Technical Report

TR-2005-012

Inversion of Time domain 3D Electromagnetic data

by

Eldad Haber, Douglas W. Oldenburg, R. Shekhtman

Mathematics and Computer Science

EMORY UNIVERSITY

Inversion of Time domain 3D Electromagnetic data

Eldad Haber* Douglas W. Oldenburg R. Shekhtman†

September 12, 2005

Abstract

Inversion of 3D time-domain electromagnetic data is a challenging problem due to its size and nonlinearity. In this paper we present a general formulation for inverting time domain electromagnetic data. To overcome the difficulties we combine and develop many computational tools including: Quasi-Newton methods and preconditioners, artificial time stepping and source terms. This combination allows us to deal with realistic geophysical problems.

keywords: Electromagnetic, Time-Domain, Inversion

1 Introduction

In this paper we develop an inversion methodology for large scale parameter identification problems which evolve from 3D time domain electromagnetic simulations. This type of problem is of major interest in geophysics, medical imaging and non-destructive testing; see for example [29, 27, 10, 6, 30, 16, 7] and references therein. The forward model consists of Maxwell's equations in time where the permeability is fixed but electrical conductivity can be highly discontinuous. The parameter regimes considered give rise to highly stiff problems in the time domain. The goal of the inversion is to recover the conductivity given measurements of the electric and/or magnetic fields.

There are many practical challenges to solving the inverse problem. First, a fast, accurate and reliable algorithm for 3D forward modeling is required. Second, the sensitivities for such problems are too numerous to be formed or stored in a reasonable amount of time and space. Finally, finding the minimum of the objective function obtained by matching the data and incorporating a priori information on the conductivity field can be difficult due to the nonlinearity of the problem.

Further difficulties are often confronted when trying to invert a particular data-set. For example, in geophysical surveys such as large-loop UTEM surveys or CSAMT surveys, the

*Department of Mathematics and Computer Science, Emory University, Atlanta, GA.

†UBC-Geophysical Inversion Facility, Department of Earth and Ocean Sciences, University of British Columbia, Vancouver, B.C., V6T 1Z4, Canada.

transmitter may be some distance away from the area of interest. Discretizing a spatial volume that includes the transmitter and receivers can result in an excessively large problem. Also, most transmitter waveforms have an on-time and off-time and the ideal scenario is that electromagnetic fields have completely decayed away before the next waveform begins. In conductive environments however the decay time might extend for a couple of periods of the input waveform. Modelling the field data then requires an extended source that consists of a few repetitions of the waveform. This increases the time for forward modelling, or effectively the size of our space-time inverse problem.

A main impediment to inverting time domain electromagnetic data is the size of the problem and the number of computations to be carried out. To be more specific, assume that the forward problem (in continuous space) is written in the form

$$\mathcal{A}(m)u - b = 0 \tag{1.1}$$

where $\mathcal{A}(m)$ is a version of Maxwell's equations (including boundary conditions), $m = \log(\sigma)$ is the log conductivity, u stands for the fields and b represents sources and boundary values. We assume for simplicity of exposition that $\mathcal{A}$ is invertible for any relevant m, that is, there is a unique solution to the forward problem.

In the inverse problem we measure some function of the fields and desire to recover the model m. Let us write the measured data as

$$d^{\text{obs}} = \mathcal{Q}u + \epsilon \tag{1.2}$$

where $\mathcal{Q}$ is a measurement operator which projects the fields (or their derivatives or integrals) onto the measurement locations in 3D space and time, and ϵ is the measurement noise. The data are finite in number and contaminated with noise, and therefore there is no unique solution. To obtain a single model which depends stably on the data we incorporate *a-priori* information and formulate the inverse problem (in continuous space) as a constrained optimization problem of the form

$$\begin{aligned} \min_{m,u} \quad & \frac{1}{2}\|\mathcal{Q}u - d^{\text{obs}}\|^2 + \beta\mathcal{R}(m) \\ \text{subject to} \quad & \mathcal{A}(m)u - b = 0. \end{aligned} \tag{1.3}$$

Here, $\beta > 0$ is the regularization parameter, and $\mathcal{R}(\cdot)$ is a regularization operator reflecting our *a-priori* information. Typically, we know that m is a piecewise smooth function over the spatial domain Ω in 3D, so we assume that $\mathcal{R}(\cdot)$ involves some norm of ∇m over Ω, e.g., weighted L_2 or L_1 or a Huber combination [24, 12]. This type of regularization can be written as

$$\mathcal{R}(m) = \int_\Omega \rho(|\nabla m|)\, d\mathbf{x} + \gamma\frac{1}{2}\int_{\partial\Omega} (m - m_{ref})^2 d\mathbf{x} \tag{1.4}$$

where ρ is given by the Huber function and m_0 is some background reference model and γ is a user-specified constant that adjusts the relative influence of the two terms in the

regularization functional. The choice of this functional ensures that the weak form has Dirichlet boundary conditions and that in regions far away from where we have data, the model converges to a known background. This choice also guaranties that the Hessian of the regularizer is invertible.

Next, the problem (1.3) is discretized using some finite volume method over a finite grid representing the domain in space and time. This yields the finite dimensional optimization problem

$$\begin{aligned} \min_{m,u} \quad & \frac{1}{2}\|Qu - d^{\text{obs}}\|^2 + \beta R(m) \\ \text{subject to} \quad & A(m)u - b = 0, \end{aligned} \tag{1.5}$$

where u, m and q are grid functions ordered as vectors and correspond to their continuous counterparts above, and Q and A are large, sparse matrices. The matrix A depends on m and is nonsingular. The discrete regularization function has the form

$$R(m) = e^\top \rho(|\nabla_h m|) + \gamma||W(m - m_{ref}||^2 \tag{1.6}$$

where ∇_h is the discrete gradient operator and e is a vector of ones.

In previous work [21] we have used an inexact, all-at-once methodology [5, 17, 3] to solve (1.5), solving the forward problem and the inverse problem simultaneously in one iterative process. While this approach can be highly efficient, it requires storage of all time-space history of the fields.

In this work we aim to deal with very large scale problems and multiple sources. For such problems one cannot store all the fields on computer hardware that is currently available to us, and we therefore turn to other avenues for the solution of the problem. A simple way to reduce storage is to use a reduced space method, which eliminates Maxwell's equations, and solves the unconstrained optimization problem

$$\min_m \ \frac{1}{2}\|QA(m)^{-1}b - d^{\text{obs}}\|^2 + \beta R(m). \tag{1.7}$$

This leads to the Euler-Lagrange system

$$g(m) = J(m)^\top (QA(m)^{-1}b - d^{\text{obs}}) + \beta R_m(m) = 0 \tag{1.8}$$

where

$$\begin{aligned} J(m) &= -QA(m)^{-1}G(m) \\ G(m) &= \frac{\partial [A(m)u]}{\partial m}. \end{aligned}$$

and $J(m)$ is the sensitivity matrix (see [20] for derivation) and $R_m(m) = \frac{\partial R}{\partial m}$.

We search for an m that solves (1.8). The evaluation of $g(m)$ involves a number of computations. To calculate $R_m(m)$ we use the chain rule

$$R_m(m) = \nabla_h^\top \rho'(|\nabla_h m|)\nabla_h m + \gamma W^\top W(m - m_{ref}).$$

Also, the forward problem must be solved to generate predicted data, and calculating the action of $J(m)^\top$ on a vector requires the solution of the adjoint problem. If a gradient descent type method is used to solve (1.8) then each iteration requires the solution of a forward and adjoint problem which means solving Maxwell's equations forward and backward in time. In addition a line search involving forward modeling must be carried out.

Newton-type methods are generally far superior to descent methods when solving (1.8). If a Gauss-Newton method is used, then at each iteration we solve the linear (but dense) system

$$(J(m)^\top J(m) + \beta R_{mm})s = -g(m) \tag{1.9}$$

where s is a model perturbation. This system is never formed explicitly but is solved using conjugate gradient (CG) methods. At each CG iteration $J(m)$ and $J(m)^\top$ are applied to a vector and hence a forward and adjoint problem must be solved. Even with reasonably good preconditioners, a few tens of CG iterations are likely required to get a good estimate of s. This results in many forward modellings and hence the cost of a Newton-CG algorithm is also too high. We therefore turn to Quasi-Newton techniques.

In a quasi-Newton method we solve a system

$$B(m)s = -g(m) \tag{1.10}$$

where B is an approximation to the Hessian. One of the most successful methods for unconstrained optimization is the limited memory BFGS (L-BFGS) method [26]. In a BFGS method either B or it's inverse, B^{-1}, is built up through successive iterations. The main computations at each iteration are construction of the gradient and performing a line search. The constituent vectors for constructing B need to be stored. The limited memory version of BFGS addresses this issue. While the method is highly attractive for inverse problems, naive implementation of L-BFGS usually fails (see [15] for details).

