

2D Time Domain TEM Simulation Using Finite Elements, an Exact Boundary Condition, and Krylov Subspace Methods

M. Afanasjew^{1,2}, R.-U. Börner¹, M. Eiermann², O. G. Ernst², S. Güttel^{2,3}, and K. Spitzer¹

¹Institute of Geophysics, TU Bergakademie Freiberg, Germany, ²Institute of Numerical Analysis and Optimization, TU Bergakademie Freiberg, Germany, ³now at: Section of Mathematics, Université de Genève, Switzerland

1 Summary

In this report we present a numerical method for solving Maxwell's equations in the time domain assuming an arbitrary two-dimensional conductivity distribution including an isolating air half-space. The method allows to carry out the computations for the subsurface only, because the air-earth interface is handled by an exact boundary condition. The spatial discretization is done with the finite element method, leading to a linear system of ordinary differential equations (ODE). We use state-of-the-art Krylov subspace methods for this ODE to advance a given initial electric field to selected times of interest. The presented theory is tested with some standard models and compared to a traditional finite difference time stepping implementation with respect to accuracy and efficiency. The results clearly demonstrate the superiority of the presented method in terms of run time given a comparable accuracy.

Keywords: Analytic boundary condition, finite element method, Krylov subspace, time domain, transient EM

2 Introduction

The transient electromagnetic (TEM) method has become a standard technique in geophysical prospecting during the past years. It is already in wide use, e. g., for the exploration of important resources like hydrocarbons, groundwater and minerals. One important aspect here is a reliable and computationally efficient simulation of the decaying electromagnetic field, which can be leveraged to get a better understanding of field behavior in complicated real-world settings as well as a building block in inversion schemes, that ultimately aim at resolving arbitrary conductivity structures from only a few well-placed measurements.

The predominant forward modeling technique in the literature is the finite difference time domain (FDTD) method, that was already introduced by Yee (1966). An explicit time-stepping technique, that already dealt with an isolating air half-space, was developed by Oristaglio and Hohmann (1984) for the two-dimensional case and later refined by Wang and Hohmann (1993) for three dimensions. Like for other explicit time-stepping methods, the size of the time steps, that the described Du Fort-Frankel scheme can stably perform, depends on the grid spacing and the lowest conductivity. Although the resistive air is already eliminated, thousands of time steps have to be performed although only a few dozen solutions are necessary to describe the decaying field.

The approach taken here is based on a finite element discretization, which allows for greater flexibility when modeling complicated conductivity structures. High accuracy is obtained with less effort compared to graded tensor product grids used with finite differences. It also helps in the construction of an analytic boundary condition, avoiding a few drawbacks the implementation by Oristaglio and Hohmann (1984) has. Contrary to the finite difference approach, the matrices resulting from the discretization are symmetric and, thus, allow for a wide range of efficient and state-of-the-art time integration techniques, which can exploit this property. One such family of time integrators is based on building a Krylov

subspace and extracting approximations to the matrix exponential function from said space, thus evaluating the sought electric field directly at a given time.

3 Theory

3.1 Governing Equations

Our governing equation derives from Maxwell's equations in the diffusive limit, with the constitutive relations used and the magnetic field already eliminated. Thus, we write

$$\nabla \times (\mu^{-1} \nabla \times \mathbf{e}) + \partial_t \sigma \mathbf{e} = -\partial_t \mathbf{j}^e \quad (1)$$

with

$\mathbf{e} = \mathbf{e}(\mathbf{x}, t)$	the electric field,
$\mu = \mu(\mathbf{x})$	the magnetic permeability,
$\sigma = \sigma(\mathbf{x})$	the electric conductivity, and
$\mathbf{j}^e = \mathbf{j}^e(\mathbf{x}, t)$	the impressed source current density.

