GEOPHYSICS, VOL. 73, NO. 1 (JANUARY-FEBRUARY 2008); P. F1–F8, 14 FIGS., 1 TABLE.
10.1190/1.2799093

Time-domain modeling of electromagnetic diffusion with a frequency-domain code

Wim A. Mulder[1], Marwan Wirianto[2], and Evert C. Slob[2]

ABSTRACT

We modeled time-domain EM measurements of induction currents for marine and land applications with a frequency-domain code. An analysis of the computational complexity of a number of numerical methods shows that frequency-domain modeling followed by a Fourier transform is an attractive choice if a sufficiently powerful solver is available. A recently developed, robust multigrid solver meets this requirement. An interpolation criterion determined the automatic selection of frequencies. The skin depth controlled the construction of the computational grid at each frequency. Tests of the method against exact solutions for some simple problems and a realistic marine example demonstrate that a limited number of frequencies suffice to provide time-domain solutions after piecewise-cubic Hermite interpolation and a fast Fourier transform.

INTRODUCTION

Controlled-source EM measurements of induction currents in the earth can provide resistivity maps for geophysical prospecting. In marine environments, the current source often uses one or a few frequencies. In shallow seawater or on land, the response of air is dominant, and time-domain measurements are more appropriate. Because EM signals in the earth are strongly diffusive, direct interpretation of measured data can be difficult. Inversion of the data for a resistivity model may provide better results. We therefore need an efficient modeling and inversion algorithm.

For time-domain modeling, there are a number of options. The simplest method uses explicit time stepping, but this is rather costly. The Du Fort-Frankel (1953) method is more efficient, but it involves an artificial light-speed term. Implicit methods can compete only if a fast solver is available. Haber et al. (2002, 2004) provide examples for time-domain modeling. Druskin and Knizhnerman (1994) propose a technique based on Lanczos reduction and matrix exponentials. Obviously, the Fourier transform of results from a frequency-domain code can also provide time-domain solutions. Newman et al. (1986) present examples for horizontally layered media. For general resistivity models, a finite-difference, finite-volume, or finite-element discretization of the governing equations requires an efficient solver. The multigrid method (Mulder, 2006, 2007a) allows for a reasonably fast solution of the discretized equations when used as a preconditioner for BiCGStab2 (Van der Vorst, 1992; Gutknecht, 1993), a conjugate-gradient iterative method. With stronger grid stretching, we can apply a more robust multigrid variant based on semicoarsening and line relaxation (Mulder, 2007b).

Here we compare the computational cost of these methods by complexity analysis. The complexity of an algorithm measures its computational cost in terms of the number of unknowns — in this case, the electric field components on a grid. Because it ignores the constants that define the actual run time of a code on a computer, complexity analysis provides a crude way of comparing algorithms. Our analysis suggests that the frequency-domain approach is attractive. Next, we describe the issues arising when using a frequency-domain code for time-domain modeling. These involve the choice of frequencies, the choice of the discretization grid at each frequency, interpolation of earlier results to obtain a good initial guess and thereby accelerate the convergence of the solution, and the need for a robust solver. We present a number of examples to illustrate the method's performance.

COMPUTATIONAL COMPLEXITY

There are various methods for the numerical modeling of transient EM signals. Here we consider an explicit time-stepping scheme, the Du Fort-Frankel method, implicit schemes, matrix exponentials and Lanczos reduction, and the Fourier transform of frequency-domain solutions. Complexity analysis provides a cost estimate of a numeri-

Manuscript received by the Editor 11 December 2006; revised manuscript received 13 August 2007; published online 13 November 2007.
[1]Shell International Exploration and Production B. V., Rijswijk, The Netherlands, and Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Geotechnology, Delft, The Netherlands. E-mail: wim.mulder@shell.com.
[2]Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Geotechnology, Delft, The Netherlands. E-mail: m.liam@tudelft.nl; ec.slob@tudelft.nl.

cal method in terms of the number of unknowns, without the constants that determine actual computer run time. If N is the number of unknowns, the required computer time can be expressed as $Cf(N)$, where $f(N)$ describes the dependence on the number of unknowns. The constant C is determined primarily by the algorithm but also by its implementation and by the specific hardware. The determination of C requires tedious counting of operations. Alternatively, we can determine the constant by implementing the algorithm and running the code.

Complexity analysis derives an expression for $f(N)$ and can serve as a crude tool to distinguish between the cost of various algorithms. However, some algorithms have a very bad $f(N)$ but still perform quite well on practical problems. A well-known example is Dantzig's 1947 simplex algorithm (Dantzig, 1963) for linear-programming problems, which has an exponential complexity with $f(N) = O(2^N)$ but still is quite efficient in many cases.

We review the complexity of various time-domain methods for a 3D problem with $N = O(n^3)$ unknowns, where n is the number of grid points in each coordinate. An explicit time-stepping scheme is the simplest to implement. Unfortunately, it is stable only if the time step $\Delta t \leq ch^2$, where c is a constant depending on the material properties and the discretization, and h is the smallest grid spacing used in the problem. We assume that $h = O(1/n)$. The cost of a single time step is $O(n^3)$, so the overall complexity for computing the solution over a given, fixed time span T is $(T/\Delta t)O(n^3) = O(n^5)$. In practice, this is too slow for practical purposes, except perhaps on massively parallel computers.

The Du Fort-Frankel (1953) method offers one way to get around the restrictive stability limit. An artificial light speed is introduced with size $h/(\Delta t\sqrt{2})$ that allows the time step to grow with the square root of time, without doing too much harm to the accuracy of the solution. Geophysical applications of this method to time-domain EM problems can be found in, for instance, papers by Oristaglio and Hohmann (1984) for the 2D case and Wang and Hohmann (1993) and Commer and Newman (2004) for 3D problems. Maaø (2007) presents an interesting variant. The cost of the Du Fort-Frankel method is $O(n^4)$, as shown in Appendix A.

An implicit scheme can avoid the $O(h^2)$ stability limit as well. The price paid is the solution of a large sparse linear system, which may be costly. With a sufficiently powerful solver, one or a few iterations can be enough (Haber et al., 2002, 2004). For $O(n^0) = O(1)$ iterations, the cost of solving the time-domain equations is $O(n^3)$ per time step. Together with a time step that scales with the square root of time, this method has the same complexity as the Du Fort-Frankel scheme, although the cost per step will be larger by at least an order of magnitude because of the work required by the iterative solver. The method does not require an artificial light-speed term, which may allow for larger time steps without harming the accuracy.

