doi: 10.1111/j.1365-246X.2007.03365.x

Inversion of time domain three-dimensional electromagnetic data

Eldad Haber,[1] Douglas W. Oldenburg and R. Shekhtman[2]

[1] *Mathematics and Computer Science, Emory University, Atlanta, GA* 30322, *USA. E-mail: haber@mathcs.emory.edu*
[2] *UBC-Geophysical Inversion Facility, Department of Earth and Ocean Sciences, University of British Columbia, Vancouver, BC* V6T *1Z4, Canada*

Accepted 2007 January 31. Received 2007 January 31; in original form 2005 September 12

SUMMARY

We present a general formulation for inverting time domain electromagnetic data to recover a 3-D distribution of electrical conductivity. The forward problem is solved using finite volume methods in the spatial domain and an implicit method (Backward Euler) in the time domain. A modified Gauss–Newton strategy is employed to solve the inverse problem. The modifications include the use of a quasi-Newton method to generate a pre-conditioner for the perturbed system, and implementing an iterative Tikhonov approach in the solution to the inverse problem. In addition, we show how the size of the inverse problem can be reduced through a corrective source procedure. The same procedure can correct for discretization errors that inevidably arise. We also show how the inverse problem can be efficiently carried out even when the decay time for the conductor is significantly larger than the repetition time of the transmitter wave form. This requires a second processor to carry an additional forward modelling. Our inversion algorithm is general and is applicable for any electromagnetic field ($\mathbf{E}$, $\mathbf{H}$, $d\,\mathbf{B}/dt$) measured in the air, on the ground, or in boreholes, and from an arbitrary grounded or ungrounded source. Three synthetic examples illustrate the basic functionality of the algorithm, and a result from a field example shows applicability in a larger-scale field example.

Key words: electromagnetic, inversion, time-domain.

1 INTRODUCTION

In this paper, we develop an inversion methodology for large-scale parameter identification problems which evolve from 3-D time domain electromagnetic simulations. This type of problem is of major interest in geophysics, medical imaging and non-destructive testing; see for example (Smith & Vozoff 1984; Devaney 1989; Parker 1994; Borcea *et al.* 1996; Cheney *et al.* 1999; Vogel 1999; Haber & Ascher 2001) and references therein. The forward modelling problem 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 has been considerable progress in this area, much of it stemming from initial works of Wang & Hohmann (1993) and Wang *et al.* (1994). Some of the most recent extensions of that work has been by Newman & Commer (2005) and Newman & Boggs (2004). They use explicit time stepping when solving the forward problem and non-linear conjugate gradients to solve the 3-D inverse problem. Their method is highly efficient for early times however, in the presence of large conductors small time steps are needed which makes the forward problem slow for long integration times. Here we use implicit methods to solve the forward problem and a Gauss–Newton strategy for the inverse problem.

There are many practical challenges to solving the electromagnetic inverse problem. First, a fast, accurate and reliable algorithm for 3-D forward modelling is required. Second, the sensitivities for such problems are too numerous to be formed or stored in a reasonable amount of time and memory space. Finally, finding the minimum of the objective function obtained by matching the data to some extent (e.g. depending on noise statistics) and incorporating *a priori* information on the conductivity field can be difficult due to the non-linearity 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 Controlled Source Audio Magneto-Telluric (CSAMT) surveys, the 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- 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 waveform simulation 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

$$\mathcal{A}(m)u - b = 0, \tag{1}$$

where $\mathcal{A}(m)$ is some 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

$$d^{\text{obs}} = \mathcal{Q}u + \epsilon, \tag{2}$$

where $\mathcal{Q}$ is a measurement operator which projects the fields (or their derivatives or integrals) onto the measurement locations in 3-D 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

$$\begin{aligned} &\min_{m,u} \ \frac{1}{2}\|\mathcal{Q}u - d^{\text{obs}}\|^2 + \beta\mathcal{R}(m) \\ &\text{subject to} \ \ \mathcal{A}(m)u - b = 0. \end{aligned} \tag{3}$$

Here, $\beta > 0$ is the regularization parameter, and $\mathcal{R}(\cdot)$ is a regularization operator reflecting our *a priori* information. Typically, we assume that m is a piecewise smooth function over the spatial domain Ω in 3-D, so we design an $\mathcal{R}(\cdot)$ that involves some norm of ∇m over Ω, for example, weighted L_2 or L_1 or a Huber combination (Huber 1964; Farquharson & Oldenburg 1998)

$$\mathcal{R}(\nabla m) = \begin{cases} |\nabla m| & \text{if } \ |\nabla m| \geq \theta \\ \frac{1}{2\theta}|\nabla m|^2 + \frac{\theta}{2} & \text{if } \ |\nabla m| < \theta \end{cases},$$

where θ can be seen as a jump parameter. For gradients with values smaller than θ smooth solutions are assumed but for large gradients discontinuities are allowed. This type of regularization can be written as

$$\mathcal{R}(m) = \int_\Omega \rho(|\nabla m|)\, d\mathbf{x} + \gamma \frac{1}{2}\int_{\partial\Omega} (m - m_{\text{ref}})^2 d\mathbf{x}, \tag{4}$$

where ρ is given by the Huber function, m_{ref} 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 guarantees that the Hessian of the regularizer is invertible.

Next, the problem (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

$$\begin{aligned} &\min_{m,u} \ \frac{1}{2}\|Qu - d^{\text{obs}}\|^2 + \beta R(m) \\ &\text{subject to} \ \ A(m)u - b = 0, \end{aligned} \tag{5}$$

where u, m and b 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 non-singular. The discrete regularization function has the form

$$R(m) = e^\top \rho(|\nabla_h m|) + \gamma ||W(m - m_{\text{ref}}||^2, \tag{6}$$

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

In previous work (Haber *et al.* 2004), we have used an inexact, all-at-once methodology (Biros & Ghattas 2005; Haber & Ascher 2001; Ascher & Haber 2003) to solve (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 the time–space history of all of the fields. In practical field situations, where we are concerned with large volumes, small cells, and multiple transmitters, this approach is not yet feasible.

In this work, we aim to generate a practical time-domain inversion that is applicable to 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, that is, solve for the field u given the model m. This yields the unconstrained optimization problem

$$\min_m \ \frac{1}{2}\|QA(m)^{-1}b - d^{\text{obs}}\|^2 + \beta R(m). \tag{7}$$

This leads to the Euler-Lagrange system

$$g(m) = J(m)^\top [QA(m)^{-1}b - d^{\text{obs}}] + \beta R_m(m) = 0, \tag{8}$$

where

$$J(m) = -QA(m)^{-1}G(m)$$

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

and $J(m)$ is the sensitivity matrix (see Haber *et al.* 2000, for derivation) and $R_m(m) = \frac{\partial R}{\partial m}$.