In this paper we propose a practical strategy that combines the strengths of Gauss-Newton and quasi-Newton approaches. Firstly, we alter the usual Tikhonov methodology to use an iterated Tikhonov approach that partially overcomes difficulties in solving the optimization problem when the regularization parameter is small. We then explore a particular implementation of L-BFGS and use the constructed approximate inverse Hessian as a preconditioner for the Gauss-Newton iteration. This substantially reduces the number of CG iterations in solving (1.9). We also present strategies for dealing with some of the practical issues mentioned earlier. In particular we introduce a source correction procedure that allows us to reduce the spatial volume of the region to be inverted and ameliorates discretization errors. We also introduce a technique to work with problems that have an extended time source.

The paper is divided as follows. In Section 2 we describe the solution of the forward problem. In Section 3 we discuss how to modify the right hand side of the forward problem such that the size of the discretized domain is smaller. In Section 4 we discuss the overall approach for the solution of the problem and motivate the use of Quasi-Newton methods. In Section 5 we discuss efficient implementation of Quasi-Newton methods. In Section 6 we

discuss how to deal with extended waveforms source-signals. We give numerous examples in Section 7 and the paper is summarized in Section 8.

2 Solution of the forward problem

In this section we present our forward problem, Maxwell's equations, and discuss solution procedures suitable for the parameter regimes of interest. Most, but not all of the present development follows our previous work [18, 16, 21].

2.1 Reformulation and discretization

The time-dependent Maxwell equations can be written as

$$\nabla \times \mathbf{E} + \mu \frac{\partial \mathbf{H}}{\partial t} = 0, \tag{2.11a}$$

$$\nabla \times \mathbf{H} - \sigma \mathbf{E} - \epsilon \frac{\partial \mathbf{E}}{\partial t} = \mathbf{s}_r(t) \tag{2.11b}$$

over a domain $\Omega \times [0, t_f]$, where $\mathbf{E}$ and $\mathbf{H}$ are the electric and magnetic fields, μ is the permeability, σ is the conductivity, ϵ is the permittivity and s_r is a source. The equations are given with boundary and initial conditions:

$$\mathbf{n} \times \mathbf{H} = 0 \tag{2.11c}$$

$$\mathbf{H}(0, \mathbf{x}) = \mathbf{H}_0 \tag{2.11d}$$

$$\mathbf{E}(0, \mathbf{x}) = 0 \tag{2.11e}$$

although other boundary and initial conditions could be used.

Since the equations are very stiff [4, 21] we turn to implicit methods, and use a BDF (Backward Difference Formula) type method. For the work here we choose the simplest, lowest order member of both these families of stiff integrators, namely, the backward Euler method. This allows us to simplify the presentation, however we have also implemented a BDF(2) integrator and the reader is referred to [19, 21, 23].

Semi-discretizing (2.11), (2.11c) over a time step $[t_n, t_{n+1}]$ yields the equations

$$\nabla \times \mathbf{E}^{n+1} + \alpha_n \mu \mathbf{H}^{n+1} = \alpha_n \mathbf{H}^n \quad \text{in } \Omega \tag{2.12a}$$

$$\nabla \times \mathbf{H}^{n+1} - (\sigma + \alpha_n \epsilon)\mathbf{E}^{n+1} = \mathbf{s}_r^{n+1} - \alpha_n \epsilon \mathbf{E}^n \quad \text{in } \Omega \tag{2.12b}$$

$$\mathbf{n} \times \mathbf{H}^{n+1} = 0 \quad \text{on } \partial\Omega. \tag{2.12c}$$

where $\alpha_n = (t_{n+1} - t_n)^{-1}$. The superscripts in (2.12) denote the time level, with solution quantities at $n+1$ being unknown while those at n are known.

Rather than working with the fields $\mathbf{E}^{n+1}$ and $\mathbf{H}^{n+1}$ we first eliminate $\mathbf{H}^{n+1}$ from (2.12)

$$\mathbf{H}^{n+1} = \mathbf{H}^n - (\alpha\mu)^{-1} \nabla \times \mathbf{E}^{n+1}.$$

Substituting, we obtain an equation for $\mathbf{E}^{n+1}$

$$\nabla \times \mu^{-1}\nabla \times \mathbf{E}^{n+1} + \alpha\sigma\mathbf{E}^{n+1} + \alpha^2\epsilon\mathbf{E}^{n+1} = \alpha(\nabla \times \mathbf{H}^n + \alpha\epsilon\mathbf{E}^n - \mathbf{s}). \tag{2.13}$$

As discussed in [18, 16] the discretization of the system (2.13) is difficult to solve using iterative methods. Introducing the potentials $\mathbf{A}, \phi$ to write $\mathbf{E} = \mathbf{A} + \nabla\phi$ we get

$$\begin{aligned}(\nabla \times \mu^{-1}\nabla \times -\nabla\mu^{-1}\nabla\cdot)\mathbf{A} + \alpha(\sigma + \alpha\epsilon)(\mathbf{A} + \nabla\phi)^{n+1} = \\ \alpha(\nabla \times \mathbf{H}^n + \alpha\epsilon\mathbf{E}^n - \mathbf{s})\end{aligned} \tag{2.14}$$

$$\nabla \cdot (\sigma + \alpha\epsilon)(\mathbf{A} + \nabla\phi)^{n+1} = \alpha\nabla \cdot \epsilon\mathbf{E}^n - \nabla \cdot \mathbf{s} \tag{2.15}$$

For ease of notation we introduce $\hat{\sigma} = \sigma + \alpha\epsilon$. The continuous equations are discretized in space using a finite volume technique on a staggered grid. See (see details in [18, 16, 21]). Variables $\mathbf{A}$ and $\mathbf{E}$ are defined at the center of cell faces, $\mathbf{H}$ is defined on the edges, and ϕ in the cell centers. Upon discretization we obtain

$$\begin{pmatrix} L_\mu + \alpha M_{\hat{\sigma}} & \alpha M_{\hat{\sigma}} G \\ D M_\sigma & D M_{\hat{\sigma}} G \end{pmatrix} \begin{pmatrix} \mathbf{A} \\ \phi \end{pmatrix} = \begin{pmatrix} \alpha(C_e\mathbf{H}^n + \alpha\epsilon\mathbf{E}^n - \mathbf{s}) \\ \epsilon\alpha D\mathbf{E}^n - D\mathbf{s} \end{pmatrix} \tag{2.16}$$

where

$$L_\mu = C_e M_\mu C_f - G M_\mu^c D$$

and the matrices C_e, C_f, G and D are respectively the discretizations of the curl on the edges and faces, the gradient and the divergence. The matrix $M_{\hat{\sigma}}$ is a discretization of $\sigma + \alpha\epsilon$ on the cell faces. It is composed of harmonic averages of $\hat{\sigma}$ for the two cells in contact at a face. The matrix M_μ consists of arithmetic averages of the permeability for cells adjacent to an edge.

This linear system can be solved using standard iterative methods [28] and effective scaleable preconditioners can be designed for it [18, 1].

2.2 Summary - the forward problem

In many applications we are concerned with multiple sources and with multiple time steps. For solving the inverse problem it is useful to formulate the forward problem as a single equation (see for example Section 1).

We note that it is possible to rewrite the system as a lower block bidiagonal system for $\mathbf{A}, \phi$ and $\mathbf{H}$ of the form

$$A(m)u = \begin{pmatrix} A_1(m) & & & & & \\ B_2 & A_2(m) & & & & \\ & B_3 & A_3(m) & & & \\ & & \ddots & \ddots & & \\ & & & B_s & A_s(m) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \\ u_s \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ \\ q_s \end{pmatrix} \tag{2.17}$$

where

$$u_n = \begin{pmatrix} \mathbf{A}_n \\ \phi_n \\ \mathbf{H}_n \end{pmatrix}, \qquad A_n(m) = \begin{pmatrix} L_\mu + \alpha_n M_{\hat{\sigma}} & \alpha_n M_{\hat{\sigma}} \nabla_h & 0 \\ \nabla_h \cdot M_{\hat{\sigma}} & \nabla_h \cdot M_{\hat{\sigma}} \nabla_h & 0 \\ \alpha_n^{-1} M_\mu^{-1} \nabla_h \times & 0 & I \end{pmatrix}$$

$$q_n = \begin{pmatrix} -\alpha_n {\mathbf{s}_r}^n \\ -\nabla_h \cdot \mathbf{s}_r^n \\ 0 \end{pmatrix}, \quad B_n = \begin{pmatrix} -\epsilon\alpha_n^2 & -\epsilon\alpha_n^2 \nabla_h & -\alpha_n \nabla_h^T \times \\ -\nabla_h \cdot \epsilon\alpha_n & -\nabla_h \cdot \epsilon\alpha_n \nabla_h & 0 \\ 0 & 0 & -I \end{pmatrix}.$$

In the case of multiple sources we obtain a block diagonal system where each block has the same structure as (2.17). Note that only the diagonal blocks in (2.17) depend on the conductivity. Also, once we have an efficient solver for one block $A_k(m)$, solving the forward problem (2.17) is straightforward (with the cost of solution increasing by a factor of s compared to the cost of solving for one block).

3 Corrective Sources

Our boundary condition requires the domain to be large enough such that the magnetic fields are dramatically reduced. This may require that a very large domain be modeled. This is especially true if the sources are far from the receivers, as they are in many EM surveys.