Druskin and Knizhnerman (1994) and Druskin et al. (1999) propose a technique that appears to be attractive for 3D applications. They apply the Lanczos method to reduce the original sparse matrix $\mathbf{A}$ that describes the linear problem to a dense but much smaller matrix. The latter is used to quickly compute the time evolution using matrix exponentials. Remis (1998) also investigates this method.

The Lanczos method constructs the small matrix iteratively. Druskin and Knizhnerman (1994) show that accurate results can be obtained by performing m iterations, where $m = O(n\sqrt{T \log n})$. As before, T is the length of time for which the solution needs to be computed, and n is the number of grid points in one of the spatial coordinates. Because the number of nonzero elements of $\mathbf{A}$ for a 3D problem is $O(n^3)$, the cost of the Lanczos decomposition is $n^4\sqrt{\log n}$ for a given T.

One can compute time-domain solutions by first selecting a number of frequencies, then solving the frequency-domain problem at those frequencies, and finally performing an inverse Fourier transform to the time domain. For n_f frequencies and assuming the availability of an efficient solver that requires $O(1)$ iterations, the complexity is $O(n_f n^3)$.

Comparison of the above methods shows that two of them have an asymptotic complexity of $O(n^4)$: the Du Fort-Frankel method and an implicit scheme with an optimal solver that convergences in $O(1)$ iterations. The method based on Lanczos reduction has an additional logarithmic factor, which in practical applications may be small enough to be neglected. The application of a frequency-domain method with an optimal solver results in a complexity of $O(n_f n^3)$, which can be favorable if n_f is small relative to n.

These are only asymptotic results. In practice, performance depends on the details of the implementation and the actual constants in the complexity estimates.

The choice of grid is another topic. Diffusion problems typically have length scaling with the square root of time. This implies that accurate modeling of a problem with a pointlike source in space and time requires an initial grid that is very fine close to the source, gradually becoming less fine. Dynamic local adaptive grid refinement will accomplish this, but it leads to complicated software. Also, the Lanczos decomposition cannot be used with dynamic adaptive grid refinement. In the Fourier domain, the computational grid should depend on skin depth and therefore on frequency (Plessix et al., 2007). Each frequency requires a different grid, but that is more easily accomplished than time-dependent adaptive local grid refinement.

Although it remains to be seen which of the four methods requires the least computer time for a given accuracy, the frequency-domain approach appears to be attractive.

METHOD

In this section, we summarize the governing equations and their discretization. Next, we review the multigrid solver (Mulder, 2006, 2007a). Because the standard approach breaks down on stretched grids, a variant based on semicoarsening and line relaxation was designed (Jönsthövel et al., 2006; Mulder, 2007b). The version described by Jönsthövel et al. (2006); is slow. We therefore accelerate the line relaxation by a nonstandard Cholesky decomposition. Then we describe the automatic selection of frequencies, followed by a discussion on how the skin depth at each frequency determines the computational grid.

The Maxwell equations and Ohm's law for conducting media in the frequency domain are

$$\imath\omega\mu_0\tilde{\sigma}\hat{\mathbf{E}} - \nabla\times\mu_r^{-1}\nabla\times\hat{\mathbf{E}} = -\imath\omega\mu_0\hat{\mathbf{J}}_s. \qquad (1)$$

The vector $\hat{\mathbf{E}}(\omega,\mathbf{x})$ represents the electric field components as a function of angular frequency ω and position $\mathbf{x}$. The current source is $\hat{\mathbf{J}}_s(\omega,\mathbf{x})$. The quantity $\tilde{\sigma}(\mathbf{x}) = \sigma - \imath\omega\epsilon_0\epsilon_r$, with $\sigma(\mathbf{x})$ the conductivity, $\epsilon_r(\mathbf{x})$ the relative permittivity, $\mu_r(\mathbf{x})$ the relative permeability, and ϵ_0 and μ_0 their absolute values in vacuum. We adopt the Fourier convention

$$\mathbf{E}(t,\mathbf{x}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{\mathbf{E}}(\omega,\mathbf{x})e^{-\iota\omega t}d\omega,$$

using SI units in the examples.

Mulder (2006, 2007a) presents a numerical method for solving the system of equations 1 in the frequency domain. The finite-integration technique (Weiland, 1977) provided a finite-volume discretization of the equations. A multigrid solver acted as preconditioner for the iterative BiCGStab2 scheme (Van der Vorst, 1992; Gutknecht, 1993). On an equidistant grid, the method converged in a fixed number of steps, independent of the number of grid points. With grid stretching, however, we lost this $O(1)$ number of iterations. The more severe the stretching, the larger the number of iterations. In addition, the number increased for a larger number of grid points. The grid stretching is necessary because we have to include artificial boundaries when we truncate the computational domain to a finite size.

The use of semicoarsening and line relaxation led to a more robust solver (Jönsthövel et al., 2006). During a single multigrid cycle, the grid was coarsened in only two of the three coordinate directions. We applied the line relaxation in the same two directions. The direction that was not coarsened alternated among the three coordinate directions between subsequent multigrid cycles. We always applied three multigrid cycles as a single preconditioning step for BiCGStab2 to ensure the invariance of the preconditioner. For the results in that paper, we used a generic subroutine for solving complex-valued band matrices. Here, we replaced this routine by a nonstandard Cholesky decomposition.

The standard decomposition factors a hermitian matrix $\mathbf{A}$ into $\mathbf{LL^H}$, where $\mathbf{L}$ is a lower triangular matrix and $\mathbf{L^H}$ is its complex conjugate transpose. In our case, the finite integration technique provides a matrix $\mathbf{A}$ that is not hermitian but complex valued and symmetric: $\mathbf{A} = \mathbf{A^T}$, where the superscript $^\mathbf{T}$ denotes the transpose. The nonstandard Cholesky decomposition factors the matrix into $\mathbf{LL^T}$. In the line relaxation scheme, $\mathbf{A}$ is a band matrix with 11 diagonals. We only need its main diagonal and five lower diagonal elements. The Cholesky decomposition replaces this matrix by $\mathbf{L}$, also containing six diagonals. We increased speed by a factor of about seven after replacing the generic band matrix solver by the nonstandard Cholesky decomposition.