We search for an m that solves (8) and yields a minimum of the function. The evaluation of $g(m)$ involves a number of computations. To calculate $R_m(m)$ we use the chain rule

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

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 (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 modelling must be carried out. In this, and in all other descent methods for solving the optimization problem, a globalization strategy is necessary to guaranty convergence to a local minimum (rather than a saddle point or a maximum). Here we have used an Armijo line search strategy (see Nocedal & Wright 1999).

Newton-type methods are generally far superior to steepest descent methods when solving (8). If a Gauss–Newton method is used, then at each iteration we solve the linear (but dense) system

$$[J(m)^\top J(m) + \beta R_{mm}]s = -g(m), \tag{9}$$

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 pre-conditioners, 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

$$B(m)s = -g(m), \tag{10}$$

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 (Nocedal & Wright 1999). 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 Haber 2005, for details).

In this paper, we build on our prior work and improve our inversion algorithm by introducing new inversion strategies.

(1) We combine the strengths of Gauss–Newton and quasi-Newton approaches.

(2) 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.

(3) We explore a particular implementation of L-BFGS and use the constructed approximate inverse Hessian as a pre-conditioner for the Gauss–Newton iteration. This substantially reduces the number of CG iterations in solving (9).

(4) We present new 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 to ameliorate 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. In Section 7, we present three synthetic examples that illuminate the basic functionality of the inversion and we also provide results from a larger-scale field example. The paper is summarized in Section 9.

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. This section is a summary of our previous work (Haber *et al.* 2000; Haber & Ascher 2001; Haber *et al.* 2004) and we add it here for completeness.

2.1 Reformulation and discretization

The time-dependent Maxwell equations can be written as

$$\nabla \times \mathbf{E} + \mu \frac{\partial \mathbf{H}}{\partial t} = 0, \tag{11a}$$

$$\nabla \times \mathbf{H} - \sigma \mathbf{E} - \epsilon \frac{\partial \mathbf{E}}{\partial t} = \mathbf{s}(t) \tag{11b}$$

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

$$\mathbf{n} \times \mathbf{H} = 0 \tag{11c}$$

$$\mathbf{H}(0, \mathbf{x}) = \mathbf{H}_0 \tag{11d}$$

$$\mathbf{E}(0, \mathbf{x}) = 0 \tag{11e}$$

although other boundary and initial conditions could be used.

It is possible to verify that the above system is very stiff (Ascher & Petzold 1998; Haber *et al.* 2004). This implies that typically many time scales exist and careful numerical treatment is needed in order to solve the problem. We therefore, turn to implicit methods, and use a Backward Difference Formula (BDF) 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 (Haber *et al.* 1999, 2004, 2000).

Semi-discretizing (11), (11c) over a time step $[t_n, t_{n+1}]$ yields the equations

$$\nabla \times \mathbf{E}^{n+1} + \alpha_n \mu \mathbf{H}^{n+1} = \alpha_n \mu \mathbf{H}^n \quad \text{in } \Omega \tag{12a}$$

$$\nabla \times \mathbf{H}^{n+1} - (\sigma + \alpha_n \epsilon)\mathbf{E}^{n+1} = \mathbf{s}^{n+1} - \alpha_n \epsilon \mathbf{E}^n \quad \text{in } \Omega \tag{12b}$$

$$\mathbf{n} \times \mathbf{H}^{n+1} = 0 \quad \text{on } \partial\Omega, \tag{12c}$$

where $\alpha_n = (t_{n+1} - t_n)^{-1}$. The superscripts in (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 (12)

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

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

$$\nabla \times \mu^{-1} \nabla \times \mathbf{E}^{n+1} + \alpha_n \sigma \mathbf{E}^{n+1} + \alpha_n^2 \epsilon \mathbf{E}^{n+1} = \alpha_n (\nabla \times \mathbf{H}^n + \alpha_n \epsilon \mathbf{E}^n - \mathbf{s}^{n+1}). \tag{13}$$

As discussed in (Haber *et al.* 2000; Haber & Ascher 2001) the discretization of the system (13) is difficult to solve using iterative methods. Introducing the potentials **A**, ϕ to write $\mathbf{E} = \mathbf{A} + \nabla\phi$ we get

$$\begin{aligned} &(\nabla \times \mu^{-1} \nabla \times - \nabla \mu^{-1} \nabla \cdot)\mathbf{A} + \alpha_n (\sigma + \alpha_n \epsilon)(\mathbf{A} + \nabla\phi)^{n+1} = \\ &\alpha_n (\nabla \times \mathbf{H}^n + \alpha_n \epsilon \mathbf{E}^n - \mathbf{s}^{n+1}) \end{aligned} \tag{14a}$$

$$\nabla \cdot (\sigma + \alpha_n \epsilon)(\mathbf{A} + \nabla\phi)^{n+1} = \alpha_n \nabla \cdot \epsilon \mathbf{E}^n - \nabla \cdot \mathbf{s}^{n+1}, \tag{14b}$$

where we have subtracted the stabilizing term $\nabla \mu^{-1} \nabla \cdot \mathbf{A}$ from the first equation. A discretization of the system (14) is better conditioned than (13) because the null space of the **curl** is eliminated by adding the stabilizing term.

For ease of notation we introduce $\hat{\sigma} = \sigma + \alpha_n \epsilon$. The continuous equations are discretized in space using a finite volume technique on a staggered grid (see details in Haber *et al.* 2000; Haber & Ascher 2001; Haber *et al.* 2004). Variables **A** and **E** are defined at the centre of cell faces, **H** is defined on the edges, and ϕ in the cell centres. Upon discretization we obtain

$$\begin{pmatrix} L_\mu + \alpha_n M_{\hat{\sigma}} & \alpha_n M_{\hat{\sigma}} G \\ D M_{\hat{\sigma}} & D M_{\hat{\sigma}} G \end{pmatrix} \begin{pmatrix} \mathbf{A} \\ \phi \end{pmatrix} = \begin{bmatrix} \alpha_n (C_e \mathbf{H}^n + \alpha_n \epsilon \mathbf{E}^n - \mathbf{s}^{n+1}) \\ \epsilon \alpha_n D \mathbf{E}^n - D \mathbf{s}^{n+1} \end{bmatrix}, \tag{15}$$

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 $\sigma + \alpha_n \epsilon$ 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. Finally, the matrix M_μ^c is a discretization of μ in cell centres and it assumes that μ is a scalar quantity (rather than a tensor).