To make the problem tractable we introduce a correction procedure which has many elements of a primary/secondary field formulation. The methodology can be used to reduce the size of the problem and also to provide a first order correction for discretization errors. We first write the steps in a generic manner and then show how this is implemented in the time domain equations.

Let A denote a general operator which depends upon m, our physical property of interest (i.e. electrical conductivity), $\tilde{u}$ is a field, and $\tilde{q}$ is a source. $m, \tilde{u}, \tilde{q}$ are all functions with a relationship

$$A(m)\tilde{u} = \tilde{q} \tag{3.18}$$

The above equations are also provided with boundary conditions. Analytic, or quasi-analytic, solutions are available only for simple situations, for example, finding the fields in a half-space due to an electric or magnetic source. For realistically complicated earth models we obtain a numerical solution by solving a discrete system

$$A_h(m)u_h = q_h \tag{3.19}$$

where A_h is a matrix that results from the discretization of A, u_h is a vector of (discrete) field values, and q_h is a vector that contains the discretized source and boundary conditions.

Let m_0 be a model for which we can generate an exact solution to equation (3.18) and let the discrete field values be $\tilde{u}(m_0)$. It is unlikely that $\tilde{u}(m_0)$ is an exact solution for the discrete equations (3.19). The residual is

$$A_h(m_0)\tilde{u}(m_0) - q_h = s. \tag{3.20}$$

This discrepancy can be regarded as a corrective source and added to the discrete equations. We therefore solve

$$A_h(m)u_h = q_h + s. \tag{3.21}$$

It is straightforward to show that when $m = m_0$ the solution of the discrete equations $u_h = \tilde{u}(m_0)$. So the numerical solution is equal to the "exact" solution for our reference problem. The above correction can be used for the following two modeling problems.

1. Discretization errors: The corrective source is a set of currents that compensate for the differences between the discrete and continuous models for a specific conductivity. This is particularly useful for dealing with the current source. In geophysical field applications the source current is carried by a small wire, whereas in the discrete formulations the current is assumed to be distributed through the face of a cell. If a receiver is close to a transmitter, the difference in fields can be significant.

2. Reducing the size of the problem. In CSAMT problems, or large loop UTEM surveys, the data are acquired in a region that is significantly away from the source. Our data area, and region of interest, is much smaller than the volume that contains the transmitter and receivers. If the discretized volume is outside the region of the transmitter, the corrective- source will primarily be currents on the boundary of the volume.

The corrective sources provide the needed compensation for a specific conductivity. We have no mathematical procedure for evaluating how well the correction terms work when the conductivity changes significantly from m_0. However, in the problem of reducing the volume, it is expected that if a reasonable estimate of the large-scale background conductivity is known, and if the secondary fields for the structure inside our reduced volume are small on the outside of the reduced volume, then our procedure should work well. Numerical experiments substantiate this.

We now apply the above methodology to our time domain problem. Assume we have calculated the flux $\mathbf{J}_0^n$ and the magnetic field $\mathbf{H}_0^n \;\; n = 1..N$, that correspond to the conductivity σ_0 and the source $\mathbf{s}$. This can be done by analytic formulas [32] or by using 1D codes that allow for the calculation of highly accurate fields given a simplified earth model. The corresponding (discrete) electric field is

$$\mathbf{E}_0^n = M_{\sigma_0}^{-1}\mathbf{J}_0^n$$

Our basic equation is (2.13) and thus the source for time t_{n+1} is

$$\mathbf{s}_0 = C_e M_\mu C_f \mathbf{E}_0^{n+1} + \alpha M_{\sigma_0}\mathbf{E}_0^{n+1} + \alpha^2\epsilon\mathbf{E}_0^{n+1} - \alpha(C_e\mathbf{H}_0^n + \alpha\epsilon\mathbf{E}_0^n - \mathbf{s}) \tag{3.22}$$

Using the potentials we therefore have

$$\begin{pmatrix} L_\mu + \alpha M_{\hat{\sigma}} & \alpha M_{\hat{\sigma}} G \\ D M_{\hat{\sigma}} & D M_{\hat{\sigma}} G \end{pmatrix} \begin{pmatrix} \mathbf{A} \\ \phi \end{pmatrix} = \begin{pmatrix} \mathbf{s}_0 + \alpha(C_e\mathbf{H}^n + \alpha\epsilon\mathbf{E}^n - \mathbf{s}) \\ D\mathbf{s}_0 + \alpha D\epsilon\mathbf{E}^n - D\mathbf{s} \end{pmatrix} \tag{3.23}$$

Note that upon using $\mathbf{s}_0$ in (2.13), and the equivalent $\mathbf{A} - \phi$ formula (3.23) with the background conductivity σ_0, we obtain the fields $\mathbf{E}_0$ and $\mathbf{H}_0$. The right hand side of the equation is zero (up to discretization errors) where there are no electric or magnetic sources. However, the right hand side is different from zero on the boundary of our domain. This can be thought of an artificial source which emulates the source influence on a finite grid.

Numerical experiments show that the formulation (3.23) allow us to dramatically reduce the volume under consideration without sacrificing accuracy.

4 Inversion Framework

4.1 Inversion algorithms

As discussed in the introduction we aim to solve the minimization problem (1.7). There is one further complication when solving the optimization problem because, in general, the regularization parameter is unknown a-priori. Therefore, we need to solve the optimization problem a few times for different regularization parameters. The "optimal" regularization parameter has to be chosen using an appropriate criteria such as discrepancy principle [27] or GCV [22, 31]. A general sketch of the algorithm that achieves this goal is as follows

Algorithm 1 Tikhonov Inversion algorithm:
$[m, \beta] \leftarrow \text{INV}(m_0, \beta_0)$;

```
while true do
  given β approximately solve (1.7) for m_β with starting guess m_0
  if m_β fulfills chosen stopping criteria then
    break;
  end if
  update β, set m_0 ← m_β
end while
```

Algorithm 1 is a common implementation of Tikhonov regularization. However solving the optimization problem (1.7) becomes increasingly harder as the regularization parameter approaches zero. One way to alleviate this difficulty is to use an iterated Tikhonov regularization [11].

The use of iterated Tikhonov for nonlinear problems has been analyzed in [8] and it is known to give similar results to the Tikhonov regularization. We will further explore the use of Iterated Tikhonov regularization for geophysical inverse problems in a future paper, however, for now we discuss the basic theoretical properties of the method.

The iterated Tikhonov regularization is based on the observation that the nonlinear equation (1.8) can be rewritten as

$$\frac{m - m_0}{\tau} = \widehat{R}_{mm}^{-1} J(m)^\top (d^{\text{obs}} - QA(m)^{-1} b) \tag{4.24}$$

where $m_0 = 0$ and $\tau = \beta^{-1}$. Equation (4.24) is a single backward Euler step of the nonlinear ODE

$$\dot{m} = \widehat{R}_{mm}^{-1} J(m)^\top (d^{\text{obs}} - QA(m)^{-1}b) \tag{4.25}$$

$$m(0) = 0 \tag{4.26}$$

This implies that methods other than a single backward Euler method can be used. For example, in the classical implementation of the Iterated Tikhonov Regularization method a few backward Euler steps are taken with the same regularization parameter (or time step). Thus we solve the following sequence of problems with the same regularization parameter but with a changing reference model m_{ref}.

$$\min_m \ \frac{1}{2}\|QA(m)^{-1}b - d^{\text{obs}}\|^2 + \beta_k R(m - m_{\text{ref}_k}) \tag{4.27}$$

It can be shown under mild assumptions that if we set m_{ref} to the solution of the k^{th} problem, then the algorithm reduces the data misfit at each iteration. A sketch of the iterated Tikhonov regularization is as follows

Algorithm 2 Iterated Tikhonov Inversion algorithm:
$[m, \beta] \leftarrow \text{ITINV}(m_{\text{ref}}, \beta)$;

```
while true do
  given the reference model m_ref approximately solve (4.27) for m_k with starting guess m_ref
  if m_k fulfills chosen stopping criteria then
    break;
  end if
  update reference model, set m_ref ← m_k
end while
```

It is also possible to combine algorithms and change the regularization parameter as well as the reference model. In this approach **both** the regularization parameter and the reference model change at each iteration. This leads to faster reduction in the misfit.

Using an iterative regularization has one main advantage. The regularization parameter β used at each iteration is larger than the equivalent one used for classical Tikhonov regularization. This can be advantageous since the convergence of the linear solver at each iteration is highly dependent on the size of the regularization parameter. On the other hand, there are some disadvantages to the iterative regularization approach. The main one is that it is difficult to take advantage of solving problems on coarser grids [2]. Our code implements both strategies for the solution of the problem.

