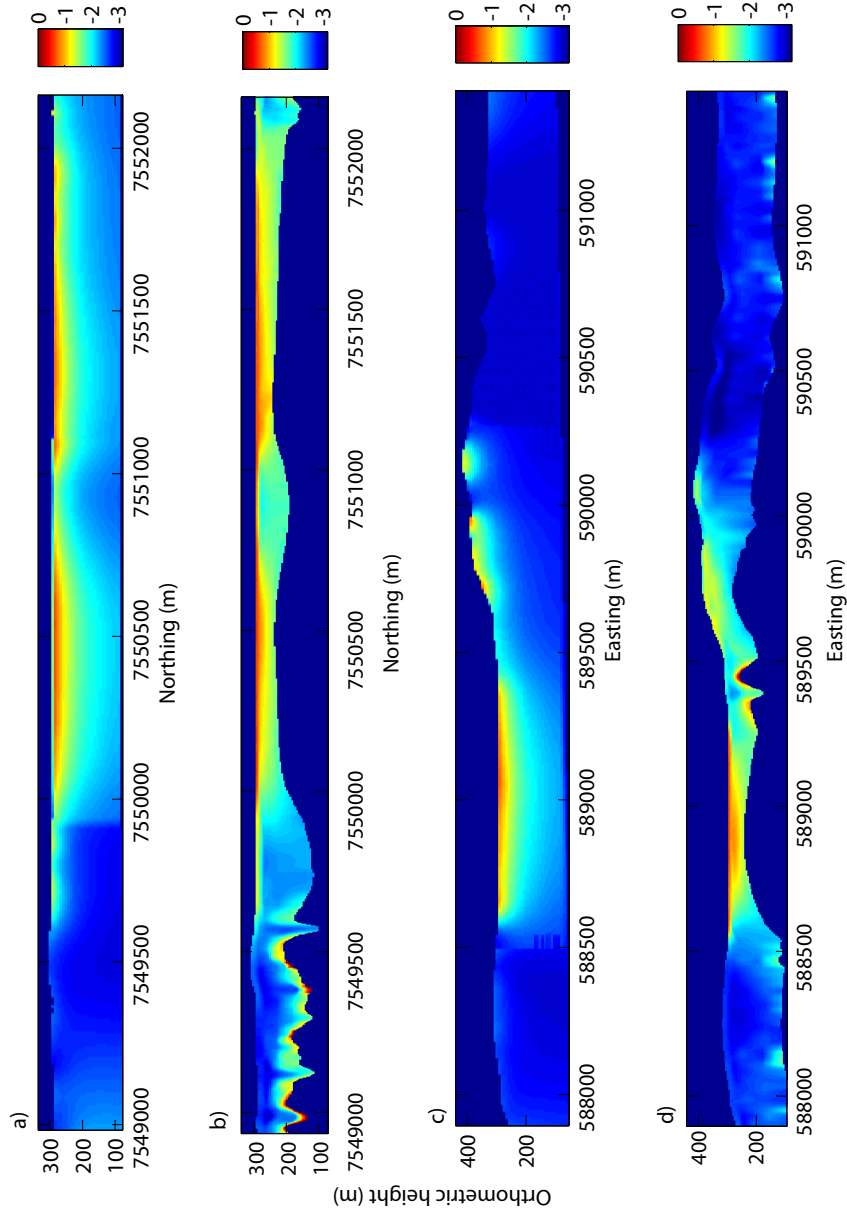


**Figure 4.6:** Predicted data misfit for the inversions on the lines shown in Fig. 4.5: a)-b) Line 10300, c)-d) Line 20590. The curves are offset by 1000 ppm for clarity. Solid lines: predicted response. Crosses: observed data. Observed data are from [21].



**Figure 4.7:** Normalized bird altitude and a measured response curve. The two observables are strongly correlated.





**Figure 4.8:** 2D (a,c)) and 1D (b,d)) inverted log conductivity models for the two survey lines highlighted in Fig. 4.5. a)-b) Line 10300, c)-d) Line 20590.