We now restrict ourselves to the two-dimensional case (xz -plane) with infinitely long line sources perpendicular to this plane. Given these assumptions, we can express the electric field as

$$\mathbf{e}(x, y, z, t) = e(x, z, t) \mathbf{y} \quad (2)$$

where \mathbf{y} is the unit vector along the y -axis and e a scalar function. Equation (1) then reduces to

$$-\nabla^2 e + \sigma \mu \partial_t e = -\mu \partial_t j^e. \quad (3)$$

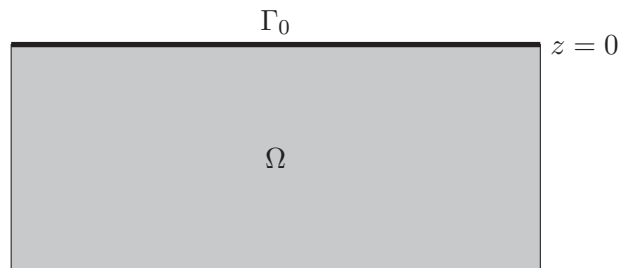


Figure 1: Computational domain with an arbitrary conductivity structure. On top is the air-earth interface (bold), the remaining boundaries are subsurface boundaries.

Our computational domain Ω (cf. Figure 1) is a rectangle and its top edge is aligned with the air-earth interface $\Gamma_0 = \{(x, z) : z = 0\}$ on which we impose an explicit boundary condition and perfect conductor boundary conditions on all other domain boundaries. It is important for these boundaries to be sufficiently far away from the sources, so that they don't distort the propagating field.

Our objective will be to compute the configuration of the electric field e_i at times t_i for $i \in \{1, 2, \dots, n\}$ given an initial field e_0 at t_0 . We use sources that are switched off at $t = 0$ and therefore the right hand side of (3) vanishes for $t > 0$.

3.2 Air-Earth Interface

We assume that e satisfies Laplace's equation in the isolating air half-space ($z < 0$)

$$-\nabla^2 e = 0. \quad (4)$$

Applying the Fourier transform in x to this equation yields

$$-\partial_{zz}\hat{e} + k^2\hat{e} = 0, \quad (5)$$

where \hat{e} is the Fourier transform of e . After a few transformations (cf. Goldman et al. 1986) one can derive the exact boundary condition at $z = 0^+$

$$\partial_z e = T e + \mu \partial_t j^e \quad (6)$$

with the linear operator T being a linear convolution operator in x . This approach is also extensible to three dimensions and is described in detail in Goldman et al. (1989).

3.3 Spatial Discretization

Our next step towards a computable model is the discretization of the spatial domain. We focus on the finite element method, but also briefly mention finite differences because those are also used in the numerical examples.

Finite Elements To be able to apply the finite element method we need a variational formulation of our problem in (3), which is given by

$$\partial_t a(e, \psi)_\sigma + c(e, \psi) + \int_{-\infty}^{+\infty} \partial_z e(x, z = 0^+, t) \psi(x, 0) dx = 0 \quad (7)$$

with the bilinear forms a and c defined as

$$a(u, v)_\sigma = \mu \int_{\mathbb{R}_+^2} \sigma(\mathbf{x}) u(\mathbf{x}) v(\mathbf{x}) d\mathbf{x}, \quad (8)$$

$$c(u, v) = \int_{\mathbb{R}_+^2} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} \quad (9)$$

where $\mathbb{R}_+^2 = \{(x, z) : z > 0\}$ denotes the lower half-space.

By using the exact boundary condition (6) we can write

$$\partial_t a(e, \psi)_\sigma + c(e, \psi) + \int_{-\infty}^{+\infty} T e(x, t) \psi(x, 0) dx = -\mu \int_{-\infty}^{+\infty} \partial_t j^e(x, t) \psi(x, 0) dx. \quad (10)$$

To be able to discretize this equation we need a computable expression for the following bilinear form

$$b(\phi, \psi) = \int_{-\infty}^{+\infty} T \phi(x) \psi(x) dx \quad (11)$$

and indeed, after some rearrangement in the Fourier domain, one obtains (Goldman et al. 1986)

$$b(\phi, \psi) = -\frac{1}{\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \log(|x - y|) \phi'(x) \psi'(y) dx dy. \quad (12)$$

The discretization of this integral on the boundary of a triangulation can now be easily computed.

The continuity of e allows us to use standard linear Lagrange elements on a triangulation of the computational domain. After discretizing the variational formulation we arrive at

$$M \partial_t \mathbf{e} = K \mathbf{e}. \quad (13)$$

Finite Differences For comparison, we also perform a finite difference discretization and get the ODE $\partial_t \mathbf{e} = A\mathbf{e}$ where A is a large sparse matrix representing the discrete curl-curl operator.