5 Efficient implementation of Quasi-Newton methods

Irrespective of whether we use a Tikhonov or iterated Tikhonov approach, an essential computation is the solution of equations (1.9). These will be solved with a CG solver and the

efficacy of solution depends strongly on the preconditioner. In previous work [21] we used the matrix $M = (0.1I + \beta W^T W)$ as a preconditioner. This worked well for large β but was less effective as β decreased. To improve computational efficiency we need to find a better approximation to the Hessian matrix $(J^\top J + \beta \mathcal{R}_{mm})$. We will use Quasi-Newton techniques to construct an approximation to the inverse Hessian and use that matrix as a preconditioner. As a prelude we discuss Quasi-Newton methods for the solution of the optimization problems (1.7) and (4.27). We start by introducing some notation, review the L-BFGS method and discuss important points in it's implementation.

For a fixed regularization parameter β and reference model m_{ref}, given two approximate solutions m_j and m_{j+1} and the two corresponding gradients g_j and g_{j+1}, we define, as is usual in Quasi-Newton methods,

$$\begin{aligned} s_j &= m_{j+1} - m_j \\ y_j &= g_{j+1} - g_j. \end{aligned}$$

The usual Secant Condition [9, 26] is

$$H_j s_j = y_j \tag{5.28}$$

where H_j is an approximation to the Hessian. In our case the Hessian has the structure

$$H_j = J_j^\top J_j + B_j + \beta R_{mm}$$

where R_{mm} is a known, sparse, easy-to-evaluate matrix and

$$C_j := J_j^\top J_j + B_j$$

is a dense hard-to-evaluate matrix which corresponds to a compact operator [11]. Common Quasi-Newton methods do not make use of the known part and evaluate the whole Hessian. However, we would like to use the known information about the Hessian.

To do that, we also note that for the applications we consider here, we have an explicit knowledge of the gradients with respect to the data objective function and the model objective function

$$g^d(m) = J(m)^\top (QA(m)^{-1} b - d^{\text{obs}}) \tag{5.29a}$$

$$g^m(m) = \beta R_m(m; m_{\text{ref}}). \tag{5.29b}$$

Using the separability of the objective function and the gradient we now discuss approximations to the Hessian.

In the BFGS method one uses equation (5.28) to approximate the inverse of the Hessian directly. The standard BFGS update is (see [9, 26])

$$H_{k+1}^{-1} = (I - \gamma_k s_k y_k^\top) H_k^{-1} (I - \gamma_k y_k s_k^\top) \; + \; \gamma_k s_k s_k^\top \tag{5.30}$$

where $\gamma_k = (s_k^\top y_k)^{-1}$. In the L-BFGS method only a limited number of vectors are kept and the matrix H, or H^{-1}, is never generated explicitly. The vectors $\{m_j, g_j\} \quad j = 1, \ldots \ell$

are saved and an inverse-Hessian matrix vector product can be calculated using recursion (see [25] for details).

Although it is possible to use the L-BFGS methods directly. As discussed in [13], if the matrix R_{mm} is constant (as when the regularization is quadratic) it is crucial to initiate the Hessian with βR_{mm}. This implies that when calculating the inverse-Hessian times a vector one needs to calculate the product $v = R_{mm}^{-1} w$. This can be done by the solution of the system $R_{mm} v = w$, which can be achieved directly, or using a multigrid method [14].

However, there are a few difficulties when using L-BFGS for the solution of our particular problem. Consider the L-BFGS solver imbedded into the algorithms 1, 2. When solving two adjacent problems, with a different regularization parameter or a different reference model, the gradients of the previous functions cannot be used. This implies that a standard implementation of L-BFGS requires restarting the process ignoring all (potentially useful) previous information. We therefore suggest a variant which makes use of previous approximations to the solution.

Assume we are in iteration ℓ of a BFGS algorithm with a reference model m_{ref_k} and a regularization parameter β_k and that we have the vectors

$$\{g_i = g_i^d + \beta_k R_m(m_i; m_{\text{ref}_\text{k}}),\ m_i\} \quad i = 1 \dots \ell$$

Now assume that the regularizer part has changed either because β_k or m_{ref_k} has changed to β_{k+1} and $m_{\text{ref}_{\text{k+1}}}$. Straight forward implementation of BFGS requires we restart the process ignoring information we have previously collected. However, we can circumvent this and define the following new set of vectors

$$\{\widehat{g}_i = g_i^d + \beta_{k+1} R_m(m_i; m_{\text{ref}_{\text{k+1}}}),\ m_i\} \quad i = 1 \dots \ell \tag{5.31}$$

It is clear that the set of vectors $\widehat{g}_i \quad i = 1, \dots, \ell$ are the gradients of the modified objective function which corresponds to previous models $m_1, \dots, m_\ell$. We can therefore use these vectors to generate a new BFGS approximation to the Hessian. Note that for the class of regularization we consider here, $R_m(m_i; m_{\text{ref}_{\text{k+1}}})$ can be calculated in linear time.

When using the new BFGS vectors we must test the Quasi-Newton condition, $y_j^\top s_j \geq 0$. If this condition does not apply to the new Quasi-Newton vectors then we simply drop the corresponding pair.

In the L-BFGS method one chooses a maximum number of vectors, (l_{max}, to be used. The convergence rate increases with l_{max} but so will storage requirements. Thus l_{max} will be problem and machine dependent.

Although the optimization problem could be solved directly with L-BFGS, we have found that it gave poor performance. We suspect that for small regularization parameters many BFGS vectors are needed in order to obtain a good approximation to the Hessian. Nevertheless, the BFGS approximation to the Hessian can still perform very well as a preconditioner for the GN system. This is the role it plays here.

6 Working with extended waveforms

Most transmitter waveforms have an on-time and off-time and the ideal scenario is that electromagnetic fields have completely decayed away before the next waveform begins. In conductive environments however the decay time might extend for a couple of periods of the input waveform. Modeling the field data then requires an extended time source that consists of a few repetitions of the waveform. As a example, consider a conductive sphere buried in a low conductivity background. Suppose that the sphere has a late-time decay constant $\tau = 100msec$. The waveform is a half-sinusoid pulse followed by an off-time. The data are acquired in the off-time and stacked over many periods. The final data set is assumed to be the steady state values. If the fundamental frequency of the transmitter is $10Hz$, then numerical experience shows that at least three cycles of a waveform need to be modeled before an approximate steady state solution is reached. Rather than carrying out this time stepping in all forward modelings for the inverse problem we now suggest a new method to deal with this difficulty.

Assume that the conductivity structure of the earth is known and that we can run the full forward problem with this conductivity structure. Let u_e be the field that corresponds to time t_e, which is the time after a few waveforms and assume that the magnetic field is measured only at time equal to, or larger than, t_e. We rewrite the system as

$$A(m)u = \begin{pmatrix} A_1(m) & & & & & & \\ B_2 & A_2(m) & & & & & \\ & \ddots & \ddots & & & & \\ & & B_e & A_e(m) & & & \\ & & & \ddots & \ddots & & \\ & & & & B_s & A_s(m) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_e \\ \vdots \\ u_s \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_e \\ \vdots \\ q_s \end{pmatrix} \tag{6.32}$$

For the inversion we are interested only in fields at times t_e and later. But to compute the field at t_e we need to know u_{t-1}, the field at t_{e-1}. If this were known then we could solve a much smaller forward problem

$$A_{\text{red}}u_{\text{red}} = \begin{pmatrix} A_e(m) & & \\ \ddots & \ddots & \\ & B_s & A_s(m) \end{pmatrix} \begin{pmatrix} u_e \\ \vdots \\ u_s \end{pmatrix} = \begin{pmatrix} q_e - B_e u_{e-1} \\ \vdots \\ q_s \end{pmatrix} \tag{6.33}$$

The problem is that the field u_{e-1} is not known unless we had previously performed a full forward modeling and computed and stored it. Then, if the forward modeling for times t_e and beyond are needed, we can avoid computing all previous fields and use the forward problem (6.33) instead of the full forward problem (6.32). This saves many of the computations.

However, in the course of the inversion process, we do not compute the same field twice. We need to compute the forward problem only after we update the model. Fortunately, since the PDE is stiff, if the change to the model is small then the change to the fields is exponentially smaller. Thus, for small changes in the model, we are able to obtain a good

enough solution to our forward problem even with the fields u_{e-1} computed from a near-by model. This observation motivates the following idea. Replace the forward modeling matrix $A(m)$ with A_{red} and guess the fields at time u_{e-1}. This implies that not only is the forward problem easier to solve, but the gradients and the product of the sensitivity matrix with a vector are also cheaper to compute. However, we must keep track of u_{e-1} and update it as m changes. This idea can be summarized by the following algorithm.

Algorithm 3 Gauss-Newton Time Domain Inversion with Approximate Initial Conditions: $[m] \leftarrow \text{GNNV}(m_{\text{ref}}, \beta)$;

Initialize: set $m = m_{\text{ref}}$ and choose β.
while true **do**
  [1] Compute the forward problem and store u_{e-1}
  [2] Compute the gradient (equation (1.8)) using the reduced forward modeling A_{red}
  [3] Compute the Gauss-Newton correction s using equation (1.9) with the reduced forward modeling A_{red}
  [4] Update the model, $m_{k+1} \leftarrow m_k + \alpha s$ where α is a line search parameter to guarantee reduction in the objective function.
  if m_{k+1} fulfill chosen stopping criteria **then**
    break;
  end if
end while

An important observation can be made when analyzing the algorithm. Note that by assuming that the perturbation in m is small between iterations we can perform step [1] only every second or third iterations. Furthermore, if we update u_{e-1} every second iteration then the computation of steps [1] and [2-3] can be done in parallel. Thus, given parallel architecture of two nodes we are able to significantly accelerate our algorithm.