In the test, we inverted several 3 km long survey lines from both orthogonal directions. As seen in Fig. 4.6, the observed responses are highly varying between measurement points. A quick comparison of the length scale of these changes and that of fluctuations observed in the height of the bird (see Fig. 4.7) suggests that the variations are not due to instrument noise, but are rather caused by the changing bird altitude. As a consequence, one would expect the predicted data to exhibit the same fluctuations as in the observed data if the forward modeling takes the varying altitude into account. This was indeed the case: the predicted data showed a close fit to a large portion of the observations and they reproduced the fluctuations caused by the flight altitude.

Inversions for the two lines highlighted in Fig. 4.5 are shown in Fig. 4.8. When compared to 1D inversions, our inverted models displayed the same ability to resolve areas of high conductivity contrasts. The conductive regions were found to be in the pond and on top of the waste dump area, which respectively correspond to the water in the pond and the conductive waste pile-up on the dump site. Features in the 1D models that are likely contributed by conductive geological structures were all recovered in the 2D models. However, most of these structures appeared smoother in our 2D inversions. We relate this to our choice of large smoothness parameters  $\alpha_x$  and  $\alpha_z$ , which have the effect of penalizing sharp features in the solution. Despite the size of the anomalies showed some discrepancies, their conductivity values found in the 2D models were in good agreement with the 1D results as well as their location and depth extent.

As for locating possible channels of tailing drainage, no clear evidence of their presence was found in any of the lines inverted. But like every good inversion work-flow will suggest, inversion should always be repeated multiple times such that decisions are made based on a collection of models rather than on a single model. Another run of inversion with better adjusted regularization parameters and smaller cell sizes is therefore necessary to obtain

a more conclusive answer.

To summarize the test results, our implementation of the 2.5D EM forward modeling and inversion scheme were successful when verified against more established 1D counterparts. The forward modeling was able to compute the magnetic response to a level of accuracy ( $\pm 1\%$ ) that is sufficiently high for inversion purposes. In addition, there is strong evidence that the inversion algorithm is correctly resolving conductive anomalies at least as accurately as a standard 1D inversion is doing.

# Chapter 5

## Conclusion and future work

In this project, we developed an inversion algorithm to recover a 2D conductivity model of the earth from frequency domain AEM data generated by horizontally coplanar vertical dipoles. We demonstrated that our implementation of the 2.5D EM problem can accurately predict the magnetic response of a three layer model and the inversion scheme can successfully and reliably be applied to field data to correctly locate and identify conductive anomalies.

However, the computation speed still poses a big problem and improvements are necessary to give our algorithm a more definite edge over other 1D or 3D inversion tools that are now available. This could be done by using adaptive meshes [10] which can greatly reduce the size of the linear system. The technique saves computing time by starting out with a coarse grid, which is then successively refined for only those regions in the grid that are rich in details. Another solution to the problem is by speeding up the matrix decomposition in the direct solve step. This can be achieved by using parallelized sparse direct solver algorithms such as MUMPS and SuperLU.

Moreover, whether the inversion is effective in correctly resolving 3D structures is still a question to be answered. As one might have quickly noticed by inspecting the aerial map of the mine site in Chapter 4, the conductive structures under the two inverted lines are rather 2D in nature.

Therefore in the future, in order to make more correct interpretations of higher dimensional geologies, our algorithm should also be tested on data produced by strongly 3D structures, for example by a conductive block with a strike extent comparable to the size of the AEM footprint.

For future work, a direct continuation of this project would be to implement the same 2.5D formulation but for time domain EM data. For this, the same inversion scheme shown earlier can be applied once the forward problem is solved. Solving time domain problems is however a little more involved as it requires time stepping, which cannot be easily parallelized. Alternatively, the time domain problem can be reformulated and solved in the frequency domain by applying a Fourier transform. But this approach is only advantageous if the number of frequencies required for an accurate transformation can be kept small.