This linear system can be solved using standard iterative methods (Saad 1993) and effective scaleable pre-conditioners can be designed for it (Aruliah & Ascher 2002; Haber *et al.* 2000).

2.2 Summary—the forward problem

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

We note that it is possible to rewrite the system as a lower block bidiagonal system for **A**, ϕ and **H** of the form

$$A(m)u = \begin{pmatrix} A_1(m) & & & & \\ B_2 & A_2(m) & & & \\ & B_3 & A_3(m) & & \\ & & \ddots & \ddots & \\ & & & B_s & A_s(m) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_s \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_s \end{pmatrix}, \tag{16}$$

where

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

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

In the case of multiple sources we obtain a block diagonal system where each block has the same structure as (16). Note that only the diagonal blocks in (16) depend on the conductivity. Also, once we have an efficient solver for one block $A_k(m)$, solving the forward problem (16) 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 modelled. 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

$$A(m)\tilde{u} = \tilde{q}. \tag{17}$$

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

$$A_h(m)u_h = q_h, \tag{18}$$

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 eq. (17) 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 eq. (18). The residual is

$$s := A_h(m_0)\tilde{u}(m_0) - q_h. \tag{19}$$

This discrepancy can be regarded as a corrective source and added to the discrete equations. Since the source does not depend on m we solve a corrected equation

$$A_h(m)u_h = q_h + s. \tag{20}$$

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 modelling 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, we are able to iteratively change m_0 in the course of the inversion process. We have proved in Haber (2006) that such a process is a convergent one.

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 (Ward & Hohmann 1988) or by using 1-D 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 (13) and thus the source for time t_{n+1} is

$$\mathbf{s}_0 = C_e M_\mu C_f \mathbf{E}_0^{n+1} + \alpha_n M_{\sigma_0}\mathbf{E}_0^{n+1} + \alpha_n^2 \epsilon \mathbf{E}_0^{n+1} - \alpha_n\left(C_e\mathbf{H}_0^n + \alpha_n\epsilon\mathbf{E}_0^n - \mathbf{s}\right). \tag{21}$$

Using the potentials, we therefore, have

$$\begin{pmatrix} L_\mu + \alpha_n M_{\hat\sigma} & \alpha_n M_{\hat\sigma} G \\ D M_{\hat\sigma} & D M_{\hat\sigma} G \end{pmatrix}\begin{pmatrix} \mathbf{A} \\ \phi \end{pmatrix} = \begin{bmatrix} \mathbf{s}_0 + \alpha_n(C_e\mathbf{H}^n + \alpha_n\epsilon\mathbf{E}^n - \mathbf{s}) \\ D\mathbf{s}_0 + \alpha_n D\epsilon\mathbf{E}^n - D\mathbf{s} \end{bmatrix}. \tag{22}$$

Note that upon using $\mathbf{s}_0$ in (13), and the equivalent $\mathbf{A} - \phi$ formula (22) 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 (22) allow us to dramatically reduce the volume under consideration without sacrificing accuracy.

4 INVERSION FRAMEWORK AND ALGORITHMS

As discussed in Section 1 we aim to solve the minimization problem (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 (Parker 1994) or Generalized Cross Validation (GCV) (Hansen 1998; Wahba 1998; Haber & Oldenburg 2001). A general sketch of the algorithm that achieves this goal is as follows

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

```
while true do
    given $\beta$ approximately solve (7) for $m_\beta$ with starting guess $m_0$
    if $m_\beta$ fulfils chosen stopping criteria then
        break;
    end if
    update $\beta$, set $m_0 \leftarrow m_\beta$
end while
```

Algorithm 1 is a common implementation of Tikhonov regularization. However solving the optimization problem (7) becomes increasingly harder as the regularization parameter approaches zero. One way to alleviate this difficulty is to use an iterated Tikhonov regularization (Engl *et al.* 1996).

The use of iterated Tikhonov for non-linear problems has been analysed in Clarenz *et al.* (2002) and it is known to give similar results to the Tikhonov regularization. Some background about the iterated Tikhonov approach is provided in the Appendix A.

The iterated Tikhonov regularization is based on the observation that the non-linear eq. (8) can be rewritten as

$$\frac{m - m_0}{\tau} = \widehat{R}_{mm}^{-1} J(m)^\top [d^{\mathrm{obs}} - QA(m)^{-1}b], \tag{23}$$

where $m_0 = 0$ and $\tau = \beta^{-1}$. Eq. (23) is a single backward Euler step of the non-linear ODE

$$\dot m = \widehat{R}_{mm}^{-1} J(m)^\top [d^{\mathrm{obs}} - QA(m)^{-1}b] \tag{24}$$

$$m(0) = 0. \tag{25}$$

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}.

$$\min_m \; \frac{1}{2}\|QA(m)^{-1}b - d^{\mathrm{obs}}\|^2 + \beta_k R(m - m_{\mathrm{ref}_k}). \tag{26}$$