Here, we used the simpler solver when the grid stretching was mild and the more robust solver when the grid stretching was more severe. On equidistant or mildly stretched grids, the number of BiCGStab2 iterations required to solve the equations at a given frequency is typically around four with the standard multigrid method, independent of the number of unknowns. One BiCGStab2 iteration involves two multigrid preconditioning steps. The more powerful method based on semicoarsening and line relaxation is less sensitive to grid stretching, but the required computer time per full BiCGStab2 iteration is almost 11 times larger. The number of BiCGStab2 iterations is typically around two. One iteration now involves two preconditioning steps consisting of six multigrid cycles.

The time-domain solutions require a large number of frequencies. We chose an adaptive approach. Following, for instance, Newman et al. (1986) and Gupta et al. (1989), we selected frequencies f_k on a logarithmic grid: $f_k = 10^{q_k}$, with $q_k = q_0 + k\Delta q, k = 0,\ldots,n_f - 1$. The frequency-domain solutions provided the electric-field components at the receivers. Shape-preserving piecewise-cubic Hermite interpolation (Fritsch and Carlson, 1980) mapped these data points to an equidistant grid of frequencies. A fast Fourier transform (FFT) provided the response in the time domain.

Instead of using an equidistant grid of values for q_k, an adaptive selection of values will reduce the required computer time. Given a minimum and maximum frequency $f_{\min}$ and $f_{\max}$, we chose values of $q = q_0 + m\Delta q^{(0)}$, $m = 0,\ldots,M-1$, where $q_0 = \log_{10} f_{\min}$ and $\Delta q^{(0)} = (M-1)^{-1}\log_{10}(f_{\max}/f_{\min})$. We then computed the solutions for these M frequencies and stored them on disk. This included both the full solutions on the computational grid and the recorded electric-field components at the receivers. Next, we selected frequencies at an interval of $1/2\Delta q^{(0)}$ in an adaptive manner.

Figure 1 explains one step of the adaptive frequency-selection procedure. The dots represent the real part of the computed electric-field component in the x-direction. Suppose we have results for q from -2 to 2 at an interval of $\Delta q = 0.5$. We remove one point — for instance, at $q = -0.5$ — and perform shape-preserving piecewise-cubic Hermite interpolation through the remaining points, resulting in the gray curve in Figure 1. The value at $q = -0.5$, marked by the square, is different from the actual value indicated by the dot. If the difference exceeds a certain threshold, such as 1% of the maximum absolute value in the plot, we select values of q for the next finer level with $\Delta q = 0.25$ on the left and on the right side of $q = -0.5$, so at $q = -0.75$ and $q = -0.25$.

We repeat this procedure for the real and imaginary part of all electric field components at all receivers for all values of $q = -1.5,-1.0,\ldots,1.0,1.5$ of the original set, excluding end points. This will produce a set of new q-values on a grid with a spacing of $\Delta q = 0.25$. We then solve the problem at those frequencies. New frequencies are selected in the same way as before to find q-values on a grid with a spacing of $\Delta q = 0.125$. We repeat this process until the differences between interpolated and computed values are smaller than the given tolerance.

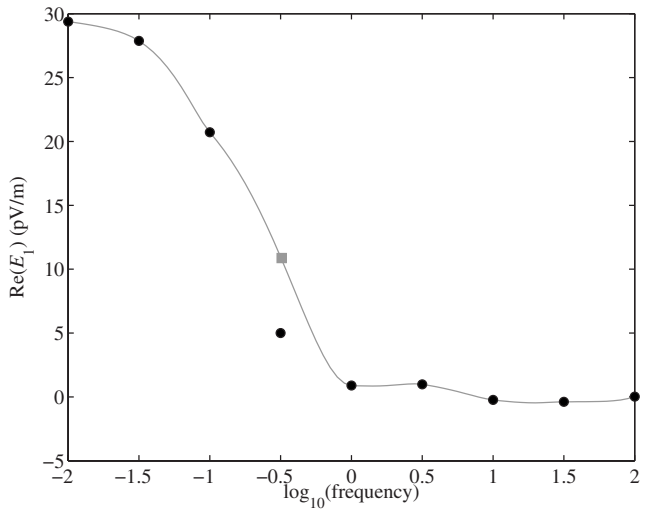

Figure 1. Example displaying the adaptive selection of frequencies. The black dots represent the real part of the horizontal component of the electric field at a given receiver for an equidistant grid of $q = \log_{10} f$ between -2 and 2 with spacing $\Delta q = 0.5$ (f is the frequency in hertz). We can remove one point, here at $q = -0.5$, and predict its value by shape-preserving piecewise-cubic Hermite interpolation through the remaining points. If the difference between the interpolated (gray square) and actual (black dots) values is too large, we add two new frequencies by selecting $q = -0.5 \pm 1/2\Delta q$. In this case, $q = -0.75$ and $q = -0.25$.

To describe this procedure in a general way, we define $\Delta q^{(l)} = \Delta q^{(0)}/2^l, l = 1,2,\ldots,l_{\max}$. Suppose we move to a new level l, with previously obtained solutions at smaller values of l available on disk. We select an existing solution at some frequency defined by $q_m^{(l')}$, $l' < l$, and determine a prediction for the receiver data by shape-preserving piecewise-cubic Hermite interpolation based on the frequencies 10^q rather than q itself. For this interpolation, we exclude the result at $q_m^{(l')}$. If the difference between the interpolated and actual value exceeds a prescribed tolerance, we select the two neighboring frequencies at $q_m(l') \pm \Delta q^{(l)}$, except at the end points where only the one inside the defined range of q is taken. In this way, we find a number of frequencies on level l and then compute solutions for these frequencies.

We could reduce the number of iterations for the frequency-domain method by determining an initial guess based on cubic Lagrange interpolation from existing solutions at the four nearest frequencies. Here, *nearest* refers to distance on a logarithm scale. If solutions for less than four frequencies were available, we switched to lower-order interpolation.

The physics of the problem dictate that at high frequencies, only a small portion of the earth affects the recorded electric field; at lower frequencies, a larger part of the earth is seen. The length scale at a frequency f is controlled by the skin depth $\Delta_s = 1/\sqrt{\pi f \mu \sigma}$, where σ is the conductivity and μ the magnetic permeability. If we take the vacuum value $\mu_0 = 4\pi 10^{-7}$ H/m for the latter, we obtain the well-known expression $\Delta_s = 503/\sqrt{\sigma f}$, all in SI units.

Several conflicting requirements guide the choice of the grid (Plessix et al., 2007). First, numerical accuracy requires three to eight points per skin depth. Second, the grid should be sufficiently fine to honor the details of the resistivity model close to the source and the receivers. Third, a point-dipole or finite-length line source generates a singular solution. For receivers at a short distance from the source, the singularity must be resolved with sufficient accuracy, requiring a fine grid. For receivers further away, the solution can have sufficient accuracy without resolving the details of the singularity, thereby requiring a less fine grid around the source. Finally, as we use perfect electric conductor boundary conditions, a boundary strip of about five skin depths is added around the model to avoid undesirable boundary effects. For the air layer, an even thicker layer is added.