7 Examples

7.1 2-prisms in a halfspace

We consider the case of a square loop with dimensions of 130×130 meters located just above the earth's surface. The transmitter current is a step-off at time zero and responses are measured in 32 logarithmically spaced times between $10^{-6} - 10^{-3}$ sec. The earth model is made of one conductive block of $10\,\Omega m$ and a resistive block of $1000\,\Omega/m$ buried in a half space of $100\Omega m$. There are 100 receiver locations.

The data are five component fields $(\mathbf{H}_x, \mathbf{H}_y, \mathbf{H}_z, \mathbf{E}_x, \mathbf{E}_y)$ on the surface inside the loop. Gaussian random noise has been added to the $16,300$ data.

The problem is discretized using 32^3 cells and the number of unknowns ($\mathbf{A}, \phi, \mathbf{H}$ for all space time and m) is roughly 14 million. The model objective function used $l_2 - norm$ of the gradients and a reference model and starting model what was equal to the true halfspace.

To solve the optimization problem, we have used the L-BFGS method. We allowed 4 L-BFGS iterations per-β before making a decision about either changing β or to continue with the same one. The results of our inversion are summarized in Figure 1. The target misfit of $16,300$ is almost achieved.

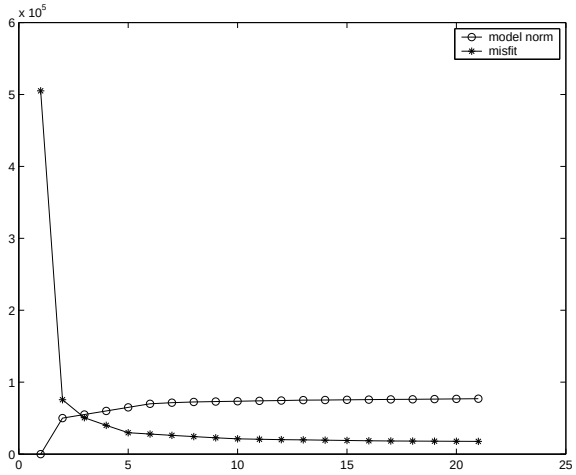

Figure 1: Misfit and model norm in the inversion process

The true and recovered images are shown in Figure 2. The conductive and resistive blocks are recovered in their true locations. As usual in these smooth model inversions, the recovered amplitudes of the target bodies are somewhat smaller than the true values and also edges are smoothed.

7.2 Reducing the volume

A $1.8km \times 1.1km$ loop is used in a UTEM survey. The waveform is a continuous on-time sawtooth. The background conductivity is $10000\,\Omega m$ and a conductive target of $1\,\Omega m$ is buried at $1250\,m$ depth. Three component magnetic field data are collected along 6 boreholes. One borehole intersects the target. The model and survey geometry are shown in Figure 3. The initial mesh for modeling the data used a volume that included the transmitter and the area of interest. It had 120,000 cells with cell sizes of $50\,m$. This mesh is too large to be used in the inversion and also the extended volume serves little purpose since the data and the volume of interest is quite confined.

A smaller mesh encompassing the volume of interest was generated using $25m$ cells in the core region. The total number of cells for the inversion was 68400. The corrective source procedure was implemented and the three components of the time derivative of the magnetic

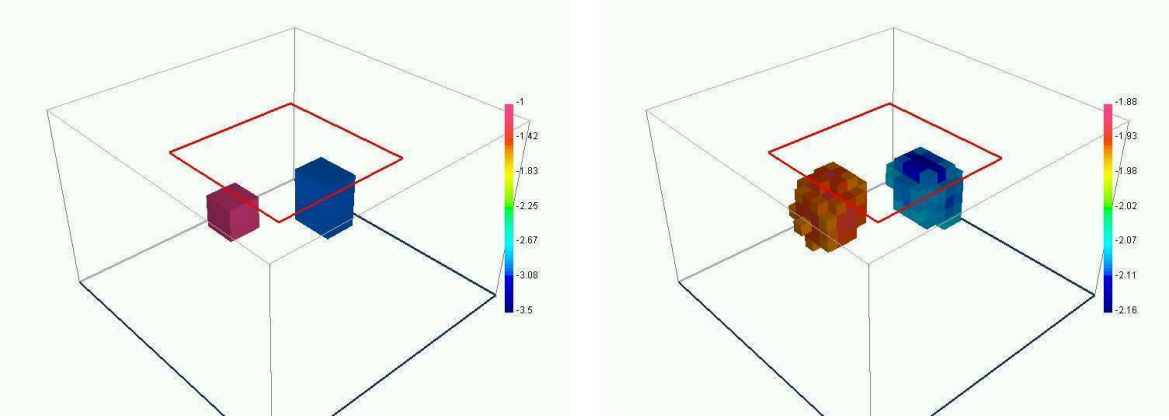

Figure 2: Inversion results (LEFT) the true model (RIGHT) recovered.

field were inverted. The true and recovered conductivities are shown in volume rendered and pixel format in Figures 4 and 5. An example of observed and predicted data for one of the boreholes is shown in Figure 6 .

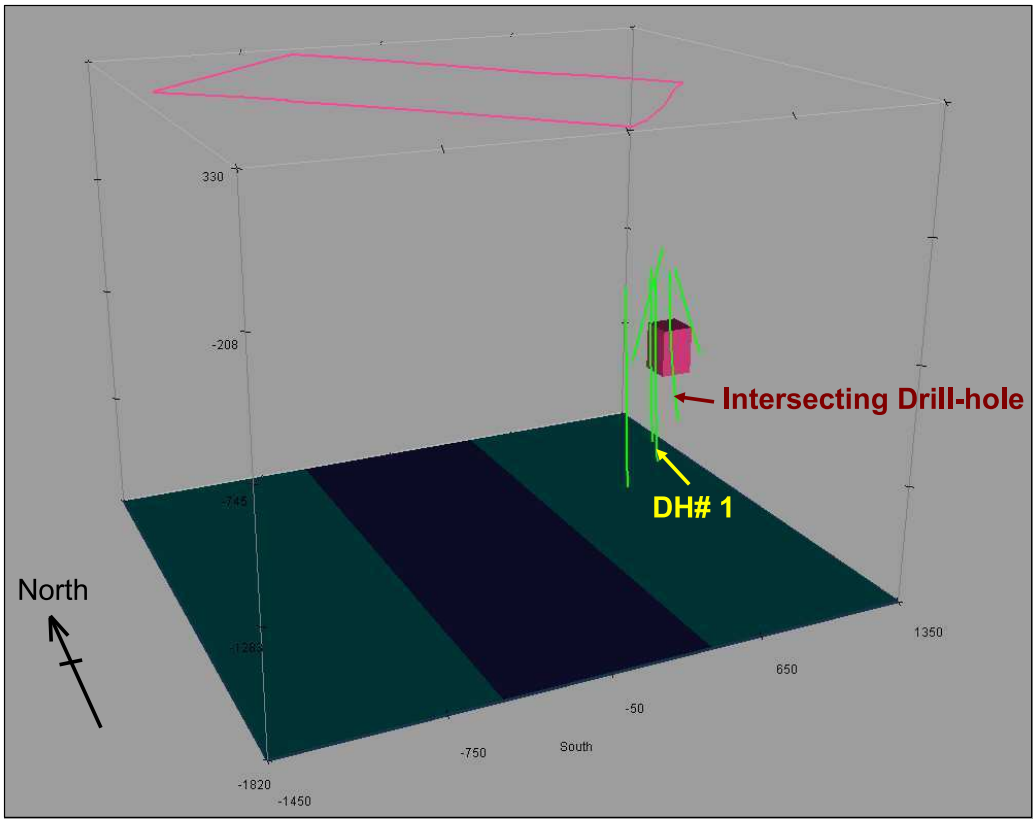

Figure 3: The source loop for the UTEM survey is at the surface. Six drill holes are shown in green. Three-component dB/dt data are acquired along the drill holes. The target is the conductive block. Drill hole ♯1 just grazes the outside of the target.