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

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

```
while true do
    given the reference model $m_{\mathrm{ref}}$ approximately solve (4.26) for $m_k$ with starting guess $m_{\mathrm{ref}}$
    if $m_\beta$ fulfils chosen stopping criteria then
        break;
    end if
    update $\beta$, set $m_{\mathrm{ref}} \leftarrow m_k$
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 (Ascher & Haber 2001). Our code, therefore, implements either the classical or iterated Tikhonov solution.

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 eq. (9), which we write again for the sake of completeness

$$[J(m)^\top J(m) + \beta R_{mm}]s = -g(m).$$

These will be solved with a CG solver and the efficacy of solution depends strongly on the pre-conditioner. In previous work (Haber *et al.* 2004), we used the matrix $M = (0.1I + \beta W^T W)$ as a pre-conditioner. 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 pre-conditioner. As a prelude we discuss quasi-Newton methods for the solution of the optimization problems (7) and (26). 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,

$$s_j = m_{j+1} - m_j$$

$$y_j = g_{j+1} - g_j.$$

The usual Secant Condition (Dennis & Schnabel 1996; Nocedal & Wright 1999) is

$$H_j s_j = y_j, \qquad (27)$$

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 (Engl *et al.* 1996). 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

$$g^d(m) = J(m)^\top (QA(m)^{-1}b - d^{\mathrm{obs}}) \qquad (28a)$$

$$g^m(m) = \beta R_m(m; m_{\mathrm{ref}}). \qquad (28b)$$

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

In the BFGS method one uses eq. (27) to approximate the inverse of the Hessian directly. The standard BFGS update is (see Dennis & Schnabel 1996; Nocedal & Wright 1999)

$$H_{k+1}^{-1} = \left(I - \gamma_k s_k y_k^\top\right) H_k^{-1} \left(I - \gamma_k y_k s_k^\top\right) + \gamma_k s_k s_k^\top, \qquad (29)$$

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 $\{m_j, g_j\}$ $j = 1, \dots \ell$ are saved and an inverse-Hessian matrix vector product can be calculated using recursion (see Kelley 1999, for details).

Although it is possible to use the L-BFGS methods directly. As discussed in Gilles *et al.* (2002), 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 (Haber 2004).

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

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

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

$$\left\{\widehat{g}_i = g_i^d + \beta_{k+1} R_m(m_i; m_{\text{ref}_{\text{k}+1}}),\ m_i\right\}\quad i = 1 \dots \ell. \tag{30}$$

It is clear that the set of vectors $\widehat{g}_i$ $i = 1, \dots, \ell$ 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, $y_j^\top s_j \ge 0$. 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 pre-conditioner 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. Modelling 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 $\tau = 100$ ms. 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 10 Hz, then numerical experience shows that at least three cycles of a waveform need to be modelled before an approximate steady state solution is reached. Rather than carrying out this time stepping in all forward modellings 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

$$u = \begin{pmatrix} A_1(m) & & & & & \\ B_2 & A_2(m) & & & & \\ & \ddots & \ddots & & & \\ & & B_e & A_e(m) & & \\ & & & \ddots & \ddots & \\ & & & & B_s & A_s(m) \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_e \\ \vdots \\ u_s \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_e \\ \vdots \\ q_s \end{bmatrix}. \tag{31}$$

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

$$A_{\mathrm{red}} u_{\mathrm{red}} = \begin{pmatrix} A_e(m) & & \\ \ddots & \ddots & \\ & B_s & A_s(m) \end{pmatrix} \begin{pmatrix} u_e \\ \vdots \\ u_s \end{pmatrix} = \begin{pmatrix} q_e - B_e u_{e-1} \\ \vdots \\ q_s \end{pmatrix} \tag{32}$$

The problem is that the field u_{e-1} is not known unless we had previously performed a full forward modelling and computed and stored it. Then, if the forward modelling for times t_e and beyond are needed, we can avoid computing all previous fields and use the forward problem (32) instead of the full forward problem (31). This saves many of the computations.

In the course of the inversion process, 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 (previous) model. This observation motivates the following idea. Replace the forward modelling 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.

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

