Fast time-domain electromagnetic modelling in the frequency domain

Dieter Werthmüller[*], **Wim A. Mulder**[*†], **and Evert C. Slob**[*]

[*]*TU Delft, Building 23, Stevinweg 1 / PO-box 5048, 2628 CN Delft;*

[†]*Shell Global Solutions International BV, Grasweg 31, 1031 HW Amsterdam;*

E-mail: Dieter@Werthmuller.org;

Keywords: *Fourier transforms, CSEM, 3D modelling.*

(6th July 2020)

Running head: **Time-domain CSEM**

ABSTRACT

Modelling three-dimensional time-domain controlled-source electromagnetic data with frequency-domain solvers is an alternative to time-domain solvers. It requires the computation of many frequencies if standard Fourier transforms are used, which can make it prohibitively expensive in comparison with a time-domain computation. The speed of time-domain modelling with frequency-domain computation is defined by three key points: solver, method and implementation of the Fourier transformation, and gridding. The faster the frequency-domain solver, the faster time-domain modelling will be. It is important that the solver is robust over a wide range of frequencies. A three-dimensional, iterative, frequency-domain multigrid solver satisfies this requirement, but any other robust solver will do. The method should require as few frequencies as possible while remaining robust. As the frequency range spans many orders of magnitude, the required frequencies are ideally equally spaced on a

1

logarithmic scale. The fast frequency-to-time domain method uses either the digital linear filter method or the logarithmic fast Fourier transform together with a careful selection of frequencies and interpolation. This methodology requires typically 15 to 20 frequencies to cover a wide range of offsets in one-dimensional test models. The gridding should be frequency-dependent, which is accomplished by making it a function of skin depth. Optimising for the least number of required cells for a given frequency should be combined with optimising for computational speed. Looking carefully at these three points results in much smaller computation times with speedup factors of ten or more over previous results. A frequency-domain computation of time-domain data can therefore be competitive to time-domain solvers if the required frequencies and the corresponding grids are carefully chosen.

INTRODUCTION

The controlled-source electromagnetic (CSEM) method is one of the common non-seismic tools in exploration geophysics, not only in hydrocarbon exploration (Constable, 2010), but also in the search for sulfides (Gehrmann et al., 2019), water (Pedersen et al., 2005), geothermal sources (Girard et al., 2015), or geological purposes (Johansen et al., 2019). While current sources with a few frequencies are used in the deep marine environment, transient measurements are more common in the shallow marine environment and on land (e.g., Ziolkowski et al., 2007; Andréis and MacGregor, 2007; Avdeeva et al., 2007). One of the main reasons is the dominance of the airwave in shallow marine and terrestrial measurements, which can be better separated in the time domain. CSEM is usually divided into frequency- and time-domain methods, depending whether the source signal is a continuous waveform, such as a sine, or a finite waveform, such as a pseudo-random binary sequence (PRBS). A numerical comparison of the two methods is given by Connell and Key (2013). Acquired CSEM data are subsequently often analysed (modelled and inverted for) in their respective domain, either the frequency or time domain. Modellers of layered media usually exploit the horizontal shift-invariance by computing the responses in the wavenumber-frequency domain followed by a 2D inverse spatial Fourier transform, also called Hankel transform, to the space-frequency domain, and a regular inverse Fourier transform if time-domain data are required (e.g., Hunziker et al., 2015).

CSEM codes for 2D and 3D computations, on the other hand, mostly compute their responses directly in the required domain, either frequency or time. There is a wealth of 3D electromagnetic codes, with a first boom in the 1990's, with a good state of the art provided by Oristaglio and Spies (1999). A second wave occurred during the CSEM boom in the

3

hydrocarbon industry in the early 21th century, see, e.g., Avdeev (2005) and Börner (2010) for comprehensive overviews. Recent publications span the width of numerical methods from integral equation codes (Kruglyakov and Bloshanskaya, 2017) and differential equation methods such as finite differences (Sommer et al., 2013), finite elements (Grayver et al., 2013), and finite volumes (Jahandari and Farquharson, 2014). Besides the different methods there are also many different types of discretisation. Commonly used, besides rectilinear Yee grids (Yee, 1966) as used by our code, are Lebedev grids (Lebedev, 1964) but also unstructured grids using, for instance, tetrahedral (Cai et al., 2017). The computational requirement to model CSEM data also depends heavily on the technique which is used to solve the system, where there are direct solvers (Grayver and Kolev, 2015), indirect solvers (Jaysaval et al., 2015), and more recently also hybrid solvers (Liu et al., 2018).

