GEOPHYSICS, VOL. 78, NO. 1 (JANUARY-FEBRUARY 2013); P. E47–E57, 12 FIGS., 3 TABLES.
10.1190/GEO2012-0131.1

Three dimensional inversion of multisource time domain electromagnetic data

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

ABSTRACT

We present a 3D inversion methodology for multisource time-domain electromagnetic data. The forward model consists of Maxwell's equations in time where the permeability is fixed but electrical conductivity can be highly discontinuous. The goal of the inversion is to recover the conductivity-given measurements of the electric and/or magnetic fields. The availability of matrix-factorization software and high-performance computing has allowed us to solve the 3D time domain EM problem using direct solvers. This is particularly advantageous when data from many transmitters and over many decades are available. We first formulate Maxwell's equations in terms of the magnetic field, $\vec{H}$. The problem is then discretized using a finite volume technique in space and backward Euler in time. The forward operator is symmetric positive definite and a Cholesky decomposition can be performed with the work distributed over an array of processors. The forward modeling is quickly carried out using the factored operator. Time savings are considerable and they make 3D inversion of large ground or airborne data sets feasible. This is illustrated by using synthetic examples and by inverting a multisource UTEM field data set acquired at San Nicolás, which is a massive sulfide deposit in Mexico.

INTRODUCTION

Electrical conductivity is a diagnostic physical property for many problems in resource exploration, environmental, and engineering applications. A variety of electromagnetic surveys can be carried out to delineate the subsurface conductivity, and a generic scenario is shown in Figure 1. Sources can be grounded or inductive, located inside, on, or above the earth's surface, and data $\vec{E}, \vec{H}$, $\partial\vec{B}/\partial t$ can be measured at any location in the 3D volume. The waveform in the transmitter can be continuous "on-time" or have an "off-time" so that data can be collected when no current flows through the transmitter. To simulate time domain data, Maxwell's equations are either directly time-stepped, or solutions in the frequency domain are Fourier transformed. The latter is often referred to as a spectral method in time. Here, we adopt the time-stepping formulation but our approach can be easily carried over to the spectral method. Our goal is to develop a flexible and practical 3D algorithm to invert the time data to recover the electrical conductivity with the assumption that the magnetic permeability is fixed.

Recovering meaningful 3D distributions of any physical property requires that the data contain adequate information. This is facilitated by measuring multiple components of the electromagnetic fields, but it also requires that the earth is illuminated by many different sources. Airborne EM surveys typically acquire data from thousands of transmitters, and even ground surveys employ multiple transmitters. Inverting these data to find the causative 3D electrical conductivity has been an ongoing challenge. In previous research (Haber et al., 2007), we developed an inversion algorithm that allowed data from a single, or a very few, transmitters to be inverted. Unfortunately, the computational demands of that algorithm were too large to invert typical ground or airborne surveys acquired from many source locations. Other recent work on time domain problems is presented in Zaslavsky et al. (2010), and it suffers from a similar drawback when the number of sources is large. Simulating data from multisources can be computationally onerous because each transmitter requires that Maxwell's equations be solved with a specific right side. In our previous work, the forward problem was solved using an iterative Krylov method, and the computation time increased linearly with the number of transmitter locations. However, as we discuss here, for a large number of sources (right sides), efficiency can be increased significantly if the forward modeling matrix is factored. Factorizing involves large

Manuscript received by the Editor 15 April 2012 ; published online 20 December 2012.
[1]University of British Columbia, Geophysical Inversion Facility, Department of Earth and Ocean Sciences, Vancouver, Canada. E-mail: doug@eos.ubc.ca; shekht@eos.ubc.ca.
[2]University of British Columbia, Department of Mathematics, Department of Earth and Ocean Sciences, Vancouver, Canada. E-mail: haber@math.ubc.ca.

computations and significant memory requirements, but once this is accomplished, solving the factored system with a different right side proceeds quickly (Oldenburg et al., 2008). The idea of decomposing the matrix system and solving many right sides for different sources is not new (Dey and Morrison, 1979; Pratt, 1999), and small problems have been solved in this manner. However, the matrices for 3D TEM problems have generally been considered to be too large to be amenable for this approach. Over the last decade, however, advances in mathematics and computational science have resulted in factorization algorithms that can be implemented on large-scale computing systems (Demmel et al., 1999; Amestoy et al., 2001). Here, we advocate the use of such codes and, in particular, we use the factorization code MUMPS and distribute the computation over many different processors. This enables us to solve large-scale problems with multiple sources in an efficient manner.

Our paper is divided into five sections. First, we introduce the forward problem. Its components include our reformulation of Maxwell's equations for the $\vec{H}$ field, the discretized equations in space and time, and practical strategies for implementing the decomposition. Then, the inverse problem is presented and synthetic examples are shown. Using synthetic examples, we compare the computation time and memory cost for the forward and inverse problems when using direct and iterative solvers. Next, we invert ground-based UTEM field data over a massive sulfide at the San Nicolás deposit. We conclude with a discussion.

MAXWELL'S EQUATIONS IN THE TIME DOMAIN

The forward model consists of Maxwell's equations in time, where the permeability is fixed but electrical conductivity can be highly discontinuous. For the case where we have N transmitters, we write Maxwell's equations as

$$\nabla \times \vec{E}^{(j)} + \mu \frac{\partial \vec{H}^{(j)}}{\partial t} = 0, \tag{1a}$$

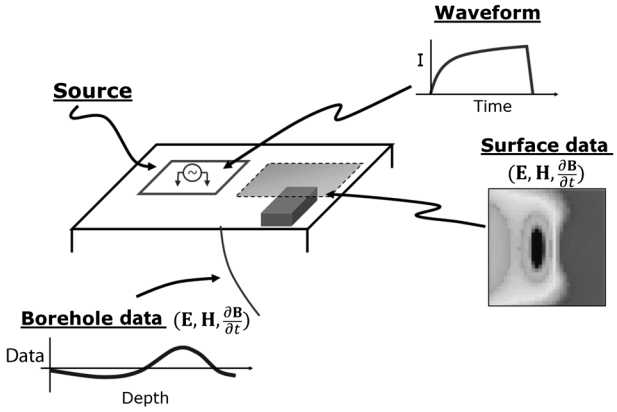

Figure 1. A generic electromagnetic experiment. The source can be grounded or inductive, and it may be located in the air, on the surface, or inside the conductive volume. The waveform is arbitrary. The time-varying current in the transmitter induces electromagnetic fields in the earth. Data $\vec{E}, \vec{H}, d\vec{B}/dt$ are recorded in the air, on the surface, or inside the conducting medium as might be the case in marine EM surveys, or in boreholes.

$$\nabla \times \vec{H}^{(j)} - \sigma \vec{E}^{(j)} - \epsilon \frac{\partial \vec{E}^{(j)}}{\partial t} = \vec{s}_r^{(j)}(t) \quad j = 1, \ldots, N, \tag{1b}$$

over a domain $\Omega \times [0, t_f]$, where $\vec{E}^{(j)}$ and $\vec{H}^{(j)}$ are the electric and magnetic fields that correspond to the source $\vec{s}_r^{(j)}$, μ is the permeability, σ is the conductivity, ϵ is the permittivity, and Ω denotes the spatial domain. The equations are given with boundary and initial conditions

$$\vec{n} \times \vec{H}^{(j)} = 0, \tag{1c}$$

$$\vec{H}(0, x) = \vec{H}_0^{(j)}, \tag{1d}$$

$$\vec{E}(0, x) = 0, \tag{1e}$$

although other boundary and initial conditions could be used. In the above, $\vec{n}$ is the unit normal on the boundary and equations 1d and 1e are the initial conditions for $t = 0$.

Solving the forward problem

In this subsection, we briefly discuss our approach for the solution of the forward problem. An elaborate discussion can be found in Newman and Alumbaugh (1995), Hyman and Shashkov (1998), Haber et al. (2000a, 2004), Aruliah and Ascher (2003), and Hu et al. (2006).