```
  Initialize: set $m = m_{\mathrm{ref}}$ and choose $\beta$.
  while true do
1   Compute the forward problem and store $u_{e-1}$
2   Compute the gradient (eq. 8) using the reduced forward modelling $A_{\mathrm{red}}$
3   Compute the Gauss–Newton correction $s$ using eq. (9) with the reduced forward modelling $A_{\mathrm{red}}$
4   Update the model, $m_{k+1} \leftarrow m_k + \alpha s$ where $\alpha$ is a line search parameter to guarantee reduction in the objective function.
      if $m_{k+1}$ fulfil chosen stopping criteria then
        break;
      end if
  end while
```

An important observation can be made when analysing 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 iteration. Furthermore, if we update u_{e-1} with a near-by model 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

In this section, we present three synthetic and one field example. The first example is a generic inversion where we invert data arising from a loop on the earth's surface; the goal is to illustrate the basic functionality of our algorithm. The second and third examples, respectively, demonstrate the use of the corrective source technique to reduce the volume, and show how extended waveforms can be handled more efficiently. We conclude with a larger-scale field example from a mineral deposit in Mexico.

In all of the examples we define the regularization functional $R(m)$ to be the discretized equivalent of

$$\mathcal{R}(m) = \int_\Omega (|\nabla(m - m_{\mathrm{ref}})|^2)\, dv + \gamma \int_\Omega (m - m_{\mathrm{ref}})^2\, dv, \tag{7.33}$$

where $\gamma = 10^{-5}$.

7.1 A generic inversion

A square loop with dimensions of 130 × 130 m is located on 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} and 10^{-3} s. The earth model is made of one conductive block of 10 Ωm and a resistive block of 1000 Ωm buried in a half-space of 100 Ωm. There are 100 receivers on the surface inside the transmitter loop and five component fields ($\mathbf{H}_x$, $\mathbf{H}_y$, $\mathbf{H}_z$, $\mathbf{E}_x$ and $\mathbf{E}_y$) are measured. Gaussian random noise has been added to the 16 000 data. The standard deviation of the noise was one percent of the data value plus a floor. Fitting the data to this small error makes the test relatively difficult.

The problem is discretized using 42^3 cells and the number of unknowns ($\mathbf{A}$, ϕ amd $\mathbf{H}$ for all space–time and m) is roughly 16 million. The regularization functional is that used in eq. (33). The initial model and initial reference model were set equal to the true half-space.

To solve the optimization problem, we have used the iterative Tikhonov algorithm along with L-BFGS method to generate a preconditioner for the matrix system (9). We allowed four L-BFGS iterations per-β before making a decision about whether to change β. The

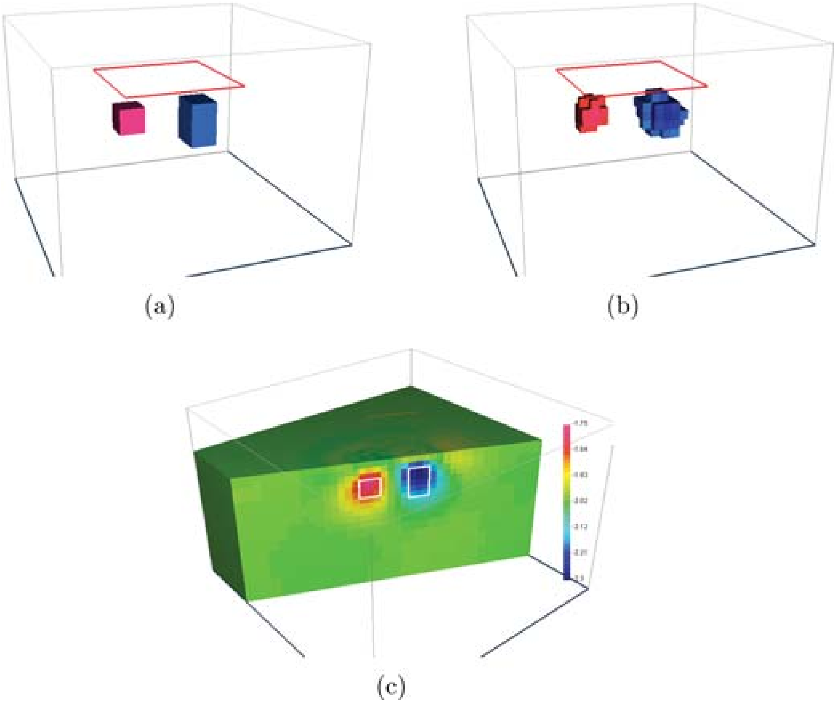

Figure 1. The true model and source loop geometry are shown in (a). A volume rendered image of the recovered conductivity is shown in (b). A cross-section of the recovered conductivity is shown in (c). The location of the two prisms are indicated on that cross-section.

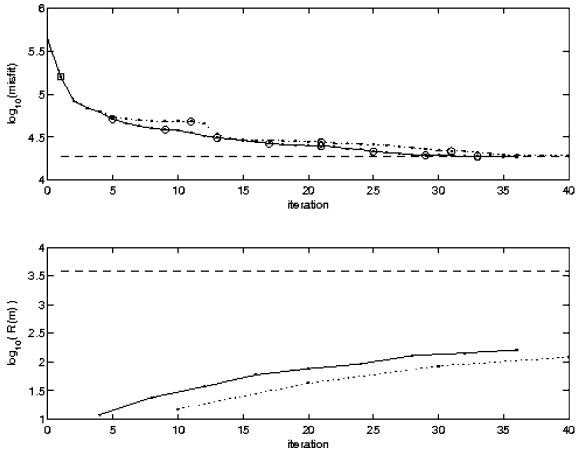

Figure 2. The misfit as a function of iteration is shown in the top panel. The solid line corresponds to an inversion in which there are four L-BFGS iterations per β value. The dashed line corresponds to 10 iterations per β. The circles indicate a changing value of β. The straight line at the bottom indicates the achieved misfit. The values of the regularization functional corresponding to the two inversions are shown in the bottom panel. The straight line indicates the value of $R(m)$ evaluated with the true model.

reference model was updated to be equal to the current model each time β changed. The true and recovered conductivity models are shown in Fig. 1. The conductive and resistive blocks are in the correct locations but, as is usual in smooth model inversions of blocky structures, the recovered conductivities of the target bodies are somewhat smaller than the true values and edges are smoothed. Because of the small amount of noise added, and because the data have been fit so well, there is little visual difference between observed and predicted data. We have therefore, not plotted these images.

The numerical progress of the inversion is summarized in Fig. 2. A misfit of 19 000 is achieved after 40 iterations and a cooling schedule that saw β decreased by five orders of magnitude in eight steps.

The inversion software is coded in Fortran 90 and was run on an Opteron 244 processor. It took roughly 64 h to complete.

In an inversion involving a cooling strategy there are tradeoffs regarding the rate at which the regularizing parameter β is reduced and how many iterations are carried out at each β. The goal is to proceed cautiously so that the solution trajectory stays close to the classical Tikhonov curve. Our situation is slightly more complicated because we are using an iterated Tikhonov approach where the reference model is changing. To obtain some confidence that the solution in Fig. 1 is a good minimizer of the initial objective function (33), we repeat the

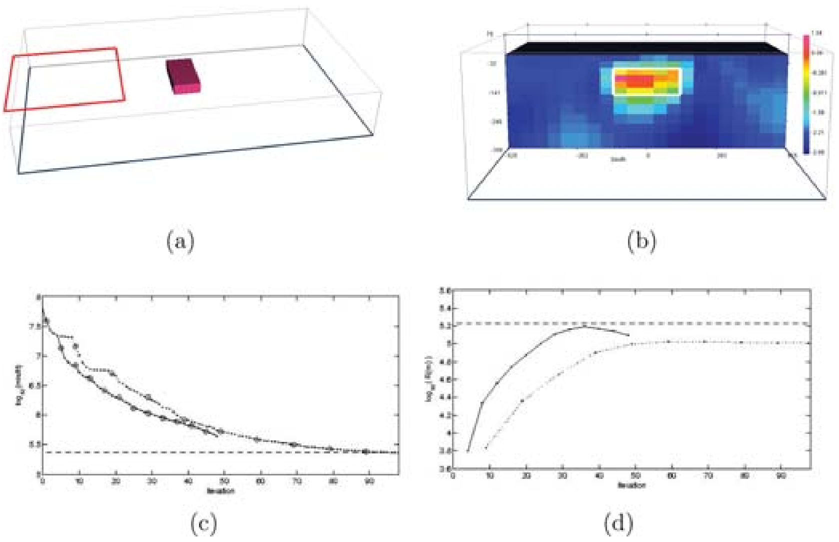

Figure 3. The true model and source loop geometry are shown in (a). A cross-section of the recovered conductivity, as well as the location of the prism, is shown in (b). The misfit as a function of iteration is shown in (c). The straight dashed line at the bottom indicates the final misfit of 231 000 so the data are fit, on average to about 1.2 per cent. The solid and dashed curves, respectively, correspond to a maximum of 4 and 10 iterations per β. The value of the model regularization functional is shown in (d). The straight line at the top of the plot shows $R(m)$ evaluated with the true model. Both recovered models have an $R(m)$ that is substantially below that and hence are valid candidates for a solution to our optimization problem.