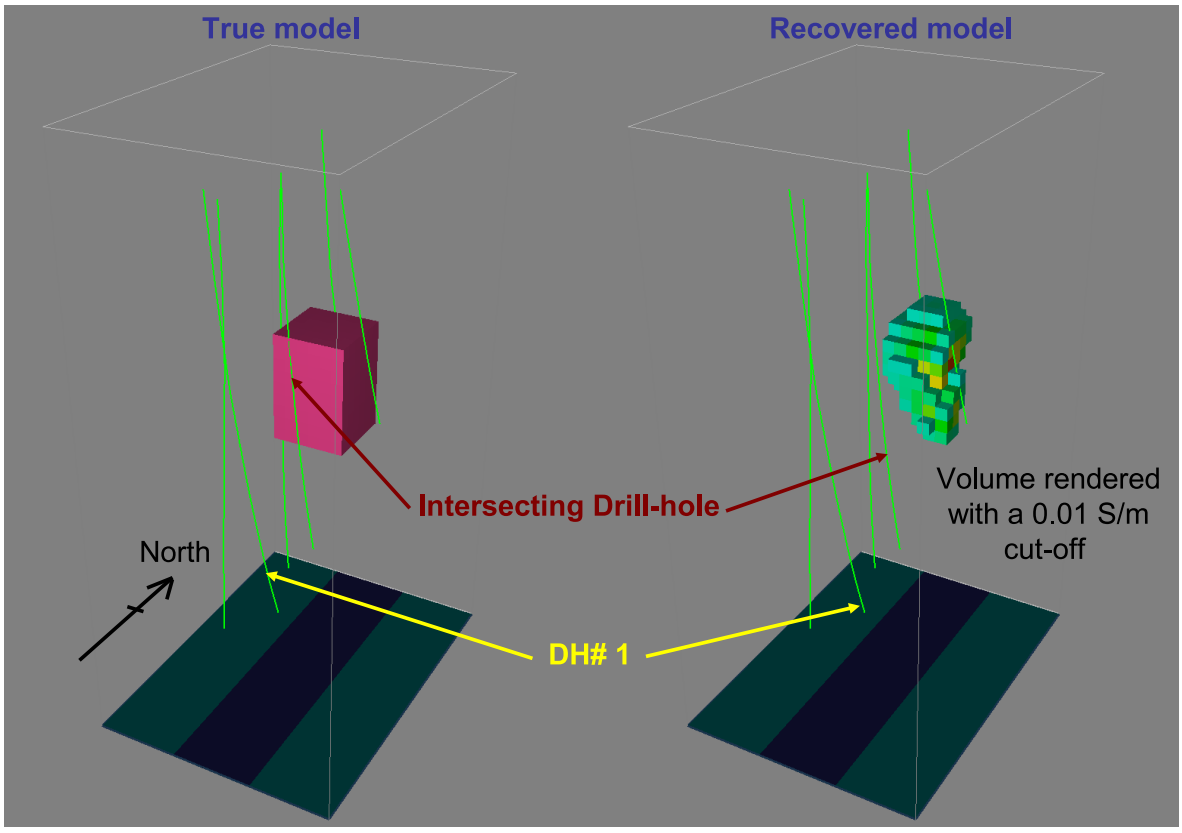

Figure 4: Volume rendered images of the true and recovered model.

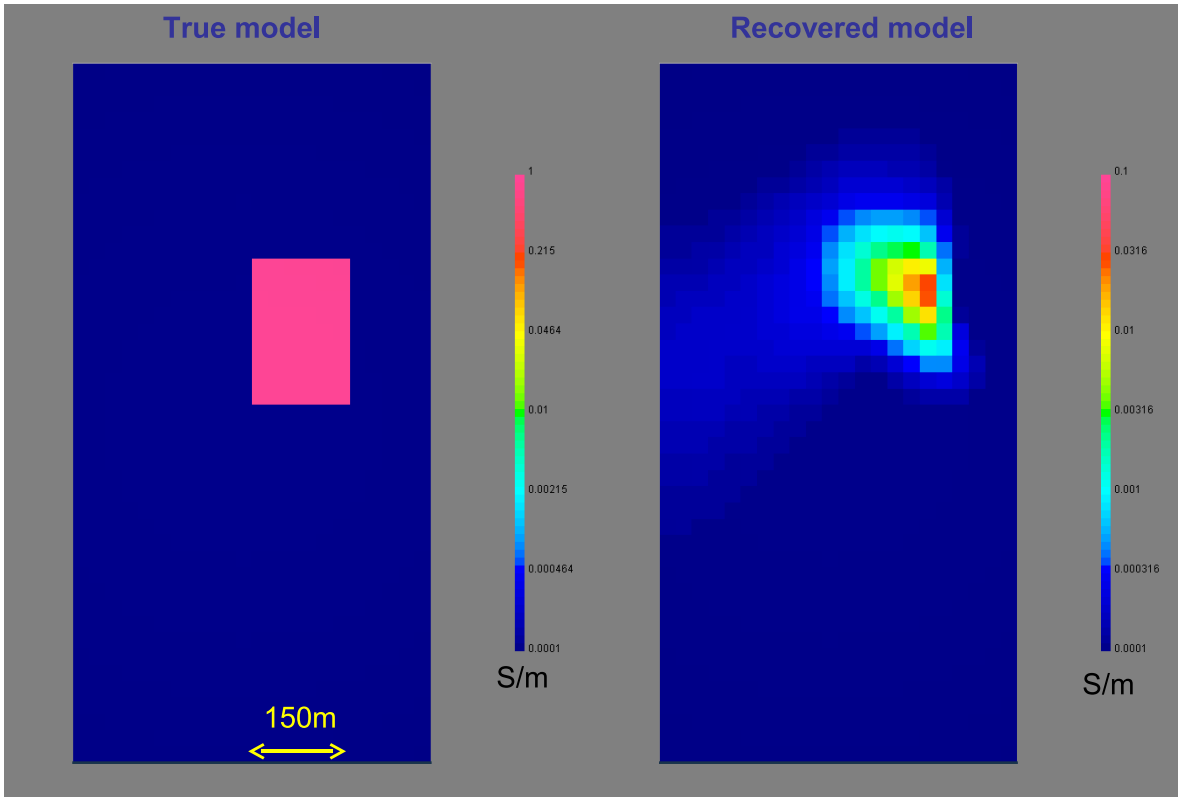

Figure 5: Cross-section of the true and recovered model.

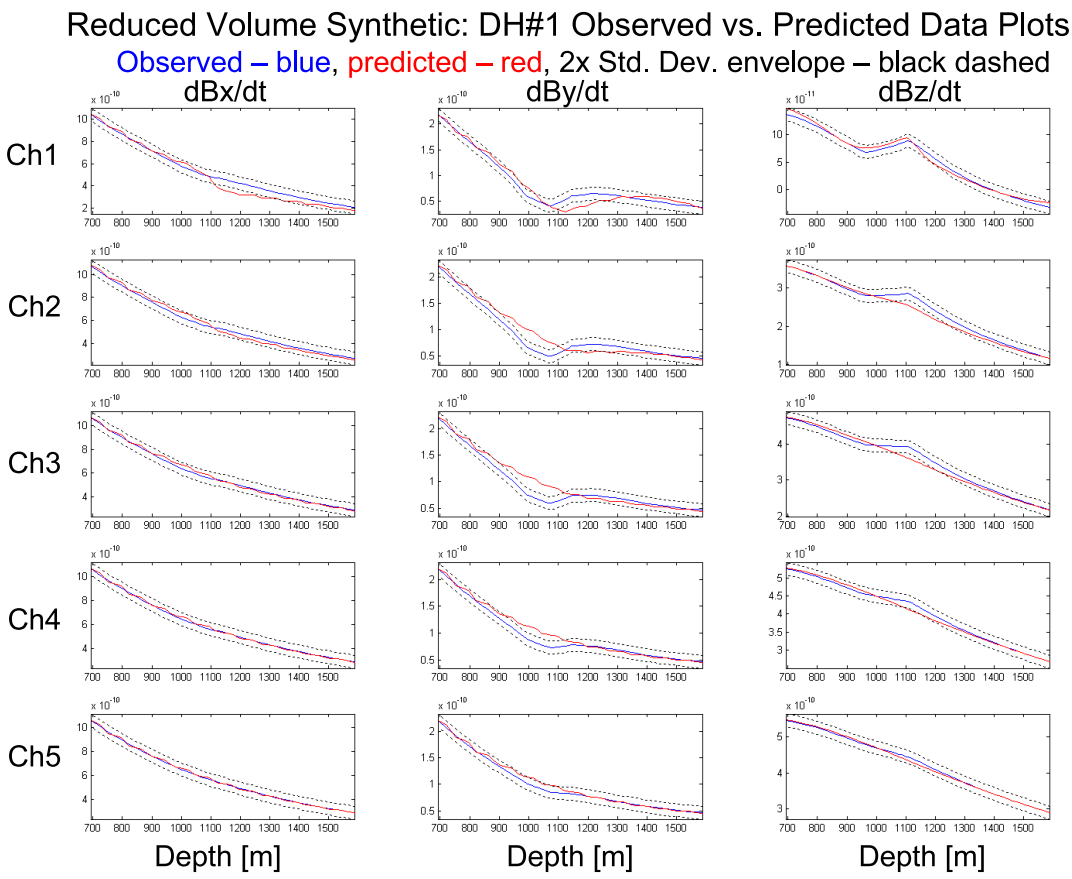

Figure 6: Observed and predicted data for the first time channels are plotted along borehole ♯1.