The following procedure, that has to be repeated in every time step, outlines the implementation of an exact boundary condition in this setting: Take the field data at the air-earth interface, interpolate it to an equidistant grid, perform an FFT to transform the data into the wave number domain, multiply every wave number with a constant (this would be a convolution in the space domain), revert the FFT and finally interpolate back to the graded grid to get the field values at a negative z , thus, inside the air half-space. Use the data resulting from this *classical upward continuation* in the usual finite difference stencil to compute the top layer in the next step.

A variation of this approach was also implemented for the numerical examples. It combines all of the above steps into a single (dense) matrix that has the same effect but can be evaluated more efficiently. This is called *matrix upward continuation* later on.

Details about the finite difference discretization can be found in Oristaglio and Hohmann (1984).

3.4 Time Integration Techniques

Krylov Subspace Methods Krylov subspace methods cover a big range of applications and are in use for several decades. What we want to focus on are Krylov subspace methods for the evaluation of matrix functions, like we can see in (13), where the function is the matrix exponential and the matrices come from the spatial discretization using finite elements. A nice side effect of the origin of those matrices is, that they are usually symmetric which many algorithms can benefit from.

Given a square matrix A of size $n \times n$, a vector \mathbf{b} of length n and a suitable scalar function f , we can write

$$f(A)\mathbf{b} = p(A)\mathbf{b}. \quad (14)$$

with p a polynomial that Hermite-interpolates in the eigenvalues of A . We will be focusing on the exponential function and on rational Krylov subspaces, that are defined as follows

$$\mathcal{Q}(A, \mathbf{b}) := q_{m-1}(A)^{-1} \mathcal{K}_m(A, \mathbf{b}) \quad (15)$$

with $\mathcal{K}_m(A, \mathbf{b}) = \{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \dots, A^{m-1}\mathbf{b}\}$ and

$$q_{m-1}(z) = \prod_{\substack{j=1 \\ \xi_j \neq \infty}}^{m-1} (z - \xi_j). \quad (16)$$

Such subspaces are constructed, e. g., with the rational Arnoldi method to obtain an orthonormal basis of \mathcal{Q}_m from which approximations to $f(A)\mathbf{b}$ can be computed with several procedures. We have to choose the poles ξ_j of the rational Krylov subspace. Their number and choice highly impact the quality of the approximations that can be extracted from the subspace. Luckily, there is a heuristic to determine good poles for the approximation of the exponential function, given a certain parameter interval (in our case the times we want to advance to) and accuracy requirements. These, and many more things concerning rational Krylov subspaces, are tackled in Güttel (2010), which is also a good starting place to dig deeper into the literature.

Looking more closely at (13) we see, that in order to apply a Krylov subspace method we need to remove the matrix M in front of the time derivative, e. g. by moving it to the right

$$\partial_t \mathbf{e} = M^{-1} K \mathbf{e}. \quad (17)$$

The inverse of the matrix doesn't have to be explicitly computed, but we have to solve a system of linear equations in every iteration step. This might be considered too expensive, so there is a technique

termed *mass lumping* which converts M into a diagonal matrix that can be easily inverted without distorting the solution too much.

However, even if we don't have to solve linear systems with the matrix M , we do have to solve shifted systems with the matrix K during the construction of the rational Krylov subspace. Therefore, an efficient solver is also an important ingredient. We will see in the numerical examples what options we have.

Time Stepping On the other hand there are simple, yet relatively efficient explicit time stepping techniques. One of them is the well-known Du Fort-Frankel method which is implemented here to perform the integration in time for the finite difference discretization.

4 Numerical Examples

All computations were carried out on a machine with four AMD Opteron 8380 quad cores, running at 2.5 GHz with a total of 128 GiB of RAM. The algorithms were implemented in pure MATLAB code running under MATLAB 2008b, with some finite element related tasks (nothing time-critical) performed by COMSOL 3.5a, both running in a 64-bit Linux environment. To generate reproducible timing we restricted ourselves to one computational thread that was explicitly pinned to one of the cores.

4.1 Model

The considered models are basically those from Oristaglio and Hohmann (1984), all based on graded tensor product grids. We extended the mesh slightly so as not to get a distortion from the boundaries for late times. The resulting grid has 236×88 cells, is 12800 m wide and 5255 m deep, with grid spacings ranging between 10 m and 240 m. To make the computations comparable with the finite difference code we decided to use exactly the same grid for the finite element code. We simply subdivided each cell into two triangles, however, the number of unknowns or degrees of freedom and their locations are identical. Technically, this is of course not necessary and a properly adapted unstructured grid would certainly yield even better results at lower cost.