inversion but extend the number of iterations per β from four to ten. The results are displayed in Fig 2. A smaller misfit is obtainable with the same set of $\beta's$ but the improvement is slight and differences in the final model are not readily visible by eye. This is further quantified in Fig 2 where $R(m)$ in (33) is evaluated as a function of iteration. In this computation the reference model remains equal to the initial half-space value. The closeness of the two curves suggests that we are doing a reasonably good job at finding a solution that minimizes our initial objective function. This is supported by the fact that $R(m)$, evaluated with the true model, is 3925. This is more than an order of magnitude larger than that achieved by our inversion results.

7.2 Corrective source for reducing the volume

The following example emulates a typical mineral exploration survey in which a large transmitter loop is deployed on the surface and the area of investigation lies outside the loop. The goal is to find a conducting target. The 1000 × 1000 m transmitter loop and the conductivity model are shown in Fig. 3. The transmitter signal is a step-off and three component magnetic field data are recorded at 31 equally spaced logarithmic times between 10^{-4} and 10^{-2} s. The data were contaminated with one percent Gaussian noise.

Forward modelling the data required using a large volume that included the transmitter (thus we avoided the ‘inverse crime’ here of inverting the data with the same method used to generate it). The resultant mesh was too large to be used in the inversion and also the extended volume served little purpose since the data and the volume of interest is quite confined. A smaller mesh encompassing the volume of interest was generated using 50 × 50 × 25 m cells in the core region that included the survey area. Padding cells then extended the final inversion volume such that the left boundary terminated inside the transmitter loop. The total number of cells in the inversion volume was 28 672. The background fields for a half-space, whose conductivity was equal to the true half-space, were used to generate the corrective sources at each time.

This inversion was run on an Pentium four 3GHz and it took 84 h to complete.

A vertical cross-section of the recovered model is shown in Fig. 3. The misfit and model regularization curves as a function of iteration are also shown. Again the inversion has been run twice, with the maximum number of iterations per β being 4 and 10. In this case, allowing more iterations per regularization value resulted in a reduced final misfit and better overall convergence.

7.3 Numerical experiments with extended waveforms

A surface loop survey is carried out over a conductive prism of 8 $\mathrm{S\,m^{-1}}$. The earth model and source geometry are shown in Fig. 5. The waveform is a half-sinusoid followed by an off-time. The frequency of the transmitter is $f = 1000$ Hz and thus the total waveform, comprised of equal length of on- and off-time segments, is $T = 0.001$ s. Because of its high conductivity and size, the time constant for the target is approximately $\tau = 0.004$ s which is greater than T. Multiple waveforms are needed and in Fig. 4 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 off-time. The number of time samples was 193. Also in Fig. 4, we plot the decay curves for H_z for the first four cycles. There is a substantial difference between successive decays but approximate steady state is being approached by the end of the fourth cycle.

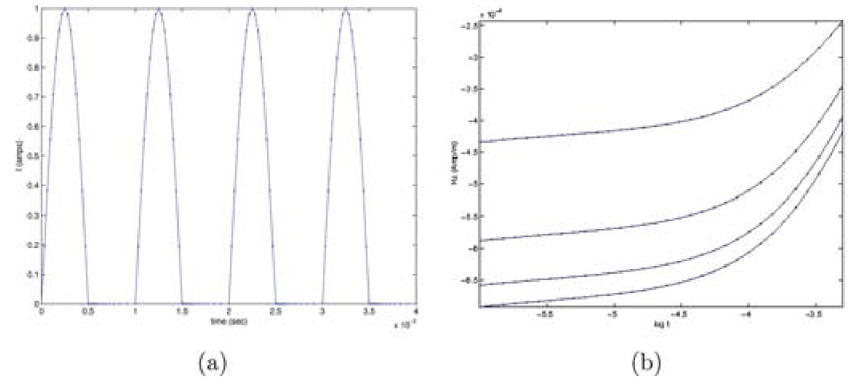

Figure 4. Four cycles of the waveform are shown. Time samplings, denoted by dots, are uniform during the on-time and logarithmic during the off-time. Data at the beginning of the off-time for the fourth waveform will be inverted. Response curves for H_z at a station near the centre of the loop are shown in (b). The top response is the decay after the first pulse; the bottom response is after the fourth pulse.

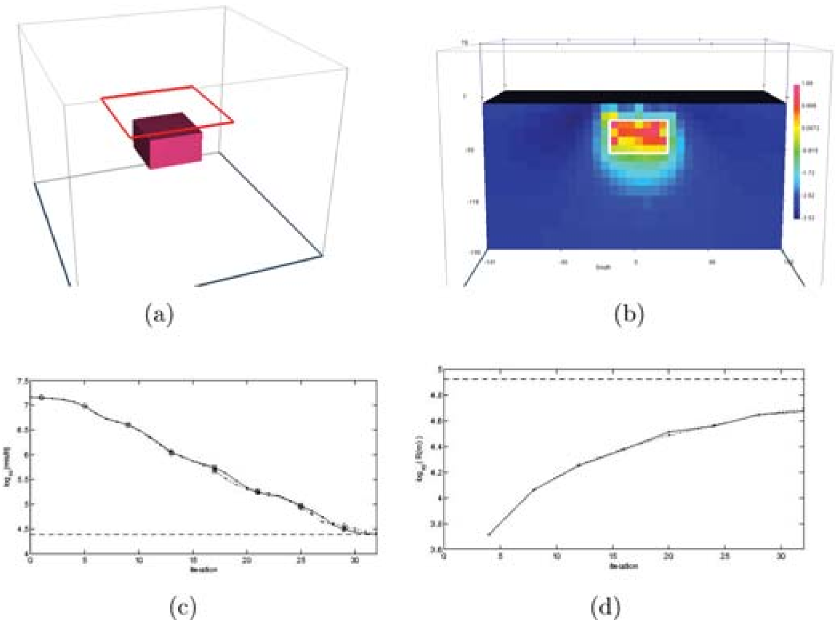

Figure 5. The conductivity model and source geometry is shown in (a). Data are acquired at 100 stations in the interior of the loop. A cross-section of the recovered model is shown in (b). The misfit and model norm curves are, respectively, shown in (c) and (d). The solid lines correspond to an inversion in which the forward modelling began at time zero. The dashed lines correspond to starting the modelling at the beginning of the fourth pulse but using the times from a previous model.

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 onset of the fourth waveform. One processor is used to run the forward modelling from (0, t_{e-1}) and it stores the field u_{e-1}. This field is updated every time 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 the fields at t_{e-1} with those obtained by carrying out a full modelling. The differences, as illustrated in Fig. 5 are minimal. Yet the time savings were important. We have run the inversion on an Opteron 244 processor. By having the first three periods of the transmitter pulse modelled by the second processor, the inversion time was reduced by roughly a factor of 3 to 73.5 hr while the run on a single (same) processor took 231 h. For completeness we plot the regularization functional in Fig. 5 as well as a cross-section of the recovered model.

8 FIELD EXAMPLE

In our final example, we invert data from a mineral deposit in Mexico. San Nicolas is a Cu–Zn massive sulphide deposit located in central Mexico in the state of Zacatecas. The sulphide is a high conductor compared to most of the geologic units but the overburden and some of the tertiary volcanic breccia, have a conductivity in the range of the sulphide. The details about the geology, inversion and interpretation are

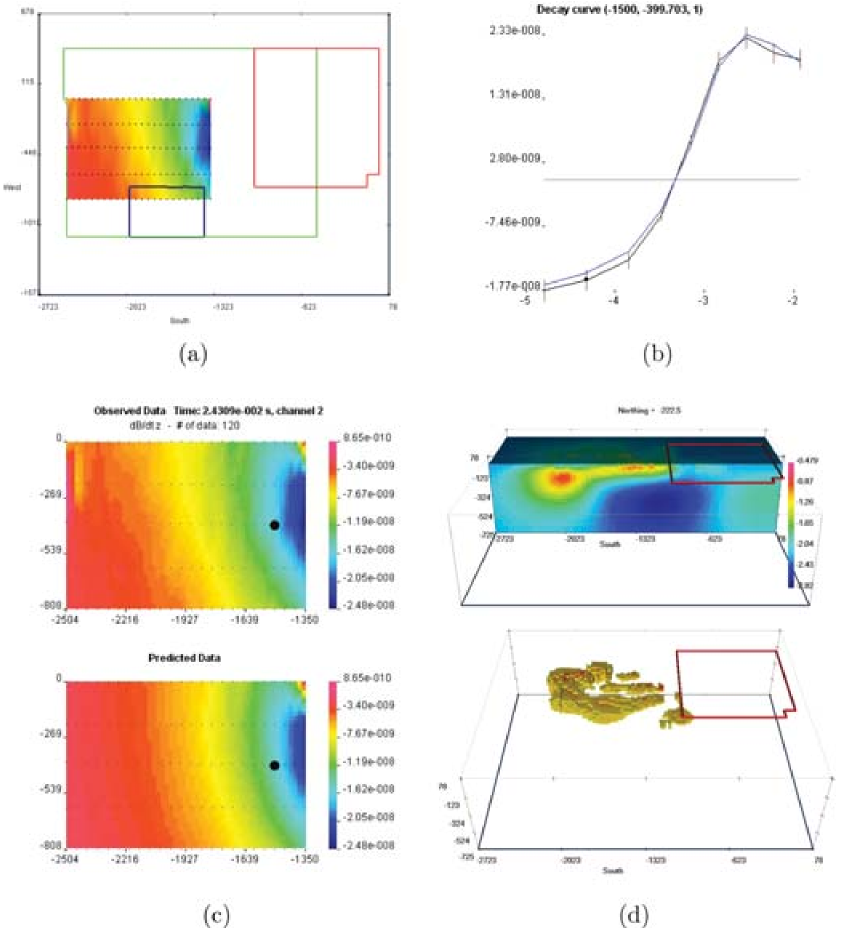