7.3 Working with extended waveforms

A surface loop survey is carried out over a conductive prism of $8S/m$. The waveform is a half-sinusoid followed by an off-time. The frequency of the transmitter is $f = 1000Hz$ and thus the total waveform, comprised of equal length of on-time and off-time segments, is $T = .001sec$. Because of its high conductivity and size, the time constant for the target is greater than T. Multiple waveforms are needed and in Figure 7 we plot the first four of these along with the time sampling. Constant time-steps were used in the on-time of the waveform, but logarithmic samples were used in the offtime. The number of time samples was 193. Also in Figure 8 we plot the decay curves for H_z for the first four cycles. There is a substantial difference between successive decays.

We next take the fourth decay and use that as data for the inverse problem. All three components of noise-contaminated magnetic field are inverted. The inversion is carried out by setting t_e to be the time of the first data channel. One processor is used to run the forward modelling from $[0, t_{e-1}]$ and stores the field u_{t_e-1}. This field is updated every time that the model is changed. The second processor, which is working on generating a model perturbation, begins each forward modelling with the stored field.

The success of this approach requires that the fields at t_e from a previous model are close to the fields that would have been obtained if a complete modelling had been done on the current model. We have compared the misfit obtained by starting with a near-by model, to that obtained by carrying out a full modelling. We note that the discrepancies between the two are reduced as the iterations increase. This is because size of the model perturbation decreases as the procedure reaches convergence.

An image of the true and constructed models are presented in Figure 9.

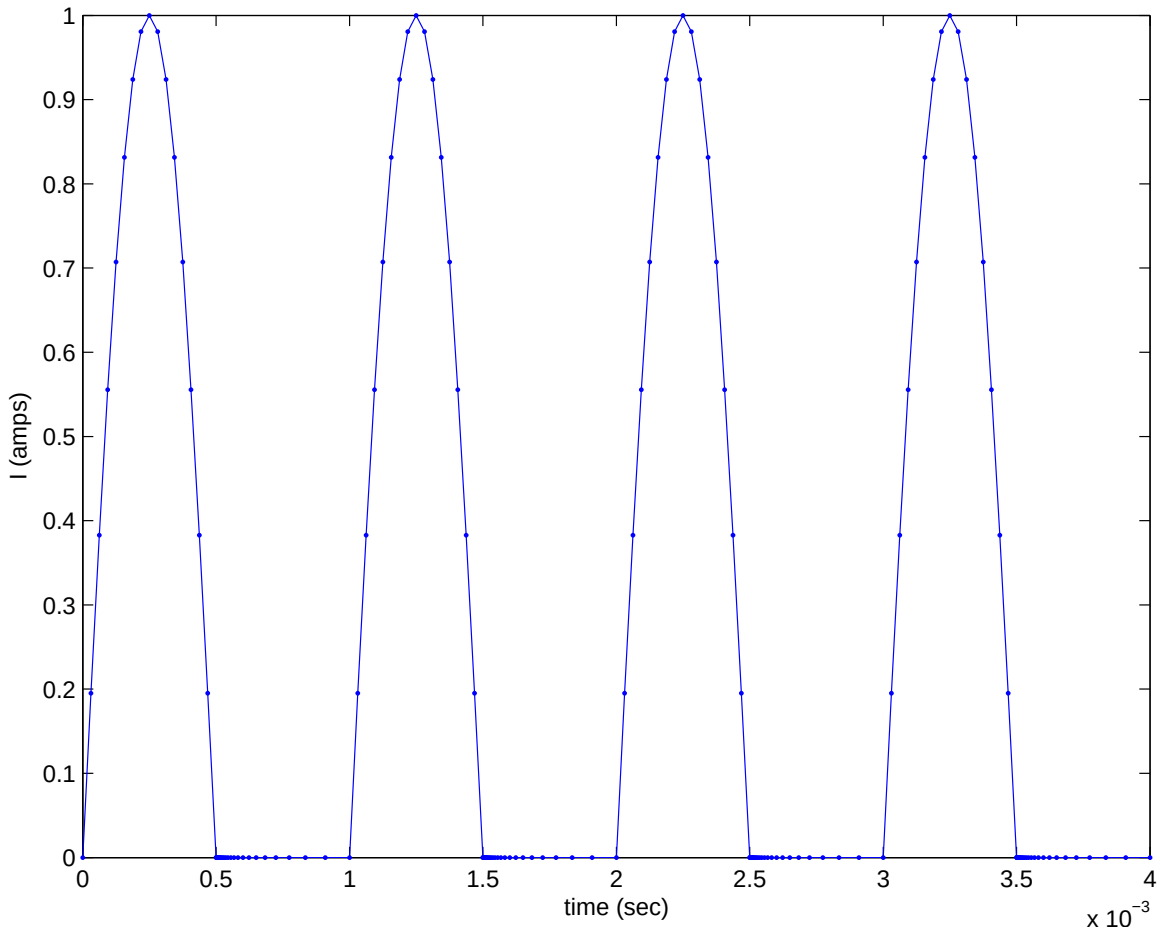

Figure 7: Four cycles of the waveform are shown. The dots denote the time sampling. Time samplings are uniform during the on-time and logarithmic during the off-time. Data at the beginning of the off-time for the fourth waveform will be inverted.

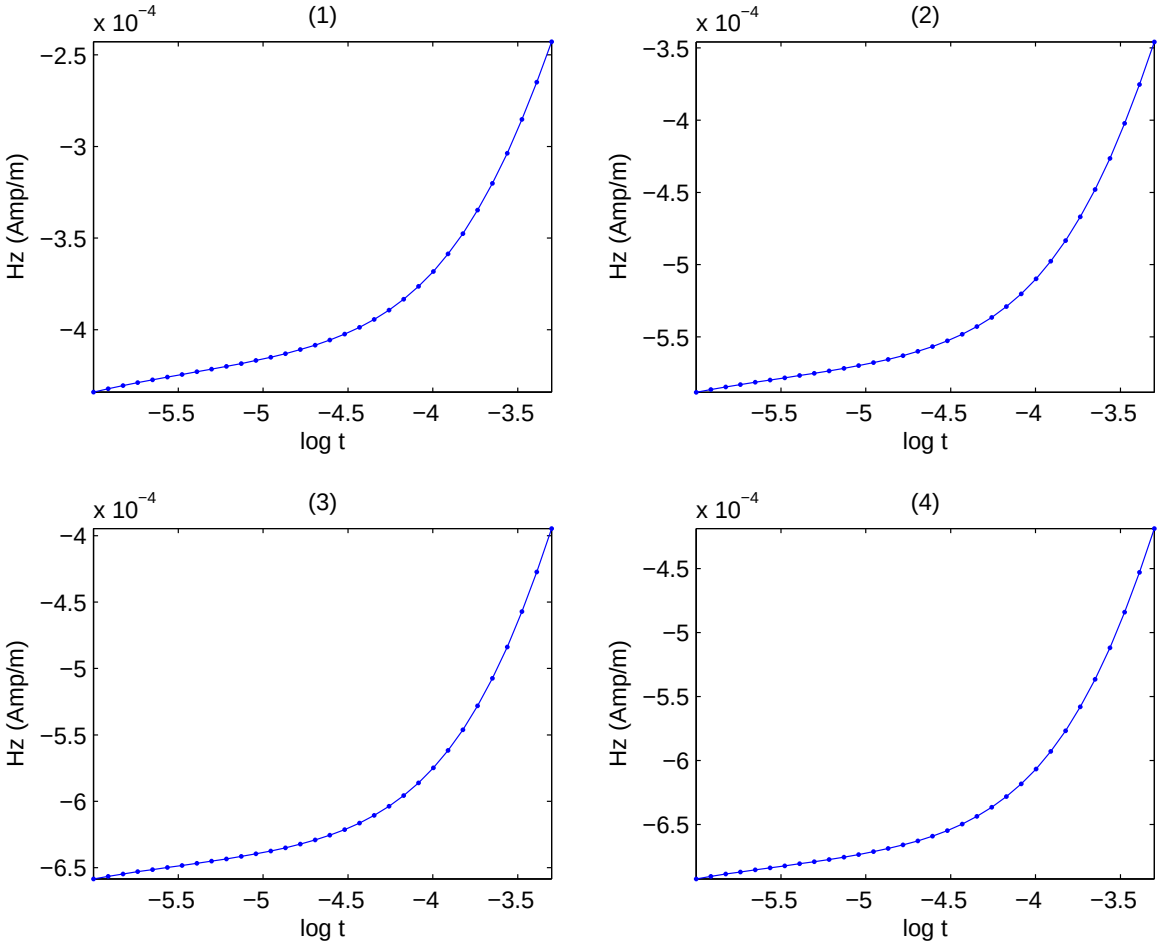

Figure 8: Response curves for Hz at a station near the center of the loop. Labels (1) - (4) respectively refer to the measurements after the respective number of on-time pulses.