The initial field e_0 at $t_0 = 10^{-6}$ s is due to two line sources at positions $(x, z) = (-500 \text{ m}, 0 \text{ m})$ and $(x, z) = (0 \text{ m}, 0 \text{ m})$, the former being negative while the latter is positive with a unit source current strength. It is computed analytically for a homogeneous half-space that corresponds to the background conductivity of the model.

We have looked at two models with different conductivity structures. See Figure 2 to get an approximate idea of their location inside the grid. The background has a resistivity of $300 \Omega\text{m}$ and the conductor (red) $0.3 \Omega\text{m}$.

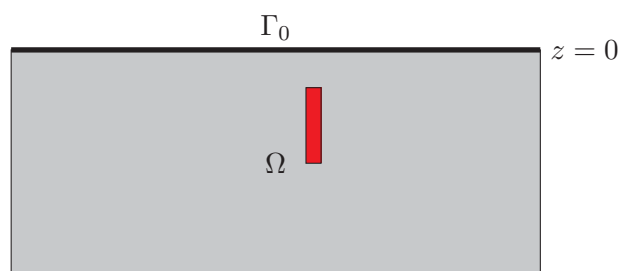


Figure 2: Schematic view of the computed models. The models are the homogeneous half-space and a conductor (red) inside a homogeneous half-space. The conductor is located at the position $290 \text{ m} \leq x \leq 310 \text{ m}$, $100 \text{ m} \leq z \leq 400 \text{ m}$ in the grid.

4.2 Methods and Setup

Independently of the method of choice we start at the time $t_0 = 10^{-6}$ s and integrate from there till we reach $t_n = 10^{-2}$ s. We seek to compute the configuration of the electric field at 10 logarithmically equidistant times per decade.

For the finite difference code we mostly stick to the implementation by Oristaglio and Hohmann (1984), with the only option to perform the upward continuation in the traditional way, which is a significant expense per time step, or we can decide to precompute the matrix formulation of this upward continuation and to apply it in every time step.

For the finite element method we have no choice regarding the exact boundary condition. However, we can choose between different variants of the rational Krylov method. For all of them we have a total of 25 poles, which should be close to optimal for this application. For the first 24 poles we alternate between $2.7826 \cdot 10^5$ and $0.0242 \cdot 10^5$. The last pole is set to infinity.

We have four implementations, that we included in this test. Two of them use mass lumping to reduce the mass matrix to a diagonal matrix. Furthermore, for each group we can either call the standard MATLAB solver or we can exploit the fact that we have multiple identical poles for which we have to solve with different right hand sides, but identical matrices. We do this by computing a sparse LU factorization (we use UMFPACK for this) for every unique pole and then leverage that factorization to speed up the solves during the construction of the rational Krylov subspace.

4.3 Results

We first look at the computation times that are pictured in Figure 3 and listed in Table 1. The times shown here are for the homogeneous half-space model, but aren't significantly different for the other models since the grid and the minimal conductivity are identical. Thus, we can use these numbers to judge the acceleration we can get by using our method.