The well-known primary-secondary formulation may offer an advantage in some cases. If we abbreviate equation 1 as $\mathbf{L}\hat{\mathbf{E}} = \hat{\mathbf{f}}$, we can split the linear operator into $\mathbf{L} = \mathbf{L}_p + \mathbf{L}_s$ and the solution into $\hat{\mathbf{E}} = \hat{\mathbf{E}}_p + \hat{\mathbf{E}}_s$ such that $\mathbf{L}_p\hat{\mathbf{E}}_p = f$ and $\mathbf{L}_p$ can easily be solved. The secondary solution then should obey $\mathbf{L}\hat{\mathbf{E}}_s = -\mathbf{L}_s\hat{\mathbf{E}}_p$. If the secondary problem has the same relative magnetic permeability as the primary problem, then $\mathbf{L}_s = \iota\omega\mu_0(\tilde{\sigma} - \tilde{\sigma}_0)$.

The secondary problem is as difficult to solve as the original one, but the advantage is a potentially more accurate solution. If the source resembles a delta function, the solution will be singular close to the source. If a receiver is located close to the source, a rather fine grid is required to resolve the singular behavior of the electric field. If the formation has a conductivity $\tilde{\sigma}_0$ around the source and $\mathbf{L}_s = \iota\omega\mu_0(\tilde{\sigma} - \tilde{\sigma}_0)$ is nonzero sufficiently far away from the source and does not have the character of an isolated point scatterer, the secondary field generally will be less singular. In that case, we can use a different grid that does not require very small cells close to the secondary source.

EXAMPLES

Here we present examples that highlight some of the issues.

Homogeneous formation

The first example is a point-current source $\mathbf{J}_s = \mathbf{j}_s\delta(\mathbf{x})\delta(t)$, $\mathbf{j}_s = (1,0,0)^{\mathbf{T}}$ A m s, in a homogeneous formation with a conductivity of $\sigma = 1$ S/m. We computed frequency-domain solutions on a grid that was adapted to the skin depth and finest near the source. We applied power-law grid stretching (Mulder, 2006) away from the source. The grid was different for each frequency. The BiCGStab2 iterations stopped as soon as the norm of the residual dropped below 10^{-6} times the norm of the residual obtained for a zero electric field. Figure 2 shows the real and imaginary parts of E_1, the x-component of the electric field, measured by a single receiver at 900 m distance from the source at the same depth. The computational grid had 128^3 cells.

First, we computed solutions at five frequencies $f = 10^q$ Hz, with $q = -2,-1,0,1,2$, so $\Delta q = 1$. We set the initial values for the electric fields to zero. Next, the computed values of the electric field recorded at the receiver for each frequency f_m were compared to a prediction based on piecewise-cubic Hermite interpolation, using the values at the other frequencies f_k and excluding the one for which the prediction was made ($k \neq m$). If the relative difference between the interpolated and actual value exceeded 1%, frequencies at $q = q_m \pm 1/2\Delta q$ were selected for the next level of computations. The circles in Figure 2 show that all four intermediate values $q = -1.5,-0.5,0.5,1.5$ were included. We then determined initial values for the electric fields from cubic interpolation of the solutions for the four frequencies nearest to the current one. Next, the relative difference between interpolated and computed receiver values was considered again for all available frequencies, and new neighboring values for q at a spacing of $\Delta q = 1/4$ were selected if the relative difference exceeded 1%. Figure 2 shows that all new values between -1 and 2 were selected.

This procedure was repeated until all relative differences were less than 1%. At $\Delta q = 1/8$, only nine new frequencies were added

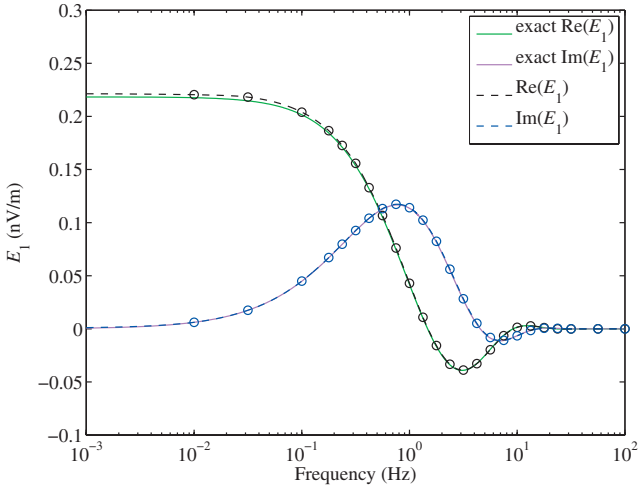

Figure 2. Real (black) and imaginary (blue) part of E_1 for various frequencies. The circles indicate the computed values; the lines were determined by shape-preserving piecewise-cubic interpolation. The real and imaginary parts of the exact solution are drawn as well.

and at $\Delta q = 1/16$ no new ones were needed. Figure 2 shows the resulting values for the inline electric field component E_1 as circles, together with the curves obtained by interpolation and the exact solution, which can be found in, for instance, Ward and Hohmann (1987).

Table 1 lists iteration counts for the various frequencies. Note that BiCGStab2 can terminate halfway through a full iteration, hence the half counts. The parameter α measures the amount of grid stretching. The maximum ratio between the widths of neighboring cells is $1 + \alpha$. If α exceeded 0.04, we switched to the more expensive multigrid preconditioner based on line relaxation and semicoarsening. For the latter, iteration counts and measured CPU times are marked by an asterisk. The results are listed in the order in which they were computed. The effect of using interpolated values as the initial guess instead of zero values can be deduced from the iteration counts farther down in the table. The speedup is not dramatic, but it helps.

The data points were interpolated by piecewise-cubic Hermite interpolation (Fritsch and Carlson, 1980) to an equidistant grid of frequencies and transformed to time by an FFT. A comparison to the exact time-domain solution (Ward and Hohmann, 1987) is shown in Figures 3 and 4. The errors are largest at early and late times, because of a lack of the lowest and highest frequencies. Also, there is a difference between the peak values of about 1% visible in Figure 3.