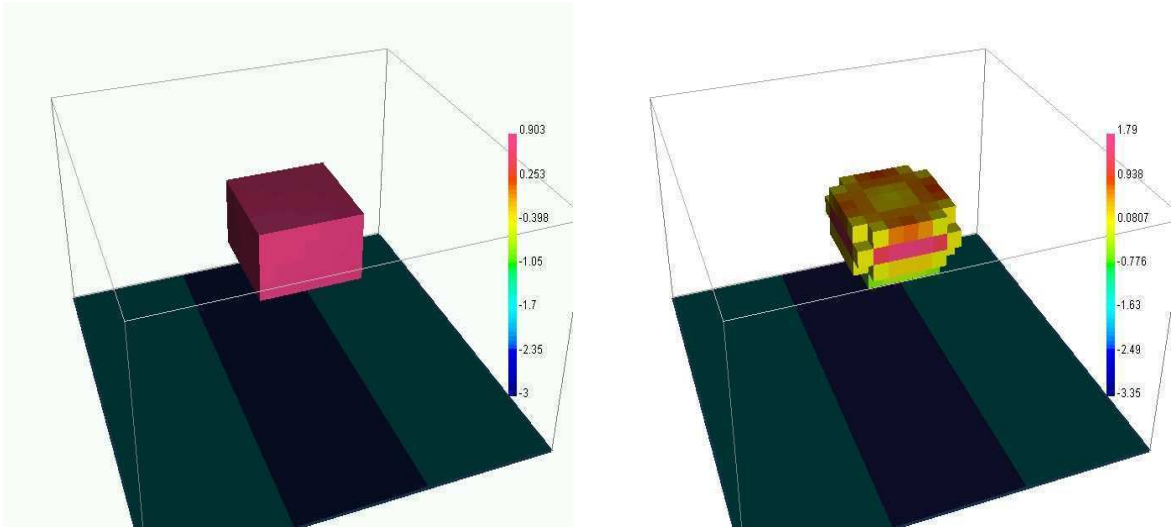

Figure 9: Inversion results (LEFT) the true model (RIGHT) recovered.

8 Summary

We have shown how time domain electromagnetic data can be inverted with a procedure that recovers the electrical conductivity model and the corresponding fields.

The problem is complex and computationally difficult. We therefore used a combination of techniques that allow us to solve this problem. These techniques allow us to reduce the volume under investigation, shorten the integration time of the forward problem and to use effective inversion methods and preconditioners to quickly converge to the solution. Results verify that our algorithm is working satisfactorily.

9 Appendix:Calculating Derivatives

In order to compute the gradient of the objective function (1.8) we need to evaluate the action of the sensitivity matrix times a vector. While the matrices $A(m)$ and Q are well define we further need to evaluate the matrix $G(m,u)$, where

$$G(m,u) = \frac{\partial[A(m)u]}{\partial m}.$$

The above formula implies that in order to calculate the matrix $G(m)$ we need to differentiate the forward modeling matrix times a vector with respect to m. This may look complicated at first; however, note that the matrix A is made of blocks and each block depends on m only through the matrix $M_{\hat{\sigma}}$. Therefore, if we know how to calculate

$$N(m,v) = \frac{\partial[M_{\hat{\sigma}}(m)v]}{\partial m}$$

then we can differentiate any product involving $M_{\hat{\sigma}}$. For example,

$$\frac{\partial}{\partial m}[\nabla_h \cdot M_{\hat{\sigma}} \nabla_h w] = \nabla_h \cdot N(m, \nabla_h w).$$

To calculate this derivative, we recall that $M_{\hat{\sigma}}$ operates on the discrete $\mathbf{A}$ or $\nabla_h \phi$ which are cell face variables. The matrix is diagonal and each of its elements has the form

$$M_{\hat{\sigma}}^{(ii)} = 2(\hat{\sigma}_1^{-1} + \hat{\sigma}_2^{-1})^{-1}$$

where $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the values of $\hat{\sigma}$ at the two sides of the face of the cell. From this form it is clear that $M_{\hat{\sigma}}^{-1}$ is *linear* with respect to $\hat{\sigma}^{-1}$ and therefore the matrix

$$N_r(v) = \frac{\partial[M_{\hat{\sigma}}^{-1} v]}{\partial[\hat{\sigma}^{-1}]}$$

is *independent of* $\hat{\sigma}$ and depends only on the vector field v at each cell. Using this observation and the chain rule we can easily calculate N,

$$\begin{aligned} N(m,v) &= \frac{\partial[M_{\hat{\sigma}}(m)v]}{\partial m} = \frac{\partial\left[((M_{\hat{\sigma}}(m))^{-1})^{-1} v\right]}{\partial m} = \frac{\partial\left[((M_{\hat{\sigma}}(m))^{-1})^{-1} v\right]}{\partial[\hat{\sigma}^{-1}]} \frac{\partial[\hat{\sigma}^{-1}]}{\partial m} = \\ & [M_{\hat{\sigma}}^{-1}]^{-2} N_r(v)\, \mathrm{diag}(\exp(-m)) = M_{\hat{\sigma}}^2 N_r(v)\, \mathrm{diag}(\exp(-m)). \end{aligned}$$

Given the above gradients we can proceed and discuss the solution of the nonlinear system (1.8).

References

[1] D. Aruliah and U. Ascher. Multigrid preconditioning for time-harmonic maxwell's equations in 3D. *Preprint*, 2000.

[2] U. Ascher and E. Haber. Grid refinement and scaling for distributed parameter estimation problems. *Inverse Problems*, 17:571–590, 2001.

[3] U. Ascher and E. Haber. A multigrid method for distributed parameter estimation problems. *ETNA*, 15:1–15, 2003.

[4] U.M. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, 1998.

[5] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization Parts I,II. *Preprints*, 2000.

[6] L. Borcea, J. G. Berryman, and G. C. Papanicolaou. High-contrast impedance tomography. *Inverse Problems*, 12, 1996.

[7] M. Cheney, D. Isaacson, and J.C. Newell. Electrical impedance tomography. *SIAM Review*, 41:85–101, 1999.

[8] U. Clarenz, M. Droske, and M. Rumpf. Towards fast non–rigid registration. In *Inverse Problems, Image Analysis and Medical Imaging, AMS Special Session Interaction of Inverse Problems and Image Analysis*, volume 313, pages 67–84. AMS, 2002.

[9] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, 1996.

[10] A. J. Devaney. The limited-view problem in diffraction tomography. *Inverse Problems*, 5:510–523, 1989.

[11] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer, 1996.

[12] C. Farquharson and D. Oldenburg. Non-linear inversion using general measures of data misfit and model structure. *Geophysics J.*, 134:213–227, 1998.

[13] Luc Gilles, C.R. Vogel, and J.M. Bardsley. Computational methods for a large-scale inverse problem arising in atmospheric optics. *Inverse Problems*, 18:237–252, 2002.

[14] E. Haber. A multilevel, level-set method for optimizing eigenvalues in shape design problems. *JCP*, 115:1–15, 2004.

[15] E. Haber. Quasi-newton methods methods for large scale electromagnetic inverse problems. *Inverse Peoblems*, 21, 2005.

[16] E. Haber and U. Ascher. Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients. *SIAM J. Scient. Comput.*, 22:1943–1961, 2001.

[17] E. Haber and U. Ascher. Preconditioned all-at-one methods for large, sparse parameter estimation problems. *Inverse Problems*, 17:1847–1864, 2001.

[18] E. Haber, U. Ascher, D. Aruliah, and D. Oldenburg. Fast simulation of 3D electromagnetic using potentials. *J. Comput. Phys.*, 163:150–171, 2000.

[19] E. Haber, U. Ascher, and D. Oldenburg. Solution of the 3D electromagntic inverse problem. In *Third Int. Symp. on 3D Electromagnetics*, Salt Lake City, Oct 1999.

[20] E. Haber, U. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse problems*, 16:1263–1280, 2000.

[21] E. Haber, U. Ascher, and D. Oldenburg. Inversion of frequency and time domain electromagnetic data. *Geophysics*, 69:1216–1228, 2004. n5.

[22] E. Haber and D. Oldenburg. A GCV based methods for nonlinear inverse problem. *Computational Geoscience*, 4, 2000. n1.

[23] E. Haber, D. Oldenburg, and U. Ascher. Modelling 3D maxwell's equations with non-continuous coefficients. In *SEG*, Calgary, Aug 2000.

[24] P. J. Huber. Robust estimation of a location parameter. *Ann. Math. Stats.*, 35:73–101, 1964.

[25] C.T. Kelley. *Iterartive Methods for Optimization*. SIAM, Philadelphia, 1999.

[26] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 1999.

[27] R. L. Parker. *Geophysical Inverse Theory*. Princeton University Press, Princeton NJ, 1994.

[28] Y. Saad. *Iterartive Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[29] N.C. Smith and K. Vozoff. Two dimensional DC resistivity inversion for dipole dipole data. *IEEE Trans. on geoscience and remote sensing*, GE 22:21–28, 1984. Special issue on electromagnetic methods in applied geophysics.

[30] C. Vogel. Sparse matrix computation arising in distributed parameter identification. *SIAM J. Matrix Anal. Appl.*, 20:1027–1037, 1999.

[31] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[32] S.H. Ward and G.W. Hohmann. Electromagnetic theory for geophysical applications. *Electromagnetic Methods in Applied Geophysics*, 1:131–311, 1988. Soc. Expl. Geophys.