Under certain conditions it can be competitive to model time-domain data with a frequency-domain code, as shown by Mulder et al. (2008). The conditions are: a sufficiently powerful solver, appropriate frequency selection and interpolation, and an automated gridding, for which they used the multi-frequency CSEM approach presented by Plessix et al. (2007). We build upon these results but improve the run time from hours to minutes. The main reasons for this significant speed-up are an adaptive, frequency-dependent gridding scheme that minimizes the required cells in each dimension, and a logarithmic Fourier transform such as digital linear filters (DLF, Ghosh, 1971) or the logarithmic fast Fourier transform (FFTLog, Hamilton, 2000) to go from the frequency to the time domain. The latter makes it also possible to only use the imaginary part of the frequency-domain response, which has advantages when it comes to interpolation.

In the next section, we briefly review the methodology as introduced by Plessix et al. (2007) and Mulder et al. (2008) and highlight their advantages and shortcomings. This is

followed by an outline of our changes to the Fourier transform and the adaptive gridding. Finally, we demonstrate the efficiency of the approach with some numerical results.

MOTIVATION

Being able to model CSEM data both in the frequency-domain and in the time-domain can be desirable, as both domains have advantages and disadvantages. One way to achieve this is to implement Maxwell's equation in both domains, as it is done, for instance, in `SimPEG` (Cockett et al., 2015). Another approach is to have Maxwell's equation only implemented in one domain, and use Fourier transforms to go to the other. However, this approach can be costly, as many frequencies over a wide range are required to go from the time domain to the frequency domain, or many times over a wide range for the opposite direction. We present a methodology which significantly reduces the amount and range of the required frequencies, translating into a significant reduction in computation time. The required computation grids for low frequencies (in our case in the order of $0.001\,\mathrm{Hz}$) and high frequencies (around $100\,\mathrm{Hz}$) are hugely different. Low frequencies can be computed on a coarser grid, but they require a much larger domain in order to avoid boundary effects. High frequencies, on the other hand, require denser gridding, but they are much more limited in reach. An adaptive gridding scheme is therefore required. We build our approach upon Plessix et al. (2007), who presented such an adaptive gridding for multi-frequency (and multi-source) CSEM modelling.

An adaptive grid in the frequency domain is naturally based on the skin depth, which is the distance after which the amplitude of the electromagnetic field is decayed by $1/e \approx 37\,\%$. The skin depth δ is a function of conductivity and frequency, and for the diffusive approximation of Maxwell's equation in an isotropic, homogeneous medium is given by (e.g.,

Ward and Hohmann, 1988, equation 1.53)

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}} \approx 503.3/\sqrt{f\sigma}, \tag{1}$$

where σ is conductivity (S/m), $\omega = 2\pi f$ is angular frequency of frequency f (Hz), and μ is magnetic permeability (H/m). The approximation is obtained by using the free-space value of magnetic permeability, $\mu_0 = 4\pi \times 10^{-7}$ H/m.

Plessix et al. (2007) define the minimum cell width $\Delta_{\min}$ as a fraction of the minimum skin depth $\delta_{\min}$, where $\delta_{\min}$ should be 2–3 times bigger than $\Delta_{\min}$. The cells have to be smallest around the source; in the marine case, the minimum skin depth is therefore defined by the conductivity of seawater. However, this yields quite large cells for low frequencies, so special care has to be taken around the source by defining a maximum allowed $\Delta_{\min}$. The grid dimension, on the other hand, is defined as a function of the skin depth for the average conductivity of the background, δ_{ave}. They use four times δ_{ave} for the x-, y-, and downward z-directions, and a fixed 50 km for the upward z-direction to account for the airwave. To reduce the number of cells, it is desirable to introduce stretching, at least in the boundary zone outside of the area where source and receivers are located.

Mulder et al. (2008) provide a computational complexity analysis of various methods to model transient electromagnetic responses directly in the time-domain, and compare it to the computation of transient EM responses in the frequency-domain with a subsequent Fourier transform. They conclude the review by stating *«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.»* Their approach is to minimize the computation time by having, in addition to the just introduced adaptive gridding, an adaptive frequency selection scheme. This scheme starts with computing the responses for a set of just five fre-

quencies, regularly sampled on a log-scale, from minimum to maximum required frequency. All the other frequencies are interpolated with a shape-preserving piecewise-cubic Hermite interpolation (PCHIP, Fritsch and Carlson, 1980). Testing the stability of the obtained response by removing a single frequency-value at a time their scheme decides if more frequencies in-between the already computed ones are required. In this way frequencies are only added if required, hence if certain criteria of response stability are not met. While this method is good and effective for a single offset, it loses all its advantages if one tries to compute different offsets within one computation, as each offset requires a different set of adaptive frequencies. Additionally, it hampers the parallelization over frequencies.