Figure 6. Survey geometry for the San Nicolas UTEM survey. Transmitter loops are shown along with data locations. The coloured area shows the location for data from the eastern most loop that is inverted here. Decay curves showing the predicted and observed data for the nine inverted UTEM time channels are shown in (b). The data are from a selected location, shown by a black dot in panel (c). Error bars are plotted. Panel (c) shows the observed and predicted data for the second UTEM time channel. A cross-section and a volume rendered image of the recovered conductivity are presented in (d).

presented elsewhere (Napier Oldenburg, in preparation). The example is included here primarily to show the application of our algorithm to a larger scale problem involving field data.

A UTEM survey was conducted over the San Nicolas deposit in December of 1998. The UTEM system, described by West *et al.* (1984), uses a continuous on-time loop transmitter with a sawtooth waveform and coil receivers which measure the time derivative of the magnetic field. At San Nicolas the UTEM survey consisted of three large loops as shown in Fig. 6. Vertical components of the field were recorded on north–south and east–west lines over the deposit.

We now invert the data from the eastern-most transmitter. The central volume of the mesh shown in Fig. 6(a) was discretized into 25 × 25 m cells with a variable thickness, from 15 m near the surface to 25 m at depth. The mesh then extended out from the core region several kilometres in each direction. The final volume consisted of 349 440 cells. The inversion used 38 time steps over $1\frac{1}{4}$ UTEM cycles. The first $3/4$ of a cycle were uniformly sampled in time. The waveform was then densely sampled in log-time on the next half cycle. Sampling on that ramp began at 1.5 decades before the first datum. The frequency of the transmitter was 30.974 Hz and the first and earliest and latest time channels were, respectively, 47 μs and 12 ms. The inversion functional is that given in (33) and the initial and reference models were equal to $200\Omega - m$. Data errors assigned were equal to 5 per cent of the datum plus a floor value equal to 3 per cent of the median value of each time channel. In total there were 1082 data. The initial and final β values were 1.0 and .0016 and the inversion used 17 iterations to achieve the desired misfit of 1082. It took 19 days on a single Opteron 244 processor. A vertical cross-section and a volume rendered image of the recovered conductivity are shown in Fig. 6. Also plotted there are the observed and predicted data for the second time channel and a decay curve for a selected location. These illustrate the good agreement between the observed and predicted data.

The field data from all three loops were simultaneously inverted but with a model that had only 90 000 cells. The results were quite comparable but, as might be expected with more transmitters, there are differences which might have significant geologic implications. These are being investigated in the paper by Napier Oldenburg, in preparation.

9 SUMMARY

We have shown how time domain electromagnetic data can be inverted to recover a 3-D electrical conductivity model. The problem is complex and computationally difficult. We therefore, used a combination of techniques to solve this problem.

(1) We precondition the Gauss–Newton iteration with approximate Hessians obtained using the BFGS method.
(2) We use the iterated Tikhonov that allow us to work with larger regularization parameters.
(3) We modify the L-BFGS method to deal with changing regularization parameters.
(4) We introduce a source correction procedure that allows us to reduce the spatial volume of the modelling and inversion region.

We used both synthetic and field data sets to test our code. The results verify that our algorithm is working satisfactorily and the next step is to apply and further develop the techniques in field applications.

ACKNOWLEDGMENTS

This research is supported in part by the DOE under grant DE-FG02-05ER25696, and the NSF under grant CCF-0427094.

REFERENCES

Aruliah, D. & Ascher, U., 2002. Multigrid preconditioning for time harmonic Maxwell's equations in 3D, *SIAM J. on Scientific Computing,* **24,** 702–718.

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

Ascher, U. & Haber, E., 2003. A multigrid method for distributed parameter estimation problems, *ETNA,* **15,** 1–15.

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

Biros, G. & Ghattas, O., 2005. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization Parts I, II, *SIAM J. on Scientific Computing,* **27,** 687–712.

Borcea, L., Berryman, J.G. & Papanicolaou, G.C., 1996. High-contrast impedance tomography, *Inverse Problems,* **12,** 835–858.

Cheney, M., Isaacson, D. & Newell, J.C., 1999. Electrical impedance tomography, *SIAM Review,* **41,** 85–101.

Clarenz, U., Droske, M. & Rumpf, M., 2002. Towards fast non-rigid registration, in *Inverse Problems, Image Analysis and Medical Imaging, AMS Special Session Interaction of Inverse Problems and Image Analysis*, Vol. **313,** pp. 67–84. AMS.

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

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

Engl, H.W., Hanke, M. & Neubauer, A., 1996. *Regularization of Inverse Problems*, Kluwer, Berlin.

Farquharson, C. & Oldenburg, D., 1998. Non-linear inversion using general measures of data misfit and model structure, *Geophys. J.,* **134,** 213–227.

Gilles, L., Vogel, C.R. & Bardsley, J.M., 2002. Computational methods for a large-scale inverse problem arising in atmospheric optics, *Inverse Problems,* **18,** 237–252.

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

Haber, E., 2005. Quasi–Newton methods methods for large scale electromagnetic inverse problems, *Inverse Problems,* **21,** 305–317.

Haber, E., 2006. A parallel method for large scale time domain electromagnetic inverse problems, *Int. J. Compt. Math.,* in press.