Table 1. Iteration counts at various frequencies. The asterisk denotes counts obtained for the more expensive multigrid variant with line relaxation and semicoarsening. The parameter α measures the amount of grid stretching. The required CPU time in seconds is included.

Δq	q	f (Hz)	α	Iterations	CPU (s)
1	2	100	0.032	3.5	292
	1	10	0.010	3.5	301
	0	1	0.022	3.5	303
	−1	0.1	0.045	1.5*	1393*
	−2	0.01	0.069	2.0*	1879*
0.5	1.5	31.6	0.020	4.0	344
	0.5	3.16	0.012	3.5	304
	−0.5	0.316	0.034	7.0	599
	−1.5	0.0316	0.057	1.5*	1419*
0.25	1.75	56.2	0.026	3.0	259
	1.25	17.8	0.015	3.5	304
	0.75	5.62	0.0076	3.0	260
	0.25	1.78	0.017	3.0	259
	−0.25	0.562	0.028	5.0	429
	−0.75	0.178	0.040	8.5	739
	−1.25	0.0562	0.051	1.0*	927*
	−1.75	0.0178	0.063	1.0*	931*
0.125	1.375	23.7	0.018	3.0	264
	1.125	13.3	0.0013	2.5	215
	0.875	7.50	0.0082	2.5	217
	0.625	4.22	0.0099	3.0	262
	0.375	2.37	0.015	2.0	174
	0.125	1.33	0.020	3.0	259
	−0.125	0.750	0.025	3.5	307
	−0.375	0.422	0.031	5.0	430
	−0.625	0.237	0.037	6.5	562

Scatterer in a homogeneous formation

The next example is a resistive scatterer in a homogeneous background with a conductivity of 1 S/m. A rectangular scatterer with x between −300 and 300 m, y between −200 and 200 m, and z between 400 and 600 m has a conductivity of 0.1 S/m. Figure 5 displays the layout. The source is the same as in the previous example. The grid, however, is different; it is equidistant inside the scatterer, and hyperbolic cosine stretching (Mulder, 2006) is applied away from the object. In this case, we used a primary-secondary formula-

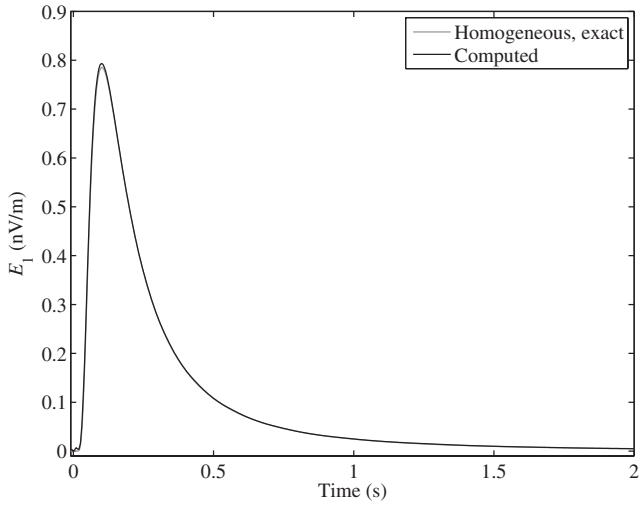

Figure 3. Time-domain solution for the homogeneous problem. The black curve represents the numerical solution of the inline component of the electric field; the gray one is the exact solution. The peak value has an error of about 1%.

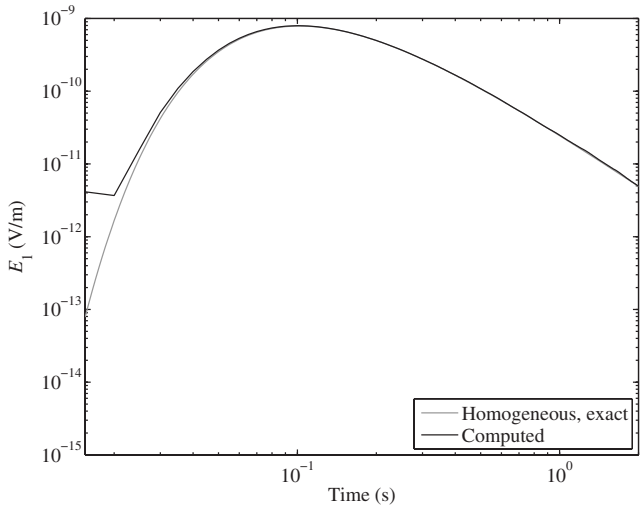

Figure 4. The same time-domain solution for the homogeneous problem as in the previous figure, but now on a logarithmic scale.

tion in which the homogeneous response is subtracted so that the source term and its singular response are replaced by a source term that involves the exact solution.

Figure 6 displays the secondary frequency-domain solution for a source at the origin and a receiver located at (900,0,0) m and computed on a grid with 128^3 cells. For comparison, we computed the full electric field for the homogeneous medium with the scatterer and subtracted the numerical solution for the homogeneous medium without the scatterer. Figure 7 shows the difference. We subtracted the numerical primary field so its numerical errors in both computations cancel, even when large. This explains the small differences between the figures. Note that the adaptive procedure selected frequencies for the full field that are different from those for the secondary field. The reason is that the primary solution dominates the full field.

Figure 8 shows the time-domain response of the secondary scattered field for the primary-secondary formulation.

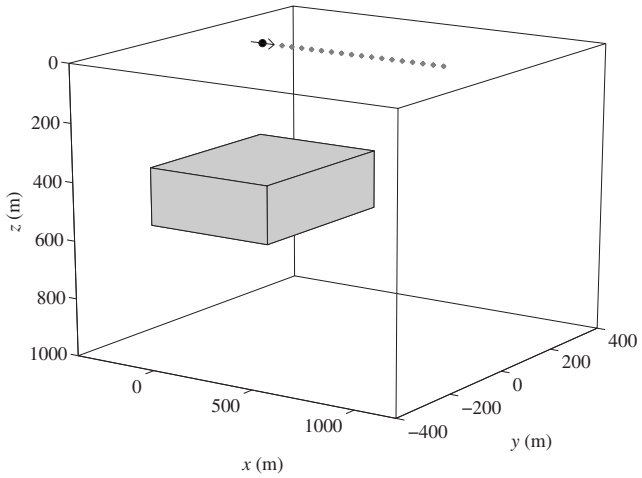

Figure 5. Resistive scatterer in a homogeneous formation. The arrow and black dot mark the point-current source; the other dots indicate the receiver positions.