For ease of notation, we consider a single source and the index i refers to the time step. Maxwell's equations in conductive media are highly stiff, and therefore we use the backward Euler discretization in time with step size δt (Ascher and Petzold, 1998; Ascher, 2010). We obtain the following semidiscretized partial differential equation for the fields at each time step

$$\nabla \times \vec{E}^{i+1} + \mu \frac{\vec{H}^{i+1} - \vec{H}^{i}}{\delta t} = 0, \tag{2}$$

$$\nabla \times \vec{H}^{i+1} - \sigma \vec{E}^{i+1} - \epsilon \frac{\vec{E}^{i+1} - \vec{E}^{i}}{\delta t} = \vec{s}_r^{i+1}. \tag{3}$$

To discretize over the spatial domain, we use a staggered grid for $\vec{E}$ and $\vec{H}$. We employ the traditional Yee's method (Yee, 1966), in which $\vec{H}$ is discretized on edges and $\vec{E}$ on faces as demonstrated in Figure 2.

With the finite volume method, the resultant matrix system is

$$C^{\top} \vec{E}^{i+1} + \alpha M \vec{H}^{i+1} = \alpha M \vec{H}^{i}, \tag{4a}$$

$$C \vec{H}^{i+1} - S \vec{E}^{i+1} - \alpha \Upsilon \vec{E}^{i+1} = \vec{s}_r^{i+1} - \alpha \Upsilon \vec{E}^{i}, \tag{4b}$$

where C and $C^{\top}$ are curl operators, respectively, going from edges to faces and from faces to edges, S and Υ are, respectively, the (harmonic) averaged conductivity and electrical permittivity, M is the arithmetically averaged magnetic permeability, $\alpha = \frac{1}{\delta t}$, and $\vec{s}_r^{i+1}$ is the volumetric integral of the original current source.

Eliminating the electric field $\vec{E}^{i+1}$ from Maxwell's equations, we obtain an equation for the magnetic field $\vec{H}^{i+1}$

$$(C^\top \Gamma^{-1} C + \alpha M)\vec{H}^{i+1} = C^\top \Gamma^{-1}\vec{s}_r^{\,i+1} - \alpha C^\top \Gamma^{-1}\Upsilon\vec{E}^i + \alpha M\vec{H}^i, \tag{5}$$

where $\Gamma = S + \alpha\Upsilon$. Once the magnetic fields have been computed, the electric fields are obtained from

$$\vec{E}^{i+1} = \Gamma^{-1}C\vec{H}^{i+1} - \Gamma^{-1}\vec{s}_r^{\,i+1} + \alpha\Gamma^{-1}\Upsilon\vec{E}^i. \tag{6}$$

Equation 5 can be written as

$$\tilde{A}(\sigma, \delta t)\vec{H}^{i+1} = \text{rhs}. \tag{7}$$

Using finite volume or finite integration techniques (Madden and Mackie, 1989; Bossavit, 1998), we obtain a symmetric positive definite forward modeling matrix $\tilde{A}$. As a consequence, we can use the Cholesky decomposition and write

$$\tilde{A} = LL^\top.$$

We now make an important observation which motivates our approach. We note that by using the same time step δt, the linear system 7 is identical for all times and all sources. Thus, a single factorization can be used to solve all the linear systems.

The system 5 must be solved with initial conditions. If the system starts from a null state, then all fields are zero. If the system starts from a steady state condition, say with a constant current flowing in a loop, then the steady state fields are needed. This requires solution of the system

$$C^\top S^{-1} C\vec{H} = C^\top S^{-1}\vec{s}_r. \tag{8}$$

This system is singular and consistent. It can be made stable by requiring that (the discrete analog of) $\nabla \cdot \mu H = 0$. Note that, on a staggered grid, the divergence operator is the (negative) transpose of the discrete gradient operator G (see, for example, Hyman and Shashkov, 1999), and given a material property matrix M (the average of μ on the cell faces), we obtain that

$$G^\top M\vec{H} = 0.$$

Multiplying from the left by $GM^{-1}\Gamma^{-1}$ and adding to equation 8 we obtain

$$C^\top S^{-1} C\vec{H} + GM^{-1}\Gamma^{-1}G^\top M\vec{H} = C^\top S^{-1}\vec{s}_r. \tag{9}$$

This is a symmetric system which ensures that $\nabla \cdot \mu\vec{H} = 0$. We choose the matrix Γ such that in regions with fixed material properties the differential operator is reduced to a vector Laplacian (see Haber and Ascher, 2001).

The final system to be solved incorporates all time steps and we can write the forward modeling generically as $A(m)u = q$ where A is the forward modeling matrix, u are the fields, and q is the right side which contains the source terms. A is a large bidiagonal matrix of the form

$$A = \begin{pmatrix} \tilde{A}(m) & \cdots & & 0 \\ B & \tilde{A}(m) & \cdots & \\ & \cdots & \cdots & \\ 0 & & B & \tilde{A}(m) \end{pmatrix}, \tag{10}$$

where

$$B^{i+1} = \alpha C^\top \Gamma^{-1}\Upsilon\vec{E}^i + \alpha M\vec{H}^i \quad q^i = C^\top \Gamma^{-1}E^i\vec{s}_r.$$

The first solution, $\tilde{A}(m)u^{(1)} = q^{(1)}$, establishes the initial fields. If the initial state is a constant current, then this recovers the DC fields. Subsequent solutions $\tilde{A}(m)u^{(i+1)} = q^{(i+1)} - B^{(i+1)}u^i$ produce the fields marching in time.

Matrix factorization is an expensive computational process. If the number of sources and time steps is small, then iterative methods for solving the forward problem can be superior. However, when the same forward-modeling matrix needs to be inverted many times, the decomposition will be greatly superior to iterative techniques. The benefits of decomposition are enhanced for the inverse problem because the same factorization can be used for the computation of the gradient as well as for the solution of the linear system which arises at each Gauss-Newton iteration.

Software for matrix factorizations

In recent years, there has been a growing effort to obtain scalable matrix factorizations on parallel machines. The mathematical progress made on this problem, coupled with the continued increased power of computer workstations, allows 3D EM problems to be solved with this technology. We use the package Multifrontal Massively Parallel Solver (MUMPS) developed by the CERFACS group (Amestoy et al., 2001). Other packages can also be used (Demmel et al., 1999), however our experience with MUMPS was favorable. MUMPS is a package for solving systems of linear equations of the form $Ax = b$, where the matrix A is sparse, not symmetric, symmetric positive definite, or general symmetric. MUMPS uses a multifrontal technique which is a direct method based on either the LU or the $LDL^\top$ factorization of the matrix. It exploits parallelism arising from sparsity in the matrix A and from dense factorizations kernels. The main features of the MUMPS package include

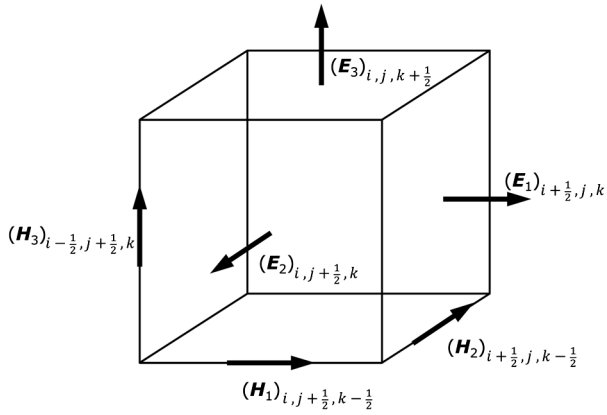

Figure 2. Staggered discretization in 3D: E is defined on the faces and H is on the edges.

the solution of the transposed system, input of the matrix in assembled format (distributed or centralized) or elemental format, error analysis, iterative refinement, scaling of the original matrix, and return of a Schur complement matrix. Finally, MUMPS is available in various arithmetics (real or complex, single or double precision). The software is written in Fortran 90. The parallel version of MUMPS requires MPI for message passing and makes use of the BLAS, BLACS, and ScaLAPACK libraries. We have tested MUMPS on a cluster of PCs under Linux. In our setting, MUMPS distributes the work tasks among the processors, but an identified processor (the host) is required to perform most of the analysis phase, distribute the incoming matrix to the other processors (slaves) in the case where the matrix is centralized, and collect the solution.