We present improvements to both the adaptive gridding and the transform from frequency domain to time domain, which makes time-domain modelling with a frequency-domain code even more competitive in comparison with time-domain codes. It is important to note that while we show the transform from frequency to time domain, the conclusions regarding the Fourier transform can equally well be applied to the transform from time to frequency domain data for a code that solves Maxwell's equations in the time domain. A variation of our method was implemented, for instance, by `custEM` (Rochlitz et al., 2019) to obtain frequency-domain data from time-domain computations (Rochlitz, private communication, 2020).

METHODOLOGY

For the numerical computations we use the open-source (Apache License 2.0) code `emg3d` (Werthmüller et al., 2019b), a multigrid solver based on Mulder (2006), which can be used as a preconditioner for Krylov subspace solvers or as a solver on its own. The multigrid approach works fine for the diffusive approximation of Maxwell's equation, which assumes

7

that $\omega\varepsilon \ll \sigma$, where ε is electric permittivity (F/m). The remaining system to solve in the frequency domain is then given by the second-order differential equation of the electric field,

$$i\omega\sigma\mathbf{E} + \nabla \times \mu^{-1}\nabla \times \mathbf{E} = -i\omega\mathbf{J}_{\mathrm{s}},\tag{2}$$

where $\mathbf{E}$ is the electric field (V/m) and $\mathbf{J}_{\mathrm{s}}$ the current source (A/m^2); time dependence is $\exp(i\omega t)$. The standard multigrid approach fails for severe stretching or strong anisotropy, for which `emg3d` has implemented known improvements such as line-relaxation and semicoarsening (Jönsthövel et al., 2006), with a non-standard Cholesky decomposition to speed up the computation (Mulder et al., 2008). One of the big advantages of the multigrid method is that it scales linearly (optimal) with the grid size in both CPU and RAM usage (Mulder, 2020). This makes it feasible to run even big models on standard computers, without the need for big clusters. All the examples in this article are run on a laptop with an i7-6600U CPU @ 2.6 GHz (x4) and 16 GB of memory, using Ubuntu 18.04 and Python 3.7. For the semi-analytical computations of layered models we use the open-source (Apache License 2.0) code `empymod` (Werthmüller, 2017).

FREQUENCY SELECTION

An important factor in terms of speed and accuracy for time-domain responses obtained from frequency-domain computations is the selection of the required frequencies. The fewer frequencies required, the quicker we obtain the time-domain result. We decided to use a regular spacing of frequencies on a log-scale, rather than an adaptive scheme. This approach is favourable if a wide range of offsets is needed, as the required frequencies change with offset and an adaptive frequency selection is therefore often tailored to a single offset. Also, it allows for straightforward parallelization over frequencies, which is not completely possible

with an adaptive scheme.

For the actual transform we use either the digital linear filter (DLF) method or the logarithmic fast Fourier transform (FFTLog). The DLF method was introduced to geophysics by Ghosh (1971), and is arguably the most common method in EM geophysics for both its simplicity and its speed. It is implemented for the Hankel and Fourier transforms in most EM modelling codes, e.g., Key (2009). A simple tool to design digital linear filters was recently presented by Werthmüller et al. (2019a), together with a comprehensive overview of the history and development of DLF in geophysics. FFTLog, introduced by Hamilton (2000), is another transform algorithm which proved to be powerful for the frequency-to-time domain transformation of EM responses, e.g., Werthmüller et al. (2014). In our tests they are both about equal in speed and accuracy. DLF requires a wider range and many more frequencies than the FFTLog. Both methods share some important characteristics in comparison with the standard FFT: the required input frequencies are equally spaced on a logarithmic scale, and they only require either the real or the imaginary part of the frequency-domain response. We can take advantage of that by using only the imaginary part of the frequency-domain response. The imaginary part goes to zero when the frequency goes to zero or to infinity, with the advantage that knowing the endpoints makes it possible to convert the extrapolation of missing frequencies into interpolation.