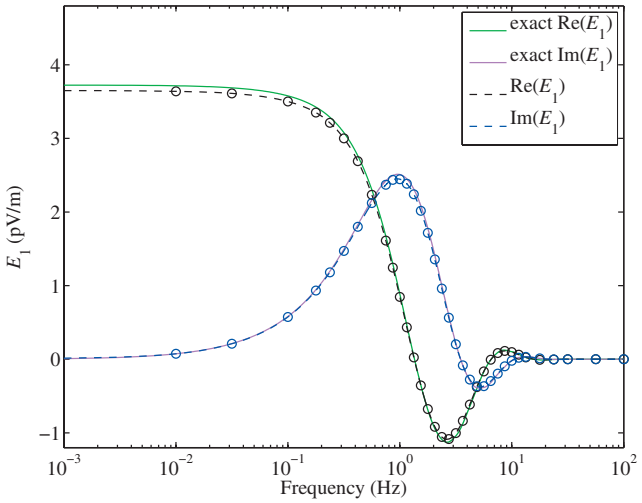

Figure 6. The secondary solution in the frequency domain for the inline electric field; the circles and lines are defined in Figure 3.

Three layers

A slightly less trivial test problem consists of three layers: air, water, and sediments. These layers have a conductivity σ of 0, 3, and 0.5 S/m, respectively, and a relative permittivity ϵ_r of 1, 80, and 17. The water depth is 200 m. A dipole source in the x-direction at a depth of 175 m generates the inline field shown in Figures 9 and 10 for a frequency of 0.5 Hz. The figures demonstrate that the code provides reasonably accurate answers in this case.

Shallow marine problem

The SEG/EAGE salt model (Aminzadeh et al., 1997) served as a template for a realistic subsurface model. This model was designed for simulating seismic wave propagation and contains a complex salt body surrounded by sediments. The seawater has depths around 120 m. Its dimensions are $13{,}500 \times 13{,}480 \times 4680$ m. We replaced the seismic velocities of the model by resistivities (σ^{-1}). For the water velocity of 1500 m/s, we chose a resistivity of 0.3 Ωm. Veloci-

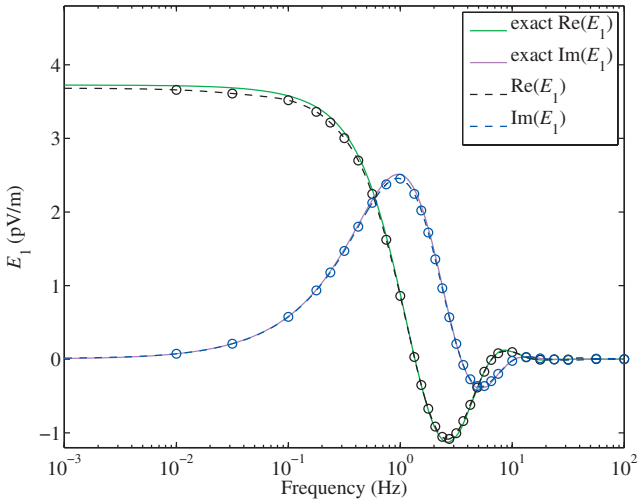

Figure 7. Response for the inline electric field, obtained by taking the difference between the full numerical solutions with and without a scatterer.

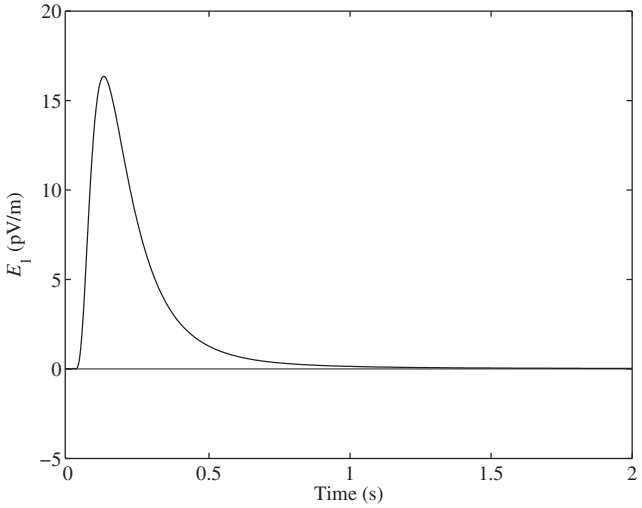

Figure 8. The time-domain secondary solution for the scatterer computed with the primary-secondary formulation on a grid with 128^3 cells.

ties above 4000 m/s, indicative of salt, were replaced by 30 Ωm. Basement, beyond 3660 m depth, was set to 500 Ωm. We determined the resistivity of the sediments by $(v/1700)^{3.88}$ Ωm, with the velocity v in meters per second. The paper of Meju et al. (2003) motivated this choice. For air, we used a resistivity of 10^8 Ωm. Figure 11 displays the resistivity on a logarithmic scale.

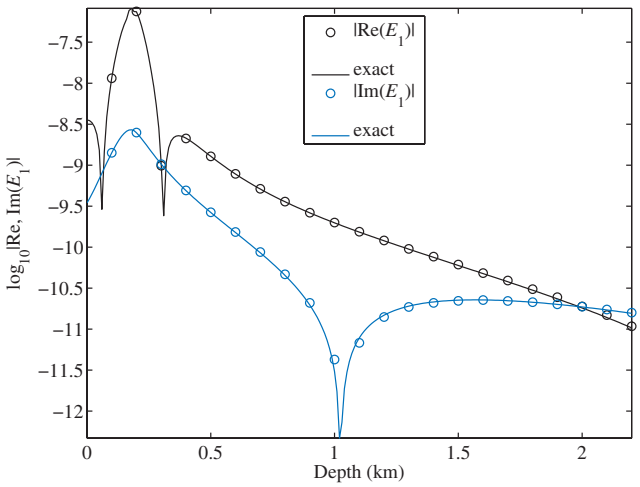

Figure 9. Inline electric field at various depths at a horizontal inline distance of 100 m from the source.

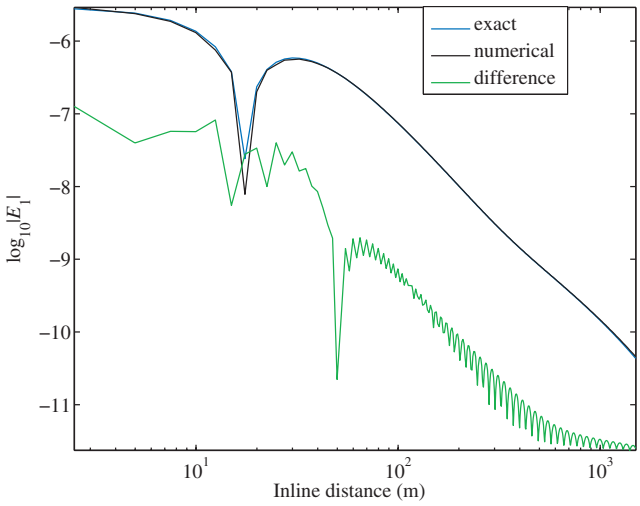

Figure 10. The inline electric field on the sea bottom.