Comparing direct and iterative solvers

To obtain insight regarding the advantages of the decomposition, we consider a generic problem in airborne geophysics. The transmitter is a square loop, of 20 m on a side, flown at 50 m elevation. Three components of $d\vec{B}/dt$ are acquired in the center of the loop. The waveform is a step-off and 35 time channels, equispaced in logarithmic time between 10^{-5} and 10^{-2} s, are measured. Data are collected every 20 m along 20 east–west lines and there are 1000 transmitter locations. We discretize the volume with a regular structured grid of n^3 cells.

The current is a step-off and hence, in the numerical forward modeling, the initial fields are steady state. Although the first observation time is at 10^{-5} s, our time stepping must start before that; typically, we begin about a decade before the first observation time. The rationale for this is that backward Euler (BE) is a highly stiff time integrator. One important, well-known feature of BE is that it "skips" fast decaying modes and, without significant loss of accuracy, captures the slow-decaying modes. (Ascher and Petzold, 1998). Thus, displacement currents and modes that have early decay can be skipped without significant loss of accuracy. This does not mean that accuracy for the slow modes is obtained without any step control and one needs to verify that the step length is sufficiently small to accurately capture the diffusion part of the system.

The total time to carry out a forward modeling is

$$T_{\text{total}} = [N_{TX} t_s + t_{fs}] + [N_{TX} N_{\delta t} t_s + t_f], \qquad (11)$$

where N_{TX} is the number of transmitters, $N_{\delta t}$ is the number of time steps taken with a time increment δt, t_s is the solution time for a single right side, and t_{fs} and t_f are respective factorization times for the initial steady state problem 9 and the time-marching problem 5). Usually t_{fs} is fractionally larger than t_f, and its factorization memory is a bit larger, but the differences are not significant for our purposes of comparison.

Our equations are to be time-stepped between $T_1 = 10^{-6}$ and $T_2 = 10^{-2}$ s. If a single δt is sought for the entire time interval, then its size is determined by the smallest time needed. Thus, $\delta t = T_1$ and so the number of time steps is $N_{\delta t} = T_2/\delta t = 10^4$ for this example. The factorization and solution times depend upon the size of the problem, the number of CPUs, and the amount of available memory. For a 70^3 mesh and our hardware configuration, $t_f = 163$ s and $t_s = 0.80$ s. Further details about the solution time are provided in the next section.

Unfortunately, the total time is 8e6 s or 92 days; this is prohibitive. The difficulty arises because of the excessive number of time steps; keeping the same δt for the entire modeling requires 10^7 forward solutions. To circumvent this, we divide the modeling time interval into P subintervals, each of which has a constant δt. One additional factorization is required for each subinterval. The time for a forward modeling is now

$$T_{\text{total}} = [N_{TX} t_s + t_{fs}] + P[N_{TX} N_{\delta t} t_s + t_f]. \qquad (12)$$

For the example here, we use subintervals that are a decade in time, and each subinterval is modeled with 15 equal time steps. Five factorizations are required, but only 60,000 forward modelings. The total time is thereby reduced to 17,373 s or about 4.8 h. The computations were carried out on a single node having six cores and 16 GB per core.

Comparing computation times using direct and iterative solvers

The choice of whether, or under what conditions, to use an iterative or direct solver is problem-dependent. If the number of sources and time steps is small, then an iterative technique might be more efficient. Also, if the number of cells is exceedingly large, there might not be enough memory to carry out and store the factorizations, and in such cases we may again have to appeal to an iterative solver. However, there is a large range of problems for which a direct solver can be much faster. To shed some quantitative light on this, we carry out the following analysis.

As a first comparison, we examine the relative computation times for a single solution of Maxwell's equations. For the iterative solver, the main aspects are the preconditioner, the desired tolerance, and the maximum number of iterations. Here, we use the algorithm BiCGstab Barrett et al. (1994), with an ILU preconditioner, and have set a tolerance to 10^{-10}. The maximum number of iterations is limited to 1000. If this number is thought to be too conservative, then it is a simple matter to adjust the final times because the time for an iterative solver is linearly related to the number of iterations. We consider different-sized meshes that range from 40^3 to 90^3. The results are summarized in Table 1. For the small (40^3) and for the large (90^3) systems, the total times for the direct and iterative solutions are approximately equal. For 50^3 and 60^3 systems, the direct solver is a factor of two to three times more efficient. The direct solver however requires progressively more memory, needing about 101 Gb for the 90^3 system.

The major benefits of the direct solver arise when the number of time steps and transmitters increase. To illustrate this, we consider the 70^3 mesh and the problem addressed at the beginning of this section. Data at 35 time channels are to be simulated. For the iterative method, we can sample in variable increments in δt, but the time marching still needs to be initiated before the first data time channel. Solution of Maxwell's equations at 42 times is sufficient. For the direct method, we divide the total time interval into four partitions and sample at 15 equispaced times in that interval. One factorization is required for each interval and another to solve the initial steady state problem. The results are summarized in Table 2. Even for a single transmitter, the total simulation time for the direct solver is substantially less than for the iterative method, 1036 s versus 4930 s. That discrepancy is exacerbated as the number of transmitters is increased.

For the above statement to be properly interpreted, we also need to factor in the number of processors being used. With the iterative approach, each transmitter can be spawned to a different processor. Thus, if n_p processors were used, then the total time for the iterative method is the time for a single transmitter multiplied by N_{TX}/n_p. For the example here, the direct solver used 12 processors.

Table 1. Comparison of solution times and memory for a single time step using direct and iterative solvers on variable-sized (n^3) meshes: t_f is the factorization time, Iterations is the number of iterations used by the iterative solver, t_s is the solution time taken for single Maxwell solution using the direct solver. All times are in seconds.

Size of mesh	Method	t_f (s)	Memory (GB)	Iterations	t_s (s)	Total time (s)
40^3	iter	—	0.1	228	—	7
40^3	dir	7	4	—	0.1	7
50^3	iter	—	0.3	1000	—	63
50^3	dir	23	10	—	0.2	23
60^3	iter	—	0.5	1000	—	113
60^3	dir	65	21	—	0.5	65
70^3	iter	—	0.8	1000	—	205
70^3	dir	163	40	—	0.8	166
80^3	iter	—	1.1	1000	—	279
80^3	dir	238	63	—	1.0	239
90^3	iter	—	1.6	1000	—	463
90^3	dir	459	101	—	1.8	461

Table 2. Comparison of iterative and direct solvers for a complete forward modeling for a variable number of transmitters on a 70^3 mesh: $N_{\delta t}$ is the number of time steps taken, t_f is the time taken to factor the matrix, t_s is the solve time for a single transmitter and a single time step for the direct solver. The final column is the total time needed for forward modeling.

N_{TX}	Method	# Factorizations	$N_{\delta t}$	t_f (s)	t_s (s)	Total time (s)
1	iter	—	42	—	—	4930
1	dir	2	10,000	163	0.8	8476
1	dir	5	60	163	0.8	1036
10	iter	—	42	—	—	4.9E4
10	dir	5	60	163	0.8	1168
100	iter	—	42	—	—	4.9E5
100	dir	5	60	163	0.8	2660
1000	iter	—	42	—	—	4.9E6
1000	dir	2	10,000	163	—	8.0E6
1000	dir	5	60	163	0.8	17,373

The above comparisons illustrate the potential benefits of using a direct solver for carrying out the forward modeling. The advantages of this will become even more apparent in the next section.