There are three key-parameters that have to be defined: Minimum and maximum frequency ($f_{\min}, f_{\max}$) and the number of frequencies per decade (n_{dec}^f). Trial-and-error with a 3D code is very time-intensive. However, a simplified, layered model for the required survey setup and a fast 1D modeller makes it possible to estimate these parameters easily. To this end, we created a graphical, interactive tool using `empymod`, as shown in Figure 1. The example model is a marine scenario with $1\,\text{km}$ water depth of resistivity $\rho = 0.3\,\Omega\,\text{m}$

$(\rho = \sigma^{-1})$ and a $100\,\mathrm{m}$ thick target of $100\,\Omega\,\mathrm{m}$ at $1\,\mathrm{km}$ below the seafloor in a background of $1\,\Omega\,\mathrm{m}$. The source is $50\,\mathrm{m}$ above the seafloor, the receiver is on the seafloor, and the response is the inline x-directed E-field. The left subplot shows the imaginary part in the frequency domain and the right plot the corresponding impulse response in the time domain. The red lines are the precise results obtained from `empymod`. The blue circles indicate the actually computed responses and the black dots the frequencies which are interpolated or set to zero. The resulting time-domain response has a relative error of less than $1\,\%$ everywhere except for very early times.

Figure 2a shows exactly the same on a logarithmic scale, without the interactive GUI-related elements. It can be seen that with the chosen frequencies the time-domain response starts to divert above about $100\,\mathrm{s}$ and below $0.3\,\mathrm{s}$. It also shows the oscillating high-frequency part, which is hard to interpolate. Figure 2b is the same as 1, but transformed with the DLF method applying the 81-point sine-cosine filter from Key (2009). The same frequencies were computed as in the FFTLog case and the missing ones interpolated. The error of the corresponding time-domain response is comparable, so either FFTLog or DLF can be used. In the above example we used 20 frequencies, but that many would not be required for the shown offset of $5\,\mathrm{km}$, a few of the lower and higher frequencies could be left out. However, with this frequency-selection we can model a wide range of offsets. This is shown in Figure 3, where the above parameters were used to model the responses at offsets $r = 1.5, 3, 6$, and $12\,\mathrm{km}$.

The shortest offset defines the highest required frequency, and the largest offset the lowest required frequency. So the more one can restrict the necessary offset range, the less frequencies are needed. Another important factor is how to interpolate and extrapolate from the computed frequencies to the frequencies required for the Fourier transform. For the

FFTLog only extrapolation for higher and lower frequencies is required. The EM response becomes highly oscillatory for high frequencies, which makes it very hard to extrapolate the response to frequencies $f > f_{\max}$. However, if $f_{\max}$ is chosen judiciously, the importance of higher frequencies for the Fourier transform can be neglected and we can set those responses to zero. The extrapolation of frequencies $f < f_{\min}$ can be changed to an interpolation by assuming a zero imaginary response at zero frequency, and we then use PCHIP to interpolate the missing frequencies (as we work on a logarithmic scale we cannot choose $0\,\mathrm{Hz}$, but instead take $10^{-100}\,\mathrm{Hz}$). In the case of the DLF method we also have to interpolate in between the computed frequencies, for which we found it better to use a cubic spline. As can be seen from Figures 1 and 2b, using the FFTLog or the DLF with the same actually computed frequencies results in very similar responses.

The changes to the Fourier transform in comparison with Mulder et al. (2008) can be summarized in three points: (1) regular log-scale spacing for the frequency selection; (2) DLF or FFTLog instead of FFT; and (3) using only the imaginary part of the frequency-domain response. The actual speed of the transform is unimportant, as the computation of the frequency-domain responses takes much longer than the transform itself. What matters is solely how many frequencies are required by it in order to be stable, and how long it takes to compute the responses for these frequencies.

GRIDDING

The computation grid consists of a core or survey domain D_{s} which should contain all source and receiver positions. The survey domain usually has no or very little cell stretching α_{s}.

The minimum cell width is defined as

$$\Delta_{\min} = \delta(f, \sigma_{\mathrm{src}})/n_\delta \, , \tag{3}$$

where σ_{src} is the conductivity of the media in which the source resides, and n_δ is a positive number that defines how many cells there should be per skin depth. The actual computation domain D_{c} is usually much bigger than D_{s} in order to avoid artefacts from the perfectly electrically conducting boundary condition. It can also have a much higher stretching α_{c}. In our scheme we have chosen D_{c} such that the distance for the signal diffusing from the source to the boundary and back to the receiver closest to the boundary is at least two wavelengths, after which the initial signal is reduced to a millionth of its strength. The wavelength to compute D_{c} is given by

$$2\lambda = 4\pi\delta(f, \sigma_{\mathrm{ave}}) \approx 6324.7/\sqrt{f\sigma_{\mathrm{ave}}} \, , \tag{4}$$

where σ_{ave} is the average conductivity, which can vary for different directions. However, the skin-depth approach fails for air, in which the EM field propagates as a wave at the speed of light. A largest computational domain is therefore enforced, defining the maximum distance from the source to the boundary; this distance is by default $100\,\mathrm{km}$, but this can be reduced in the marine case with increasing water depth. Note that for shallow marine and land cases this also applies to the horizontal dimensions, not only to the upward z-direction and similarly for very resistive basements, even in deep water. One way to circumvent this difficulty is the use of a primary-secondary formulation, where the primary field, including the air wave, is computed with a semi-analytical code for layered media. We do not consider this approach here.