# Bibliography

- [1] A. Abubakar, T. M. Habashy, V. L. Druskin, L. Knizhnerman, and D. Alumbaugh. 2.5D forward and inverse modeling for interpreting low-frequency electromagnetic measurements. *Geophysics*, 73: F165–F177, 2008. → pages 7
- [2] P. H. Calamai and J. J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39:93–116, 1987. → pages 39
- [3] L. H. Cox, G. A. Wilson, and M. S. Zhdanov. 3D inversion of airborne electromagnetic data using a moving footprint. *Exploration Geophysics*, 41:250–259, 2010. → pages 5
- [4] A. Dey and S. H. Ward. Inductive sounding of a layered earth with a horizontal magnetic dipole. *Geophysics*, 35:660703, 1970. → pages 5, 45
- [5] R. G. Ellis. Inversion of airborne electromagnetic data. *68th SEG meeting, Expanded Abstracts*, pages 2016–2019, 1998. → pages 5
- [6] M. Everett and R. Edwards. Transient marine electromagnetics: the 2.5-D forward problem. *Geophys. J. Int.*, 113:545–561, 1992. → pages 6
- [7] E. Haber. *Numerical Strategies for the solution of inverse problems*. PhD thesis, UBC, 1997. → pages 41
- [8] E. Haber, U. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16: 1263–1280, 2000. → pages 37

- [9] E. Haber, U. Ascher, and D. Oldenburg. Inversion of 3D electromagnetic data in frequency and time using an inexact all-at-once approach. *Geophysics*, 69:1216–1228, 2004. → pages 5
- [10] E. Haber, S. Heldmann, and U. Ascher. Adaptive finite volume methods for distributed non-smooth parameter identification. *Inverse Problems*, 23:1659–1676, 2007. → pages 22, 59
- [11] J. D. Jackson. *Classical Electrodynamics*. Wiley, 3 edition, 1999. → pages 14
- [12] C. T. Kelley. *Iterative Methods for Optimization*. Siam, 1999. → pages 37
- [13] K. Key and J. Owall. A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling. *Geophys. J. Int.*, 186:137–154, 2011. → pages 7
- [14] A. Ley-Cooper, J. Macnae, and A. Viezzoli. Breaks in lithology: Interpretation problems when handling 2D structures with a 1d approximation. *Geophysics*, 75:WA179–WA188, 2010. → pages 5
- [15] Y. Mitsuhashi, T. Uchida, and H. Amano. 2.5-D inversion of frequency-domain electromagnetic data generated by a grounded-wire source. *Geophysics*, 67:1753–1768, 2002. → pages 7
- [16] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999. → pages 36
- [17] A. Pidlisecky and R. Knight. FW2\_5D: A MATLAB 2.5-D electrical resistivity modeling code. *Computers & Geosciences*, 34:1645, 2008. → pages 14
- [18] J. Ryu, H. F. Morrison, and S. H. Ward. Electromagnetic fields about a loop source of current. *Geophysics*, 35:862–896, 1970. → pages 5
- [19] B. R. Spies and F. C. Frischknecht. *Electromagnetic sounding: in Nabighian, M. N., editor, Electromagnetic Methods in Applied Geophysics–Applications*. Soc. of Expl. Geophys., 1 edition, 1991. → pages 15

- [20] C. Stoyer and R. J. Greenfield. Numerical solutions of the response of a two-dimensional earth to an oscillating magnetic dipole source. *Geophysics*, 41:519–530, 1976. → pages 6
- [21] Teck Resources Ltd. Technical report. → pages viii, 53, 54
- [22] M. J. Unsworth, B. J. Travis, and A. D. Chave. Electromagnetic induction by a finite electric dipole source over a 2-d earth. *Geophysics*, 58:198–214, 1993. → pages 7
- [23] S. H. Ward and G. W. Hohmann. *Electromagnetic theory for geophysical applications: in Nabighian, M. N., editor, Electromagnetic Methods in Applied Geophysics-Applications*. Soc. Expl. Geophys., 1988. → pages 5, 10, 45
- [24] G. A. Wilson, A. P. Raiche, and F. Sugeng. 2.5D inversion of airborne electromagnetic data. *Exploration Geophysics*, 37:363–371, 2006. → pages 7
- [25] X. Xie, J. Macnae, and J. Reid. The limitations of 1-D aem inversion for 2-D overburden structures. *68th SEG meeting, Expanded Abstracts*, pages 760–763, 1998. → pages 5
- [26] S. Xu, B. Duan, and D. Zhang. Selection of the wavenumbers  $k$  using an optimization method for the inverse fourier transform in 2.5D electrical modelling. *Geophysical Prospecting*, 48:789–796, 2000. → pages 14
- [27] D. Yang and D. W. Oldenburg. Three-dimensional inversion of airborne time-domain electromagnetic data with applications to a porphyry deposit. *Geophysics*, 77:B23–B34, 2012. → pages 5