INVERSION

The real benefit of using decomposition becomes apparent when we treat the inverse problem. Our inversion algorithm is essentially that described in Haber et al. (2007) and it bears similarity to other work (Madden and Mackie, 1989; Newman and Alumbaugh, 1997a, 1997b; Dorn, 2000; Newman and Commer, 2005; Commer and Newman, 2005). It is based upon a Gauss-Newton procedure where a model $m = \ln \sigma$ is sought. We thus solve the following optimization problem

$$\begin{matrix} \min & \phi(m) = \phi_d + \beta\phi_m(m, m_{\text{ref}}) \\ s.t & m^L \le m \le m^U \end{matrix}. \tag{13}$$

In equation 13, ϕ_d is the data misfit, β is an adjustable tradeoff parameter, and ϕ_m is the regularization term. Here, we choose

$$\phi_d = \frac{1}{2}\|W_d(QA(m)^{-1}q - d^{\text{obs}})\|_2^2, \tag{14}$$

where d^{obs} are the observed data and W_d is a diagonal data weighting matrix whose elements are the reciprocals of the estimated uncertainty for each datum. Symbol Q is an interpolation matrix (a discretization of a projection operator) that extracts data from the computed fields $u = A(m)^{-1}q$.

The regularization functional, ϕ_m, is defined as

$$\phi_m(m, m_{\text{ref}}) = \frac{\alpha_s}{2}\|W_s(m - m_{\text{ref}})\|_2^2 + \sum_{i=1}^{3}\frac{\alpha_i}{2}\|W_i(m - m_{\text{ref}}\|_2^2, \tag{15}$$

where $(i = 1, 3)$ corresponds to the spatial directions (x, y, z). The first term in ϕ_m is used to design a solution that is close to a reference model m_{ref}. The remaining terms penalize variation in the three spatial directions. Symbol W_s is a diagonal weighting matrix containing cell volumes that arise from the midpoint discretization of the integral

$$\int_\Omega (m - m_{\text{ref}})^2 dv,$$

and W_x, W_y, W_z are first-order finite difference matrices of the corresponding orientation multiplied by a cell volume term. The α_s, α_x, α_y, α_z are adjustable constants that allow the user to emphasize structural variation in particular directions or closeness to a reference model. Although seemingly complicated, the flexibility afforded by this generic objective function is important in incorporating a priori information into the inversion and also in handling artifacts that can arise when using a limited number of transmitters and receivers.

By writing

$$\phi_m(m, m_{\text{ref}}) = \frac{1}{2}\|W_m(m - m_{\text{ref}}\|^2, \tag{16}$$

and differentiating the objective function $\phi(m)$ in equation 13, we obtain the gradient

$$g(m) = J(m)^T W_d^T W_d (d(m) - d^{\text{obs}}) + \beta W_m^T W_m m_{\text{ref}} \quad (17)$$

where $J(m)$ is the sensitivity matrix. For a fixed β, the minimization of $\phi(m)$ is achieved by using a Gauss-Newton approach, and at each iteration, the system of equations equals

$$(J(m)^T J(m) + \beta W_m^T W_m)\delta m = -g(m), \quad (18)$$

is approximately solved to obtain a perturbation δm. We use a preconditioned conjugate gradient least squares (CGLS) solver. The preconditioner is a quasi-Newton estimate of the Hessian and is composed of BFGS (Nocedal and Wright, 1999) update vectors that are accumulated at successive iterations in the inversion. The determination of the unknown regularization parameter β is handled through an iterative cooling process. The CGLS solver requires that the matrix on the left side of equation 18 be applied onto a vector many times. Symbol W_m represents a sparse matrix, so application of W_m onto a matrix requires minimal computation. The sensitivity matrix $J(m)$ is a large full matrix and we cannot store it. It is well-known (Haber et al., 2000b; Vogel, 2001) that the sensitivity matrix can be expressed in factored form

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

where Q is the interpolation matrix, $A(m)$ is the forward-modeling operator given in equation 10, u are the fields, and $G(m,u)$ is a known sparse matrix,

$$G(m,u) = \frac{\partial [A(m)u]}{\partial m}. \quad (19)$$

We note that G requires the fields for all transmitters and for all times. Although these have been computed to evaluate the misfit and gradient, they need to be stored or recomputed to carry out the effect of multiplying J or J^T on a vector.

The application of Jv or $J^\top v$ requires that a forward modeling be carried out, and thus the implications of factoring the forward-modeling matrix now become apparent. First, the computation of the gradient requires two forward modelings; one to compute the responses with the current model m, and another to apply the Jacobian. Second, even with a good preconditioner, the number of CG iterations needed to get a good solution, N_{CG}, is typically 10–20. Third, once the perturbation δm is calculated, a line search is required to find the scale factor γ to update the model as $m^{n+1} = m^n + \gamma \delta m$. If $N_{CG} = 15$, the number of forward modelings is 32 plus any required for a line search. If there are multiple transmitters, then a single Gauss-Newton (GN) iteration can involve tens of thousands of solutions of Maxwell's equation. Because 10–15 GN iterations typically are needed to complete the inverse problem, the number of solutions of Maxwell's equations needed for a single inversion can easily reach a million. For example: $N_{TX} = 100$, $N_{\delta t} = 50$, $N_{CG} = 15$,, and number of GN iterations is 15, requires 2,400,000 solutions. Factorization is thus highly important. Our experience has been that even with a single transmitter, inversion using direct solvers can be substantially faster than our previous inversion software which used iterative solvers.

Last, we note that the above procedure necessitates the storage of all factorizations used to solve a complete forward problem. This requires computational resources with substantial memory.

Working with different field components

The above GN procedure is valid for any of the fields $\vec{E}, \vec{H}, d\vec{B}/dt$, but there are important details that need to be incorporated in practice. Because the equations are formulated in terms of $\vec{H}$, the sensitivity is provided as $J(m) = -QA(m)^{-1}G(m,u)$ where Q is an interpolation matrix. If the data are dB/dt, then time derivatives of the fields are required. This is done by first carrying out the forward modeling at all times and then taking the derivative. These operations can be coded directly into the matrix Q.

For electric field data, an alteration is needed. The electric field is obtained from the magnetic field by equation 20

$$\vec{E}^{i+1} = \Gamma^{-1} C_e \vec{H}^{i+1} - \Gamma^{-1} s_r^{i+1} + \alpha \Gamma^{-1} \Upsilon \vec{E}^i. \quad (20)$$

The sensitivity is thus more complicated than for $\vec{H}$ alone. Detail is provided in the Appendix A.

Inversion of synthetic data

For a synthetic example, we consider a conductive (1 S/m) block and a resistive (10^{-4} S/m) block in a 0.01 S/m half-space. Our code is designed to invert any combination of fields $\vec{E}, \vec{H}, d\vec{B}/dt$ from either grounded or loop transmitters and thus there are many potential survey designs that could serve as test examples. We select two that are most commonly employed; we invert $d\vec{B}/dt$ data from loop transmitters, and $\vec{E}$ field data from grounded sources. The earth model is that shown in Figure 3. The conductive prism is a relatively easy target for any inductive data, but the

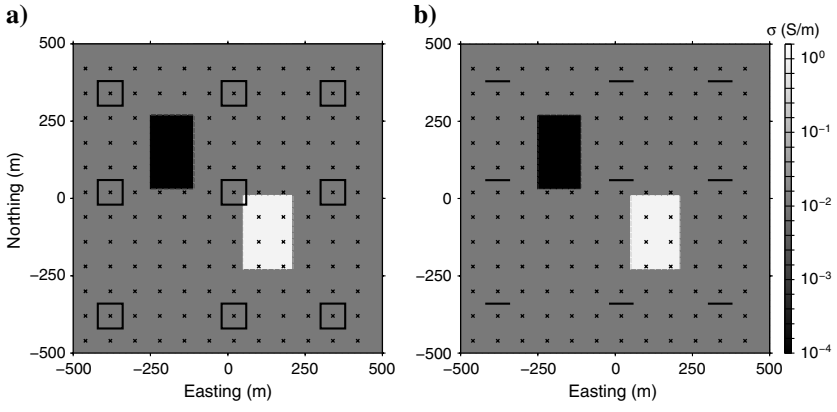

Figure 3. Geometry for ground surveys. In (a), there are nine loop transmitters and a grid of 144 receivers measuring three components of dB/dt. In (b), there are nine grounded transmitters and a grid of 144 receivers measuring the two horizontal components of E.

resistive prism is more difficult. However, the resistor should be visible by using the E-field survey.