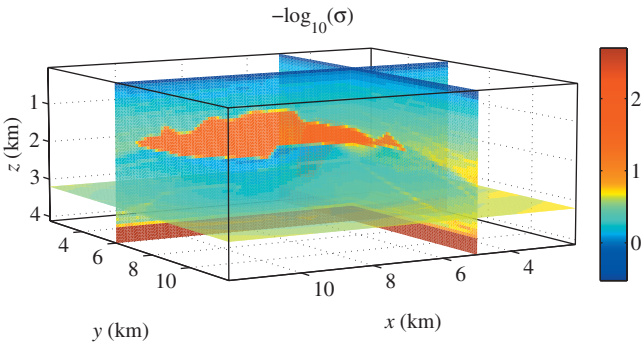

Figure 11. Logarithm of the resistivity ($\log_{10}\sigma^{-1}$, SI units) for a model with a salt body.

We positioned a finite-length current source between (6400, 6500, 50) and (6600,6500,50) m. The receivers were placed on the sea bottom. Initial solutions were computed at frequencies of 10^q Hz, with q between −2.5 and 2.5 at an increment of 0.5. The adaptive scheme added more frequencies where needed. As before, cubic interpolation or extrapolation of solutions for other frequencies provided an initial guess for the iterative solution method. The spatial grid was again based on a balance between the skin depth at the given frequency and the details of the model. In the water layer, the grid was equidistant in the vertical direction, and we used power-law stretching away from the surface and the maximum depth of the water layer. In the horizontal directions, we applied power-law stretching away from the center of the source.

Figure 12 shows one of the frequency-domain solutions, and Figure 13 displays the time-domain response. The airwave shows up as an early peak. Of course, the air interface also affects diffusion fronts that come in later. The anticausal part must be caused by missing high frequencies and numerical errors in the higher frequencies.

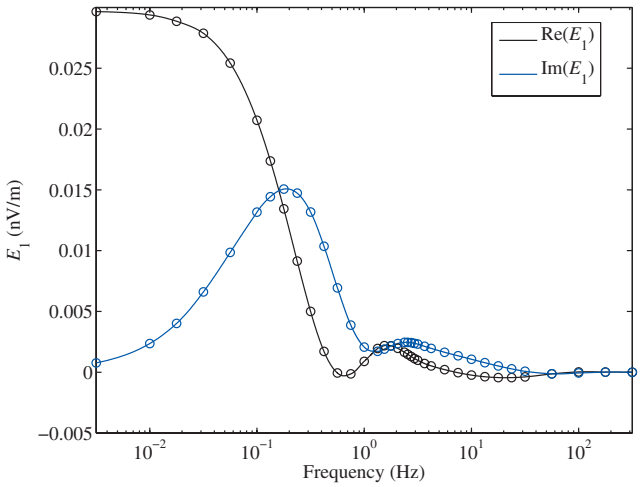

Figure 12. Frequency response for E_1, for a source at (6500,6500, 50) m and a receiver at (9000,6500,100) m on the sea bottom. See Figure 3 for an explanation of the circles.

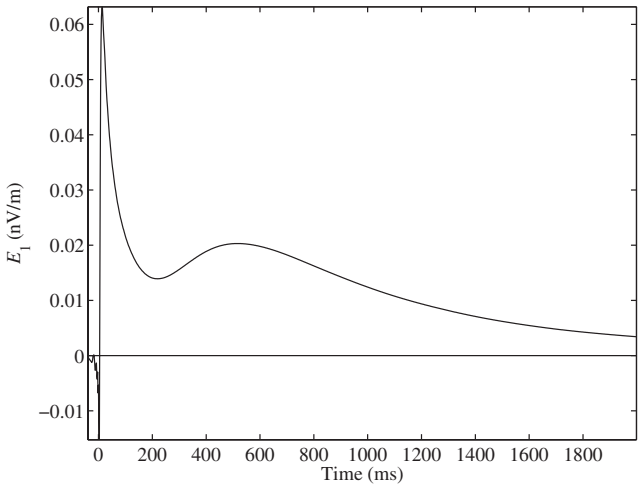

Figure 13. Time response for the inline field component E_1 for a 200-m finite-length source centered at (6500,6500,50) m and a receiver at (9000,6500,90) m on the sea bottom.

CONCLUSIONS

Complexity analysis of time-domain methods for modeling EM diffusion shows that some popular methods have an $O(n^4)$ complexity, where n is the number of points per spatial coordinate. Synthesizing time-domain solutions by using a frequency-domain method has a complexity of $O(n_f n^3)$, with n_f the number of frequencies if the solver converges in a fixed number of iterations. We accomplished this on stretched grids with a multigrid variant based on line relaxation and semicoarsening. On uniform or mildly stretched grids, we used a simpler multigrid scheme.

When the number of frequencies n_f is small relative to n, this frequency-domain method appears to be attractive. However, because our complexity analysis only provides estimates in terms of the number of unknowns and the actual required computer time also depends on the constants in the estimates, a true comparison of methods should involve the operation count or CPU time measured for an actual implementation. Also, n_f might become as large as the number of time steps required for an implicit time-domain code for complex resistivity models. Furthermore, if early times are not recorded and the receivers are not too close to the source, the initial time-step size can be relatively large, leading to a smaller number of time steps. Note that the time-domain computations do not require complex arithmetic. Therefore, an implicit method may compete or even be more efficient. In the frequency domain, however, it is easier to adapt the grid to the characteristic length scales of the solution.

The included examples show how frequencies can be selected and how time-domain solutions can be obtained by monotone piecewise-cubic Hermite interpolation and an FFT.

ACKNOWLEDGMENTS

M. Wirianto received financial support from the sponsors of the Delphi consortium.

APPENDIX A

TIME STEP INCREASING WITH THE SQUARE ROOT OF TIME

For accuracy reasons, we let the time step grow proportional to the square root of time. An explicit time-stepping scheme has $\Delta t_{\text{expl}} = C\mu\sigma h^2$ for the diffusive case, where the $O(1)$ constant C depends on the number of spatial dimensions. Here, μ is the magnetic permeability and σ the conductivity. If the first time step is chosen to be the same as for an explicit scheme, then we have $\Delta t = \sqrt{\Delta t_{\text{expl}} t}$. The time interval after k steps is denoted by t_k.