Haber, E. & Ascher, U., 2001. Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients, *SIAM J. Sci. Comput.,* **22,** 1943–1961.

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

Haber, E. & Oldenburg, D., 2001. A GCV based methods for nonlinear inverse problem, *Comput. Geosci.,* **4,** 41–63.

Haber, E., Ascher, U. & Oldenburg, D., 1999. Solution of the 3D electromagntic inverse problem, in *Third Int. Symp. on 3D Electromagnetics*, Salt Lake City, Oct.

Haber, E., Ascher, U. & Oldenburg, D., 2000. On optimization techniques for solving nonlinear inverse problems, *Inverse problems,* **16,** 1263–1280.

Haber, E., Oldenburg, D. & Ascher, U., 2000. Modelling 3D maxwell's equations with non-continuous coefficients, in *SEG*, Calgary.

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

Haber, E., Ascher, U. & Oldenburg, D., 2004. Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach, *Geophysics,* **69,** 1216–1228 n5.

Hansen, C., 1998. *Rank Deficient and Ill-Posed Problems*, SIAM, Philadelphia.

Hanke, M. & Groetsch, C., 1998. Nonstationary iterated Tikhonov regularization, *J. Optim. Theory and Applications,* **98,** 37–53.

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

Kelley, C.T., 1999. *Iterative Methods for Optimization*, SIAM, Philadelphia.

Napier, S. & Oldenburg, D., *Workflows for practical inversions of time domain electromagnetic data*, submitted.

Newman, G.A. & Boggs, P., 2004. Solution accelerators for large-scale three-dimensional electrmagnetic inverse problems, *Inverse Problems,* **20,** S151–S170.

Newman, G.A. & Commer, M., 2005. New advances in three dimensional transient electromagnetic inversion, *Geophys. J. Int.,* **160,** 5–32.

Nocedal, J. & Wright, S., 1999. *Numerical Optimization*, Springer, New York.

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

Saad, Y., 1993. *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, New York.

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

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

Vogel, C., 2001. *Computational Methods for Inverse Problems*, SIAM, Philadelphia.

Wahba, G., 1998. *Spline Models for Observational Data*, SIAM, Philadelphia.

Wang, T. & Hohmann, G.W., 1993. A finite-difference, time-domain solution for three-dimensional electromagnetic modelling, *Geophysics,* **58,** 797–809.

Wang, T., Oristaglio, M., Tripp, M. & Hohmann, G.W., 1994. Inversion of diffusive transient electromagnetic data by a conjugate gradient method, *Radio Sci.,* **29,** 1143–1156.

Ward, S.H. & Hohmann, G.W., 1988. Electromagnetic theory for geophysical applications, *Electromagn. Methods Appl. Geophys.,* **1,** 131–311. Soc. Expl. Geophys.

West, G.F., Macnae, J.C. & Lamontagne, Y., 1984. A time domain EM system measuring the step response of the ground *Geophysics,* **26,** 1010–1026.

APPENDIX A: REGULARIZATION EFFECT OF TIME STEPPING

Assume that we are given an ill-posed problem of the form

$$F[m] = d + \epsilon$$

and we would like to obtain a solution m that fits the data. Since the problem is ill-posed we need to use regularization, that is, to obtain a model that fits the data to some extent and is not sensitive to small changes in the data. To that extent, we embed the solution m in a family of models $m(t)$ where t is a regularization parameter. As can be found in Vogel (2001) any regularization technique has the following two criteria

C1 Let $m(t_1)$ and $m(t_2)$ be two solutions where $t_1 < t_2$ then $\|m(t_1)\|_W \leq \|m(t_2)\|_W$ and $\|F[m(t_2)] - d\| \leq \|F[m(t_1)] - d\|$ where $\|\cdot\|_W$ is some weighted norm.

C2 $\lim_{t\to\infty} m(t) = m^*$ where $F[m^*] = d$.

It is well known that Tikhonov regularization fits this criteria. To see that we write Tikhonov regularization as the solution of the optimization problem

$$\min \quad \phi(m) = \frac{1}{2}\|F[m] - d\|^2 + \frac{1}{2t}\|m\|_W^2.$$

Then, assuming ϕ is convex we obtain better data fit for larger t and obtain a perfect fit for $t \to \infty$. Furthermore, for $t_1 < t_2$ we have $\|m(t_1)\|_W \leq \|m(t_2)\|_W$.

In the case that $F[m] = Am$ (i.e. the forward problem is linear), and simple quadratic norm, assuming the Singular Value Decomposition of A as

$$A = U\Lambda V^\top$$

it is possible to write the solution as

$$m(t) = \sum_i f_T(\lambda_i\,; t)\frac{u_i^\top d}{\lambda_i},$$

where $f_T(\lambda;t)$ is referred to as Tikhonov filter function

$$f_T(\lambda, t) = \frac{\lambda^2}{\lambda^2 + t^{-1}}.$$

For $\lambda \gg t$ we have $f_T \approx 0$ and for $\lambda \ll t$ we have $f_T \approx 1$. Thus, the effect of the filter is to damp the vectors that are associated with small singular values while leaving the larger ones untouched.

The Tikhonov regularization, may be the most known regularization technique. However, a different regularization technique can be obtained by embedding the solution into a dynamical system. Consider first the case of the linear forward modelling and define the regularized solution as a solution of the dynamical system which is described by the following Ordinary Differential Equation

$$\frac{dm}{dt} = A^\top(Am - d).$$

Using the SVD of A it is easily verified that the solution of this ODE is

$$m(t) = \sum_i f_D(\lambda_i; t)\frac{u_i^\top d}{\lambda_i},$$

where

$$f_D(\lambda; t) = 1 - \mathrm{e}^{-\lambda^2 t}.$$

We see that the above approach yields filters (and hence solutions) that are very similar to the Tikhonov filter functions. More details about this type of regularization can be found in Hanke & Groetsch (1998).

For non-linear problems the dynamical system is a non-linear one and can be written as

$$\frac{dm}{dt} = J(m)^\top(F[m] - d).$$

In this case there is no close solution however, similar to the case of Tikhonov regularization, assuming convexity it is possible to show that the two conditions above apply.

To solve the above non-linear dynamical system one requires a numerical strategy. In the paper, we have used the implicit backward Euler approach.