For these synthetic examples, the earth is discretized with a $67 \times 67 \times 53$ mesh yielding 237,917 cells. Cell sizes in the interior of the mesh are $20 \times 20 \times 15$ m and the padded volume is a cube of about 5 km on each side. The waveform is a step-off and data are obtained at 16 time channels in the time range $10^{-5} \sim 10^{-2}$ s. The spatial dimension of cells in the core region is smaller than the diffusion distance for the earliest time for a background half-space conductivity of 10^{-2} S/m. The outer boundaries of the volume are greater than the diffusion distance associated with latest time channel of data. The equations are time-stepped using three subintervals of uniform δt and a total of 50 time steps. One factorization of the matrix is required to generate the initial fields and so four factorizations are needed in total. A comparison of the responses with a 1D code, and a uniform conductivity of 10^{-2} S/m, confirmed the validity of the spatial and time discretizations.

We chose not to add random noise to the data prior to inversion but we did assign an uncertainty to each datum to be inverted. Irrespective of whether a datum was $\vec{E}$ or $d\vec{B}/dt$, the uncertainty assigned was 1% of the absolute value of the datum plus a floor. The inversions were carried out by minimizing the objective function in equation 3 with a reference and starting model equal to the half-space resistivity.

The memory required to factor the forward modeling matrix was about 14 Gb. There are four factorizations needed and additional memory is required to store the fields from all transmitters and to carry out computations. In total, we used 60 Gb of memory and distributed the workload across 12 processors (two machines). The time for a factorization was $t_f = 75$ s and the solve time was $t_s = 0.40$ s. Once the matrix was factored, a complete forward solution to simulate the data for nine transmitters and 50 time steps took about 104 s.

Experiment 1: Inductive source with $d\vec{B}/dt$ data

We first invert data obtained from nine transmitter loops on the surface a flat earth. The data are three components of $d\vec{B}/dt$ on a 12×12 surface grid. The layout of the transmitters and the data locations are shown in Figure 3a.

There were 62,208 data (9 Tx, 144 Rx location, three components, 16 time channels). The inversion was stopped once the target misfit ($\phi_d = N$) was achieved. Thirteen Gauss-Newton steps were taken with a maximum of 10 CG iterations per GN step. The progress of the inversion is encapsulated in Figure 4.

Cross sections and plan view maps of the recovered model are shown in Figure 5a. The conductive prism is well-imaged. Its maximum conductivity is somewhat larger than the true value but this overshoot is mostly the result of generating a smooth solution when the true model is blocky. The resistive prism is barely visible in the final solution. This is not surprising. Resistive bodies can be difficult to image with inductive sources and magnetic field data.

Experiment 2: Grounded sources with E-field data

In this example, we substitute grounded dipoles for loop transmitters. The locations of the transmitters are shown in Figure 3b. The data are E_x, E_y measured at the same receiver sites as in Example 1. The total number of data is 41,472. The inversion

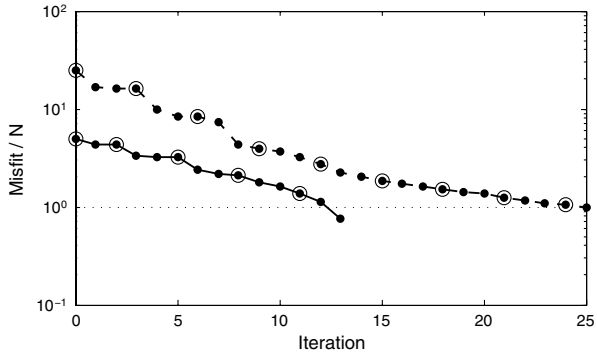

Figure 4. Convergence curves for the nine Tx example. The results for dB/d data (Experiment 1) are shown by the solid line. The results for $\vec{E}$ field data (Experiment 2) are shown by the dashed line. The target misfit is $\phi_d = N$. The circles on the misfit curves indicate when β is changed.

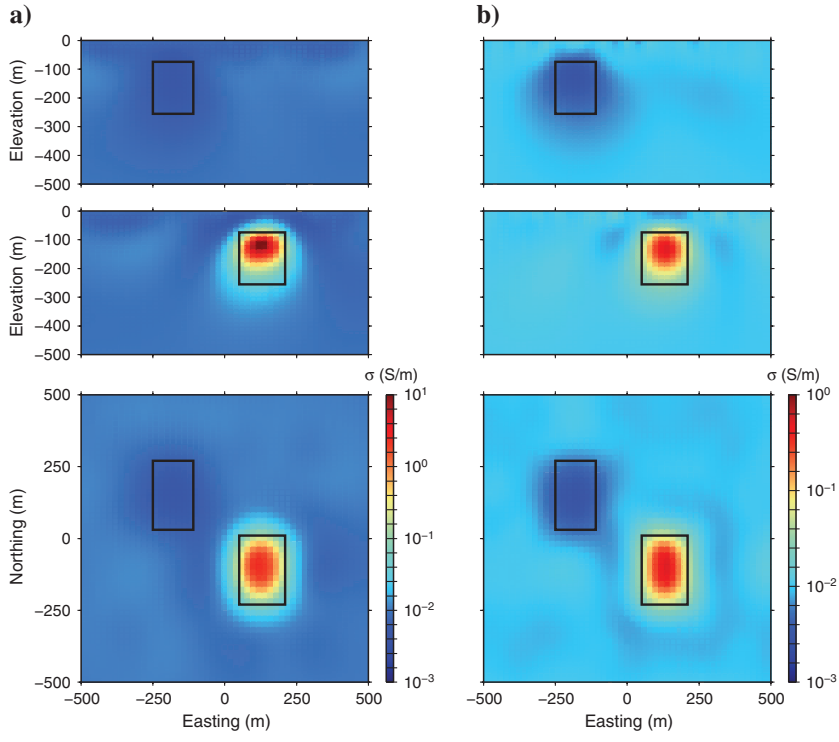

Figure 5. Cross sections and plan view maps of the recovered conductivity. The black squares indicate the locations of the resistive and conductive prisms. The left column (a) corresponds to the model obtained from inverting dB/dt data (Experiment 1), the right column (b) corresponds to the E-field data (Experiment 2).

was run with identical parameters as were used for Example 1. The convergence curve and model are shown, respectively, in Figures 4 and 5b. Twenty-five GN iterations were required to achieve the target misfit. The planview and cross sections show that the conductive and resistive prisms are well-imaged, but the magnitude of the conductivity and resistivity contrasts with respect to the half-space has been underestimated.

Solution times for iterative and direct solvers

To compare the solution times for the inverse problem using iterative and direct solvers, we consider a single transmitter in the center of the model domain and use the same mesh and time channels. Both inversion algorithms begin with the same β regularization factor and use two CG iterations. We invert the surface E-field data and the total number of data is 41,472. The results are summarized in Table 3 and Figure 6. The solution using the direct method took about 3.5 h, whereas the iterative method took about 3.5 days.

The iterative solver requires more Gauss-Newton iterations to achieve the desired misfit than does the direct solver. Correspondingly, it has also used a smaller value of β and thus the additive perturbations and final model are somewhat rougher. This is quantified by evaluating the model norm. The model norms for the two solutions are, respectively, 4.96e4 and 5.80e4 (see Table 3). The difference in the two solutions is likely caused by poorer accuracy in solving the forward problem with the iterative method. The maximum number of iterations was set to 1000 and, for many modelings, our desired tolerance was not achieved.

The above example used a single transmitter. For the direct solver, when the number of transmitters is increased to nine, the total time for inversion increases only to 7.25 h using six processors. (Table 3).