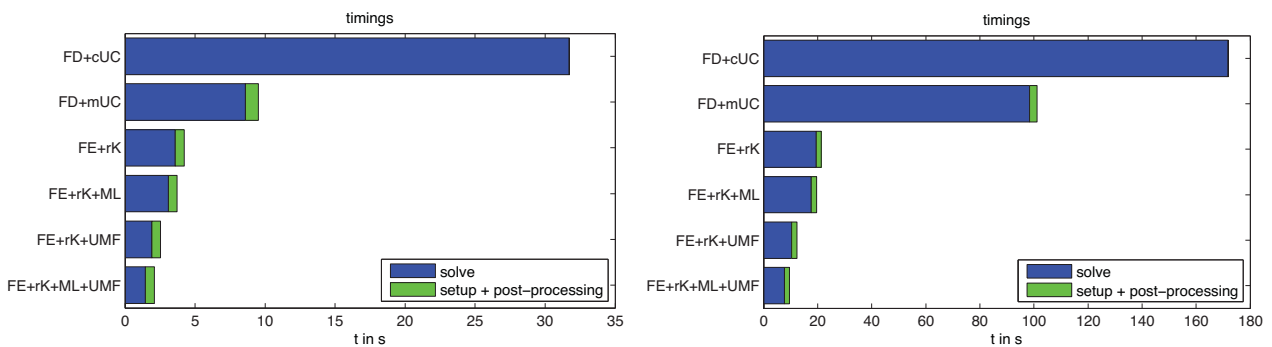


Figure 3: Computation times split into solve and setup/post-processing step, cf. also Table 1. On the *left* are the computation times for the default grid and on the *right* for the regularly refined version of this grid with approximately four times as many nodes and degrees of freedom. The following abbreviations were used: classical upward continuation (cUC), matrix upward continuation (mUC), rational Krylov (rK), mass lumping (ML), UMFPACK solver (UMF).

As is clearly visible, all FE-based methods are significantly faster than their finite difference counterparts, sometimes even 22 times faster.

Looking at the transients (cf. Figure 4) at some select points inside the mesh, we see that although these newer methods are so much faster, we don't really sacrifice accuracy. In fact, in many cases the finite element solution is more accurate than the one obtained by finite differences.

We conclude our numerical examples with showing some cross-sections (cf. Figure 5), that are nothing but a few snapshots of an animation that is just not suitable for this medium, but nevertheless gives a coarse idea of how the electric field propagates with time.