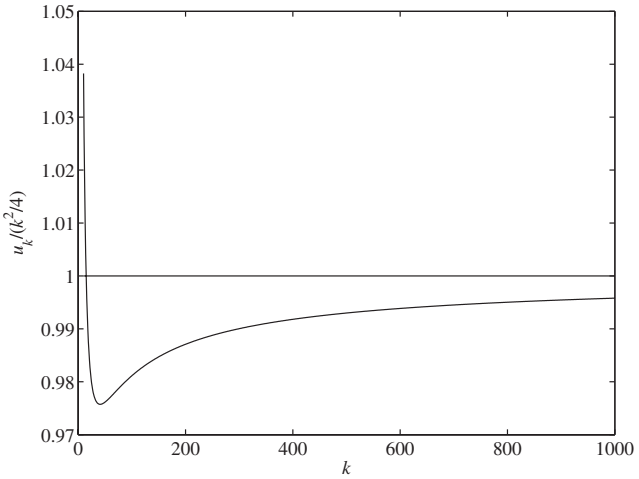

Figure A-1. The graph of $u_k/(k^2/4)$ as a function of k shows that $u_k \sim k^2/4$ for large k.

The above choices imply $t_0 = 0$, $t_1 = \Delta t_{\text{expl}}$, and $t_{k+1} = t_k + \sqrt{\Delta t_{\text{expl}} t_k}$ for $k \geq 1$. Let $t_k = \Delta t_{\text{expl}} u_k$. Then $u_1 = 1$ and $u_{k+1} = u_k + \sqrt{u_k}$, resulting in $u_2 = 2$, $u_3 = 2 + \sqrt{2}$, and so on. One can see from Figure A-1 that $u_k \sim k^2/4$ for large k. A time span T will require $n_t \simeq \sqrt{4T/\Delta t_{\text{expl}}}$ time steps. Using $h = O(1/n)$, we obtain $n_t \sim O(n)$ and an overall cost of $O(n^4)$ for 3D problems.

REFERENCES

Aminzadeh, F., J. Brac, and T. Kunz, 1997, 3-D salt and overthrust models: SEG.

Commer, M. A., and G. A. Newman, 2004, A parallel finite-difference approach for 3D transient electromagnetic modeling with galvanic sources: Geophysics, **69**, 1192–1202.

Dantzig, G. B., 1963, Linear programming and extensions: Princeton University Press.

Druskin, V. L., and L. A. Knizhnerman, 1994, Spectral approach to solving three-dimensional Maxwell's diffusion equations in the time and frequency domains: Radio Science, **29**, 937–953.

Druskin, V. L., L. A. Knizhnerman, and P. Lee, 1999, New spectral Lanczos decomposition method for induction modeling in arbitrary 3D geometry: Geophysics, **64**, 701–706.

Du Fort, E. C., and S. P. Frankel, 1953, Stability conditions on the numerical treatment of parabolic differential equations: Mathematical Tables and Other Aids to Computation (Mathematics of Computation), **7**, 135–152.

Fritsch, F. N., and R. E. Carlson, 1980, Monotone piecewise cubic interpolation: SIAM Journal on Numerical Analysis, **17**, 238–246.

Gupta, P. K., A. P. Raiche, and F. Sugeng, 1989, Three-dimensional time-domain electromagnetic modelling using a compact finite-element frequency-stepping method: Geophysical Journal International, **96**, 457–468.

Gutknecht, H. H., 1993, Variants of BiCGStab for matrices with complex spectrum: SIAM Journal on Scientific and Statistical Computing, **14**, 1020–1033.

Haber, E., U. M. Ascher, and D. W. Oldenburg, 2002, 3D forward modeling of time domain electromagnetic data: 72nd Annual International Meeting, SEG, Expanded Abstracts, 641–644.

———, 2004, Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach: Geophysics, **69**, 1216–1228.

Jönsthövel T. B., C. W. Oosterlee, and W. A. Mulder, 2006, Improving multigrid for 3-D electro-magnetic diffusion on stretched grids: Proceedings of the ECCOMAS European Conference on Computational Fluid Dynamics.

Maaø, F. A., 2007, Fast finite-difference time-domain modeling for marine-subsurface electromagnetic problems: Geophysics, **72**, no. 2, A19–A23.

Meju, M. A., L. A. Gallardo, and A. L. Mohamed, 2003, Evidence for correlation of electrical resistivity and seismic velocity in heterogeneous near-surface materials: Geophysical Research Letters, **30**, 1373–1376.

Mulder, W. A., 2006, A multigrid solver for 3D electromagnetic diffusion: Geophysical Prospecting, **54**, 663–649.

———, 2007a, Geophysical modelling of 3D electromagnetic diffusion with multigrid: Computing and Visualization in Science; http://dx.doi.org/10.1007/s00791-007-0064-y.

———, 2007b, A robust solver for CSEM modelling on stretched grids: 69th Meeting, EAGE, Extended Abstracts, D036.

Newman, G. A., G. W. Hohmann, and W. L. Anderson, 1986, Transient electromagnetic response of a three-dimensional body in a layered earth: Geophysics, **51**, 1608–1627.

Oristaglio, M. L., and G. W. Hohmann, 1984, Diffusion of electromagnetic fields into a two-dimensional earth: A finite-difference approach: Geophysics, **49**, 870–894.

Plessix, R.-E., M. Darnet, and W. A. Mulder, 2007, An approach for 3D multisource multifrequency CSEM modeling: Geophysics, **72**, no. 5, SM177–SM184.

Remis, R. F., 1998, Reduced-order modeling of transient electromagnetic field: Ph.D. thesis, Delft University of Technology.

Van der Vorst, H. A., 1992, BI-CGSTAB, a fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems: SIAM Journal on Scientific and Statistical Computing, **13**, 631–644.

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

Ward, S. A., and G. W. Hohmann, 1987, Electromagnetic theory for geophysical applications, *in* M. N. Nabighian, ed., Electromagnetic methods in applied geophysics — Theory, vol. 1: SEG, 131–311.

Weiland, T., 1977, A discretization method for the solution of Maxwell's equations for six-components fields: Electronics and Communications, **31**, 116–120.