In our comparisons of direct and iterative solvers, we limited the number of CG iterations to two. The resultant perturbations have been satisfactory because they have ultimately generated a final model that acceptably fits the data. However, a better solution, one that produces the same misfit but has less structure (evaluated by the numerical value of the model norm) might be found if the GN equations were solved more accurately at each iteration to yield a higher quality update δm. The result of using $N_{CG} = 10$ is illustrated in Table 3 and Figure 6. The final solution is achieved in only 17 iterations. The computation time, compared to $N_{CG} = 2$, increases from 3.5 to 4.25 h. (See Table 3). The final β for $N_{CG} = 10$ is 3.2×10^{-2} and the model norm is 3.3×10^{4}. Thus, a superior solution is obtained compared to using $N_{CG} = 2$. A solution with the iterative solver and $N_{CG} = 10$ would have taken about 17 days.

The synthetic inversions carried out here illustrate the essential workings of the algorithm. We next turn to the more challenging problem of working with field data.

INVERSION OF UTEM DATA AT SAN NICOLÁS

San Nicolás is an unmined, volcanic-hosted, Cu-Zn massive sulfide deposit located in central Mexico in the state of Zacatecas. The reserve estimates are 72 million tons, grading 1.4% copper and 2.27% zinc, making it the largest massive sulfide deposit yet discovered in Mexico, and a world-class deposit. The deposit is a continuous but geometrically complex body which is covered by 175–250 m of variable composition overburden. The local geology is also somewhat complex and contains numerous sedimentary and volcanic units. The deposit has been thoroughly drilled and logged and, as a result, the geology and physical properties of the deposit are reasonably well understood. Figure 7 shows an east–west cross section through the deposit. The massive sulfide is characterized by having high density, magnetic susceptibility, electrical conductivity, and chargeability. As a result, the deposit has been investigated using several geophysical surveys, including gravity, magnetics, DC resistivity, induced polarization, airborne EM, and ground-based CSEM/CSAMT surveys (Phillips, 2001; Oldenburg and Pratt, 2007). It has had a long history of being a test case for the application of geophysics to the exploration of massive sulphides, and hence we have adopted it here.

For this paper, we are interested in the electrical conductivity. The ore-body and the variable thickness overburden are moderately conductive ($0.02 \sim 0.2$ S/m) compared to the background host rocks ($0.01 \sim 0.001$ S/m). The goal is to locate the conductive sulfide, but the challenge is that the overburden, a tertiary volcanic breccia, has a conductivity in the range of that found in the sulfide. This masks the sulfide.

A UTEM survey was conducted over the San Nicolás deposit in December of 1998. The UTEM system uses a continuous on-time loop transmitter with a sawtooth waveform. The system is described in detail in the paper by West et al. (1984). At San Nicolás, the UTEM survey consists of three large loops; a main loop which

Table 3. Time and memory requirements for single and multiple transmitter inversions. The modeling mesh is $67 \times 67 \times 53$ mesh yielding 237,917 cells. Variable T_{for} is the time in seconds to carry out a forward modeling with all transmitters and all times. The memory needed for factorization is 60 GB. The memory for the stored fields is 0.26 GB for each Tx. Here, Φ_m denotes the final norm.

Data type	N_{TX}	Method	T_{for} (s)	# GN	# CG	# β	Total time (Hr:min)	Total memory (GB)	Φ_m	Final β
E	1	dir	22	22	2	8	3:32	60	4.96E4	1.28E-3
E	1	iter	—	25	2	9	3.5 days	1.5	5.80E4	2.56E-4
E	9	dir	104	21	2	7	7:14	63	3.28E4	6.40E-3
E	1	dir	22	17	10	6	4:15	60	3.33E4	3.20E-2
E	9	dir	104	14	10	5	12:31	63	2.59E4	1.60E-1
dB/dt	9	dir	98	13	10	5	9:43	63	2.44E4	1.60E-1

is completely external to the deposit, a loop south of the deposit, and a loop to the east of the deposit. Data, vertical component of dB/dt at nine time channels, were recorded along north–south and east–west lines over the deposit. The survey geometry is shown in Figure 8. The initial report on the UTEM survey indicated that the San Nicolás deposit produced a late time response of, at most, 5% of the primary field of the transmitter. However, a more complete interpretation of the data was hampered by the complexity of the local geology and inability to invert the data.

The data used here have been inverted previously by Napier et al. (2006) and Napier (2007) using our original code which is the Gauss-Newton implementation of the work done in Haber et al. (2004). The inversion result was satisfactory, but it took many weeks to run.

The inversion of any field data set requires a workflow. Here, we have capitalized on the work of Napier and use his final data and assigned uncertainties. Our earth volume was discretized using a rectangular mesh. The central volume consisted of 25 cubic meter cells and padding cells that extended the mesh boundary out to several kilometers from the survey area. The final volume consisted of $75 \times 54 \times 54 = 218{,}700$ cells.

The UTEM waveform is a triangular function starting at time $t = 0$. We found that the forward modeling needed to be time-stepped for 3/4 of a cycle to achieve equilibrium. The resultant waveform and our discretization are shown in Figure 9. The initial 3/4 cycle is modeled with a constant δt, and hence can be computed with a single factorization. The waveform is densely sampled in log-time on the final ramp and sampling on that ramp begins a decade before the first datum time. The equations are therefore time-stepped from $T_1 = 10^{-6}$ to $T_2 = 10^{-2}$ and this requires four factorizations. The complete problem uses five factorizations which must be stored.

Standard deviations were assigned based on a percentage ranging from 3% for the earliest time channel to 7% for the latest time channel. The floor value of the standard deviation was set to 5% of the median of the absolute value for the data in each channel. The total number of data from three loops and nine time channels was 3523.

The inversion was run with half-space starting and reference models and the convergence is illustrated in Figure 10. The final misfit was 1660. A top view and a cross section of the conductivity cube are shown in Figure 11. The high-conductivity overburden appears to be quite variable in thickness and it has a strong northwest–southeast trend over the deposit. The body near the center and at a depth of about 300 m is the sought deposit. We see a separation between the top of the deposit and the body.

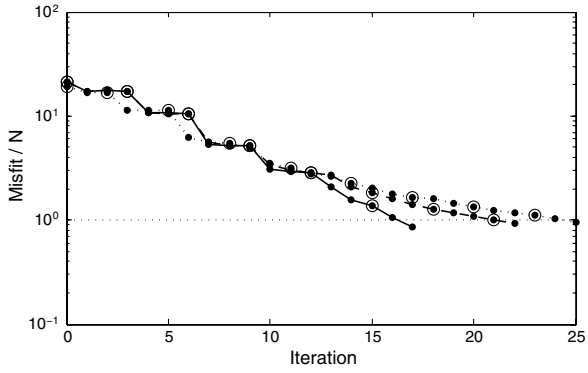

Figure 6. Comparison of iterative and direct solvers for inversion. Dotted line: Iterative solver with two CG iterations (total time 3.5 days). Dashed line: Direct solver with two CG iterations (total time 3:32 hr:min). Solid line: Direct solver with 10 CG iterations (total time 4:15 hr:min).

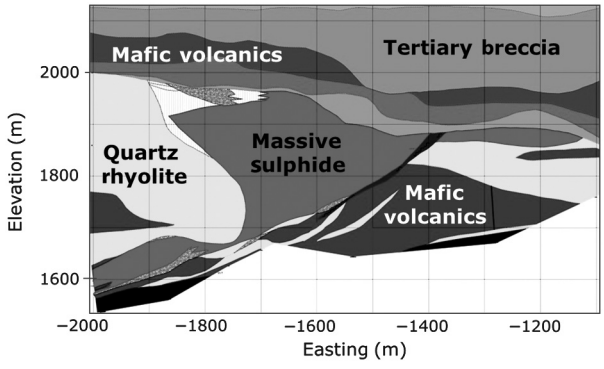

Figure 7. A geologic cross section of the San Nicolás deposit.