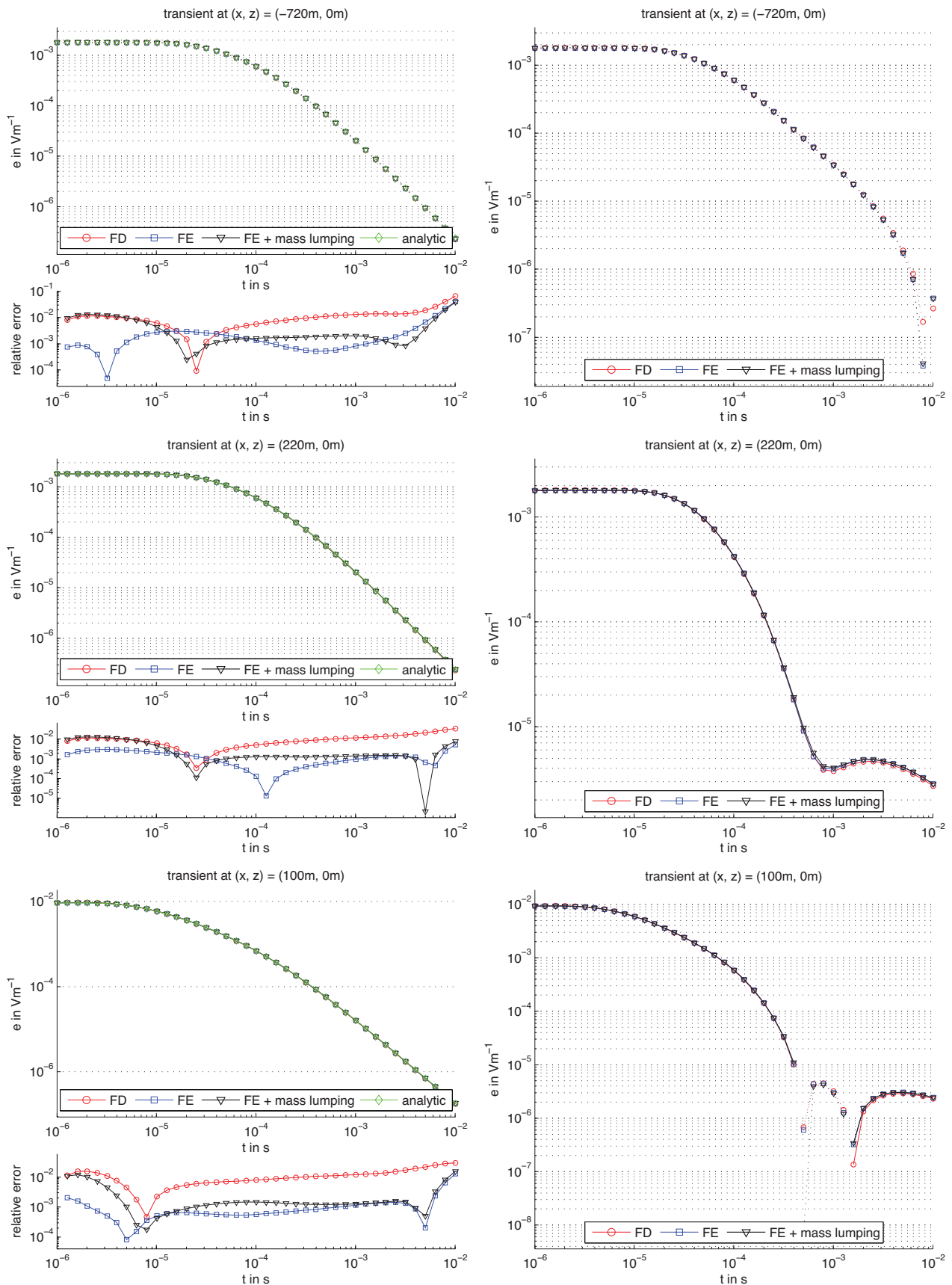


Figure 4: Transients of the homogeneous half-space model with relative errors (left) and the model with a conductor (right). Negative field values are denoted by dashed lines while positive values are drawn with a solid line. We can see that we have a very good agreement between the FD and FE solution. When comparing against an analytical solution in the homogeneous half-space, we see that the relative errors are often lower than those of the FD solution, despite the lower numerical effort.