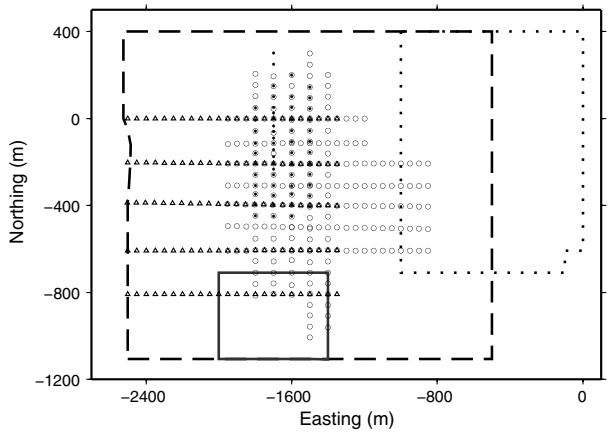

Figure 8. UTEM survey geometry at San Nicolás. Three transmitter loops are indicated by the solid, dashed, and dotted lines. The associated receiver locations are denoted by the dots, circles, and triangles, respectively.

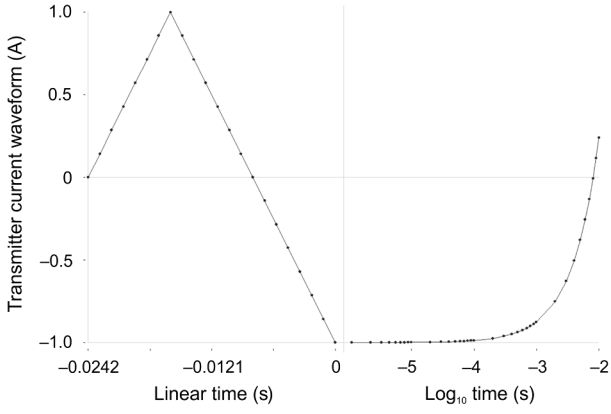

Figure 9. UTEM transmitter current waveform. The first 3/4 cycle is plotted on a linear scale. The right portion of the plot shows the time sampling on a logarithmic time axis. The dots indicate times at which Maxwell's equations are solved.

Considerable drilling and logging has been carried out at San Nicolás, and a 3D conductivity model has been generated. The deposit is 3D and consists of a bulky upper mass with a small keel extending outward and downward from the bottom. The units of high conductivity are the overburden and the deposit and these are shown in two perpendicular cross sections in Figure 12. The dashed lines on Figure 12 represent an approximate outline of the sulfide body in two orthogonal directions. These outlines have been transferred to the respective cross-sectional plots in Figure 12.

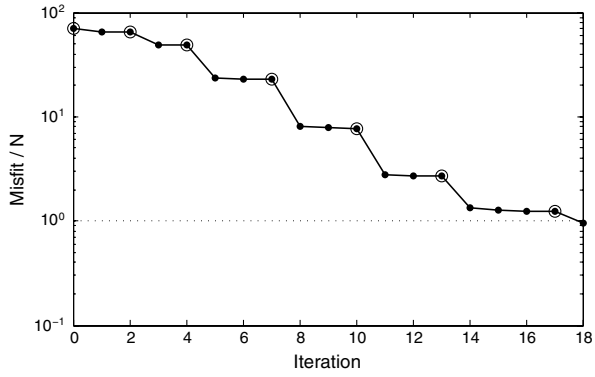

Figure 10. Convergence curve for San Nicolás inversion.

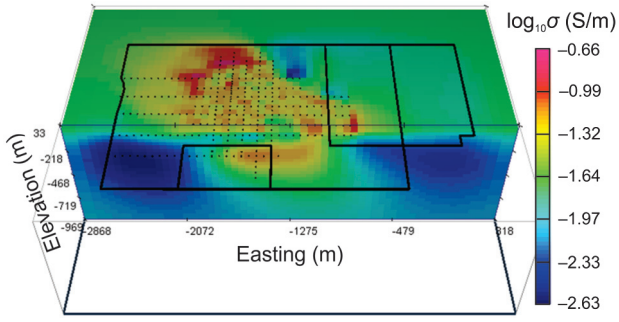

Figure 11. Three-dimensional conductivity model at San Nicolás

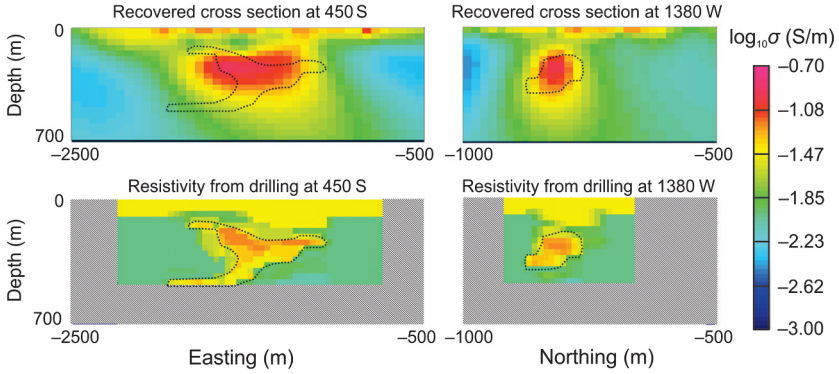

Figure 12. Three-dimensional conductivity model and rock models at San Nicolás. Perpendicular cross sections of the inverted conductivity model are displayed on the top row. The bottom row consists of cross sections of a 3D conductivity model obtained from drill results. The dashed line, obtained from the rock model, denotes an approximate boundary of the sulfide. This is superposed on the recovered conductivity in the top row.

Overall, the inversion has imaged the deposit in the lateral and vertical directions. The keel is not observed, but it has a small volume. Numerical modeling shows that it would be undetectable from a surface experiment.

CONCLUSION

Recent advances in matrix decomposition have allowed us to generate a practical 3D inversion methodology for multisource time domain electromagnetic data. The methodology is general in the sense that either inductive or grounded sources can be used and any combination of electromagnetic fields, collected in the air, on the surface, or in boreholes can be inverted. Small and midsized problems, 10^5 cells, can readily be handled on a single node, multicore machine, and inversion times are less than a day. Larger problems require multinodes with extended memory. The efficacy of our technique was illustrated by inverting multisource UTEM field data set acquired at San Nicolás.

The efficiency of an iterative solver depends upon the selected preconditioner and the tolerance level for solution. In our synthetic studies, we were able to achieve a better solution, as judged by the final model norm, using a direct solver. Despite the attractiveness of using a direct solver, the downside is that larger problems require expanded memory, and at some point, the available memory may be insufficient. One option is to break the model domain into segments and introduce a tiling operation, or use a more formal domain decomposition methodology where Maxwell's equations are factored into smaller subdomains. Another option is to revert back to an iterative technique.

ACKNOWLEDGMENTS

This work was carried out under NSERC CRD and IRC programs and sponsoring companies: Cameco, Teck, Newmont, Barrick, Vale, Xstrata Nickel, and Anglo American.

APPENDIX A

COMPUTATION OF SENSITIVITIES

There are two critical steps in the inversion of large-scale data. The first is the forward problem, and the second is the computation of the sensitivity matrix times a vector. It is important to note that the sensitivity matrix is not needed explicitly and a sensitivity-vector product and its adjoint can be computed effectively without computing the sensitivity matrix directly (Haber et al., 2000b).

Recall that the sensitivities are defined as the derivatives of the data with respect to the conductivity. Its discrete analog is

$$J_H = Q_H \frac{\partial \vec{H}}{\partial m} \quad \text{and} \quad J_E = Q_E \frac{\partial \vec{E}}{\partial m}, \tag{A-1a}$$

where Q_H and Q_E are sparse matrices that compute the data given the magnetic/electric fields. Considering the finite volume formulation equation 4 we have

$$\frac{\partial}{\partial m}(C^\top \vec{E}^{i+1} + \alpha M \vec{H}^{i+1} - \alpha M \vec{H}^{i}) = 0, \tag{A-2a}$$

$$\frac{\partial}{\partial m}(C\vec{H}^{i+1} - S\vec{E}^{i+1} - \alpha \Upsilon \vec{E}^{i+1} - \vec{s}_r^{\,i+1} + \alpha \Upsilon \vec{E}^{i}) = 0. \tag{A-2b}$$

Equation A-2a does not depend on σ, and therefore we obtain

$$C^\top \vec{E}_m^{i+1} + \alpha M \vec{H}_m^{i+1} - \alpha M \vec{H}_m^{i} = 0, \tag{A-3}$$

where $(\cdot)_m$ represents a derivative with respect to m. Differentiating the second equation, we obtain

$$C\vec{H}_m^{i+1} - S\vec{E}_m^{i+1} - \alpha \Upsilon \vec{E}_m^{i+1} + \alpha \Upsilon \vec{E}_m^{i} + \frac{\partial}{\partial m}(S\vec{E}^{i+1}) = 0. \tag{A-4}$$

Moving the new terms to the right side, we have

$$C\vec{H}_m^{i+1} - S\vec{E}_m^{i+1} - \alpha \Upsilon \vec{E}_m^{i+1} + \alpha \Upsilon \vec{E}_m^{i} = -\frac{\partial}{\partial m}(S\vec{E}_m^{i+1}). \tag{A-5}$$

The equations are identical to the forward problem, but with a different right side. They imply that to obtain the sensitivities we need to solve the forward problem for the sensitivity matrices $\vec{E}_m^{i+1}$ and $\vec{H}_m^{i+1}$ where the right side is the matrix

$$V_{i+1} = \frac{\partial}{\partial m}(S\vec{E}_m^{i+1}). \tag{A-6}$$

Thus, the sensitivity is a product of three matrices, the matrix $V = [V_1^\top, \ldots, V_n^\top]^\top$ and Q which are sparse in general, and the inverse of the forward modeling matrix which is dense. Although storing the matrix is generally impossible, computing a matrix vector product is possible by solving a linear system. Furthermore, if direct methods have been used to compute a decomposition of the matrix, then such product has a negligible cost.

REFERENCES

Amestoy, R., I. Duff, J.-Y. L'Excellent, and J. Koster, 2001, A fully asynchronous multifrontal solver using distributed dynamic scheduling: SIAM Journal on Matrix Analysis and Applications, **23**, 15–41, doi: 10.1137/S0895479899358194.

Aruliah, D., and U. Ascher, 2003, Multigrid preconditioning for time-harmonic Maxwell's equations in 3D: SIAM Journal on Scientific Computing, **24**, 702–718, doi: 10.1137/S1064827501387358.

Ascher, U., 2010, Numerical methods for evolutionary PDE's: SIAM.

Ascher, U., and L. Petzold, 1998, Computer methods for ordinary differential equations and differential-algebraic equations: SIAM.

Barrett, R., M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, 1994, Templates for the solution of linear systems: Building blocks for iterative methods: SIAM.

Bossavit, A., 1998, Computational electromagnetism. Variational formulation, complementarity, edge elements: Academic Press.

Commer, M., and G. A. Newman, 2006, An accelerated time domain finite difference simulation scheme for three-dimensional transient electromagnetic modeling using geometric multigrid concepts: Radio Science, **41**, 1–15, doi: 10.1029/2005RS003413.

Demmel, J. W., S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, 1999, A supernodal approach to sparse partial pivoting: SIAM Journal on Matrix Analysis and Applications, **20**, 720–755, doi: 10.1137/S0895479895291765.

Dey, A., and H. F. Morrison, 1979, Resistivity modeling for arbitrarily shaped three-dimensional structures: Geophysics, **44**, 753–780, doi: 10.1190/1.1440975.

Dorn, O., 2000, A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets: Inverse Problems, **16**, 1119–1156, doi: 10.1088/0266-5611/16/5/303.

Haber, E., and U. Ascher, 2001, Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients: SIAM Journal on Scientific Computing, **22**, 1943–1961, doi: 10.1137/S1064827599360741.

Haber, E., U. Ascher, D. Aruliah, and D. Oldenburg, 2000a, Fast simulation of 3D electromagnetic using potentials: Journal of Computational Physics, **163**, 150–171, doi: 10.1006/jcph.2000.6545.

Haber, E., U. Ascher, and D. Oldenburg, 2000b, On optimization techniques for solving nonlinear inverse problems: Inverse Problems, **16**, 1263–1280, doi: 10.1088/0266-5611/16/5/309.

Haber, E., U. Ascher, and D. Oldenburg, 2004, Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach: Geophysics, **69**, 1216–1228, doi: 10.1190/1.1801938.

Haber, E., D. Oldenburg, and R. Shektman, 2007, Inversion of time domain 3D electromagnetic data: Geophysical Journal International, **171**, 550–564, doi: 10.1111/j.1365-246X.2007.03365.x.

Hu, J., R. Tuminaro, P. Bochev, C. Garasi, and A. Robinson, 2006, Towards an independent algebraic multigrid method for Maxwell's equation: SIAM Journal on Scientific Computing, **27**, 1669–1688, doi: 10.1137/040608118.

Hyman, J., and M. Shashkov, 1998, Mimetic discretizations for Maxwell's equations and equations of magnetic diffusion: Technical report, Los Alamos National Laboratory (LA-UR-98-1032).

Hyman, J., and M. Shashkov, 1999, Mimetic discretizations for Maxwell's equations: Journal of Computational Physics, **151**, 881–909, doi: 10.1006/jcph.1999.6225.

Madden, T., and R. Mackie, 1989, Three-dimensional magnetotelluric modeling and inversion: Proceedings of the IEEE, **77**, 318–321.

Napier, S., 2007, Practical inversion of 3D time domain electromagnetic data application to the San Nicolas deposit: M.Sc. thesis, The University of British Columbia.

Napier, S., D. Oldenburg, E. Haber, and R. Shekhtman, 2006, 3D inversion of time domain data with application to San Nicolas: 76th Annual International Meeting, SEG, Expanded Abstracts, 1303–1307.

Newman, G., and D. Alumbaugh, 1995, Frequency-domain modelling of airborne electromagnetic responses using staggered finite differences: Geophysical Prospecting, **43**, 1021–1042, doi: 10.1111/gpr.1995.43.issue-8.

Newman, G., and D. Alumbaugh, 1997a, Three-dimensional massively parallel electromagnetic inversion — I, theory: Geophysical Journal International, **128**, 345–354, doi: 10.1111/gji.1997.128.issue-2.

Newman, G., and D. Alumbaugh, 1997b, Three-dimensional massively parallel electromagnetic inversion — II, analysis of a crosswell electromagnetic experiment: Geophysical Journal International, **128**, 345–354, doi: 10.1111/gji.1997.128.issue-2.

Newman, G. A., and M. Commer, 2005, New advances in three dimensional transient electromagnetic inversion: Geophysical Journal International, **160**, 5–32, doi: 10.1111/gji.2005.160.issue-1.

Nocedal, J., and S. Wright, 1999, Numerical optimization: Springer.

Oldenburg, D., and D. Pratt, 2007, Geophysical inversion for mineral exploration: A decade of progress in theory and practice, *in* B. Milkereit, ed., Proceedings of Exploration 07: Fifth Decennial International Conference on Mineral Exploration, 6195.

Oldenburg, D. W., E. Haber, and R. Shekhtman, 2008, Forward modelling and inversion of multi-source TEM data: 78th Annual International Meeting, SEG, Expanded Abstracts, 559–563.

Phillips, N., 2001, Geophysical inversion in an integrated exploration program: Examples from San Nicolas deposit: M.Sc. thesis, The University of British Columbia.

Pratt, R., 1999, Seismic waveform inversion in the frequency domain, Part 1: Theory, and verification in a physical scale model: Geophysics, **64**, 888–901, doi: 10.1190/1.1444597.

Vogel, C., 2001, Computational methods for inverse problem: SIAM.

West, G., J. Macnae, and Y. Lamontagne, 1984, A time domain EM system measuring the step response of the ground: Geophysics, **26**, 1010–1026, doi: 10.1190/1.1441716.

Yee, K., 1966, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media: IEEE Transactions on Antennas and Propagation, **14**, 302–307, doi: 10.1109/TAP.1966.1138693.

Zaslavsky, M., V. Druskin, and L. Knizhnerman, 2010, Solution of 3D time-domain electromagnetic problems using optimal subspace projection: Geophysics, **76**, no. 6, F339–F351, doi: 10.1190/geo2011-0088.1.