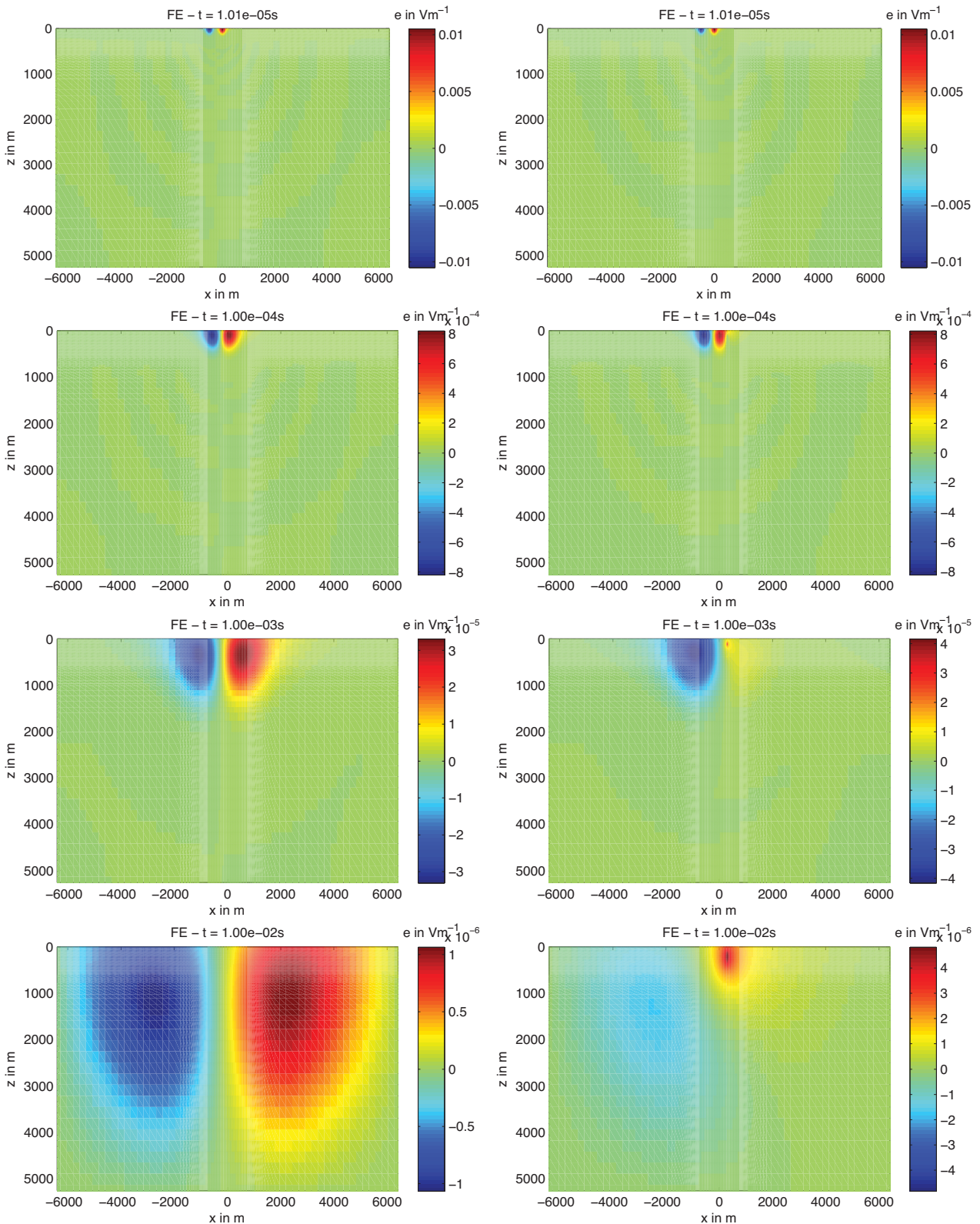


Figure 5: Cross-sections of a homogeneous half-space model (left) and the model with a conductor (right). These cross-sections were plotted to give a visual idea of how the fields advance in time. The common initial configuration was omitted. We decided to plot the results of the FE simulation, but there was hardly any visible difference compared to the FD solution. On the right we can nicely see how the field gets *captured* inside the high conductivity structure and creates an anomaly compared to the homogeneous case.

Method	Default grid		Refined grid	
	Solve	Other	Solve	Other
FD, time stepping, classical upward continuation	31.7 s	< 0.1 s	171.8 s	< 0.1 s
FD, time stepping, matrix upward continuation	8.6 s	0.9 s	98.4 s	2.7 s
FE, rational Krylov	3.6 s	0.6 s	19.4 s	1.9 s
FE, rational Krylov, mass lumping	3.1 s	0.6 s	17.7 s	1.9 s
FE, rational Krylov, UMFPACK	1.9 s	0.6 s	10.4 s	1.9 s
FE, rational Krylov, mass lumping, UMFPACK	1.4 s	0.6 s	7.7 s	1.9 s

Table 1: Computation times for the default grid and one, that was obtained from this with regular refinement. The column *Solve* denotes the time spent for performing the integration in time, while the column *Other* lists the time spent in setting up the boundary condition and the computation of field values from the solution vectors in case of the FE method. Other setup times are not shown. The advantage of FE-based methods can be clearly seen.

5 Conclusions

We were able to leverage several state-of-the-art techniques to create a combined efficient forward modeling code that is significantly faster than traditional codes. We have also seen that we don't have to make sacrifices regarding the accuracy of those computations. This was already achieved by using the mesh from the finite difference discretization, which was helpful for comparison, but which also has many limitations due to its regular structure. An unstructured grid—properly adapted to the problem—could be used to further increase accuracy or speed.

We are working on creating a similar framework for the three-dimensional time domain TEM problem. All of the methods and tools are already existing or have a straightforward extension to 3D. Given the results from 2D we expect an even greater speed-up when applying this to three dimensions.

6 Acknowledgments

This work was supported by the German Research Foundation (DFG) under signature Spi 356/9.

References

- Goldman, Y., C. Hubans, S. Nicoletis, and S. Spitz (1986). *A finite-element solution for the transient electromagnetic response of an arbitrary two-dimensional resistivity distribution*. *Geophysics* 51(7): 1450–1461.
- Goldman, Y., P. Joly, and M. Kern (1989). *The Electric Field in the Conductive Half Space as a Model in Mining and Petroleum Prospecting*. *Mathematical Methods in the Applied Sciences* 11: 373–401.
- Güttel, S. (2010). *Rational Krylov Methods for Operator Functions*. PhD thesis. TU Bergakademie Freiberg.
- Oristaglio, M. L. and G. W. Hohmann (1984). *Diffusion of Electromagnetic Fields into a Two-Dimensional Earth: A Finite-Difference Approach*. *Geophysics* 49(7): 870–894.
- Wang, T. and G. W. Hohmann (1993). *A Finite-Difference, Time-Domain Solution for Three-Dimensional Electromagnetic Modeling*. *Geophysics* 58(6): 797–809.
- Yee, K. S. (1966). *Numerical Solution of Initial Boundary Problems Involving Maxwell's Equations in Isotropic Media*. *IEEE Trans. Ant. Propag.* 14: 302–309.