In summary, the adaptive gridding takes f, D_{s}, σ_{src}, σ_{ave}, n_δ, and ranges for α_{s}, α_{c}, where we usually fix $\alpha_{\mathrm{s}} = 1$ or keep it at least below 1.05, and let α_{c} be anything between

$[1, 1.5]$. The minimum cell width $\Delta_{\min}$ can further be restricted by a user-defined range. Given these inputs the adaptive gridding will search for the smallest possible number of cells which fulfils these criteria. The multigrid method implemented in `emg3d` puts some constraints on the number of cells, of which the adaptive gridding takes care (the number of cells have to be powers of two multiplied by a low prime, e.g., $\{2, 3, 5\} \cdot 2^n$).

The main difference with Mulder et al. (2008) is that their adaptive gridding searches for the optimal stretching factor α fulfilling certain criteria, for a fixed number of cells. Our adaptive gridding, on the other hand, searches for the smallest number of cells that still fulfil the given criteria. The number of cells becomes therefore also a function of frequency. To go from the model grid to the computational grid, we use the volume-averaging technique on logarithmic resistivities, as used in Plessix et al. (2007). While this technique ensures that the total resistivity in the subsurface remains the same, it does not consider effective-medium theory (Davydycheva et al., 2003), for instance, the apparent anisotropy from a stack of finely layered formations of varying resistivity.

NUMERICAL EXAMPLES

Homogeneous space

The first example is the inline electric field from a source at the origin measured by an inline receiver with an offset of $900\,\mathrm{m}$ in a homogeneous space of $1\,\Omega\,\mathrm{m}$. We chose this simple example to compare it with the analytical solution and with previously published results. We used the following values to define the required frequencies: $f_{\min} = 0.05\,\mathrm{Hz}$, $f_{\max} = 21\,\mathrm{Hz}$, using FFTLog with 5 frequencies per decade. This results in 14 frequencies to compute from $0.05\,\mathrm{Hz}$ to $20.0\,\mathrm{Hz}$. The complete frequency range for the transform includes

30 frequencies from $0.0002\,\text{Hz}$ to $126.4\,\text{Hz}$. For the adaptive gridding the following inputs were used: $n_\delta = 12$, minimum cell width must be between 20 and $40\,\text{m}$, and $\alpha_\text{s} = 1$, $\alpha_\text{c} = [1, 1.3]$. This created grids with cell numbers between $46\,080$ ($80 \times 24 \times 24$, for $20.0\,\text{Hz}$) and $128\,000$ ($80 \times 40 \times 40$, for $0.05\,\text{Hz}$) cells. The run times for each frequency, the corresponding number of cells, minimum cell width and computation domain stretching factor are listed in Table 1. The total run time to compute this model was less than two minutes.

Figure 4 (a) shows the frequency-domain result, where the blue dots are the computed responses and the black dots correspond to the interpolated values or the values set to zero. Most of the computed values stay below a relative error of $1\,\%$, our chosen adaptive gridding only starts to generate considerable errors at higher frequencies. Figure 4(b) shows the corresponding time-domain result, where the dashed black line is the result from `emg3d`, on top of the red line which is the analytical result. The relative error is mostly below $1\,\%$, except for early times. However, for practical reasons that is more than enough. Figure 5 shows the same on a logarithmic scale, with times up to 10 seconds. It clearly shows that if later times are required, we would need to adjust our Fourier transform parameters. Note that for the gridding we chose $n_\delta = 12$, which is very dense. This was necessary because we are relatively close to the source. If the offsets of interest are larger this factor can be lowered considerably; 3–4 is often enough.

This model corresponds to the one presented in Table 1 and in Figures 3 and 4 of Mulder et al. (2008). The response here appears to be more accurate, their reported peak-error is roughly $1\,\%$, whereas we are below $0.1\,\%$ at the peak (there are no error-plots presented, so visual inspection is all we have). However, the difference in run time is dramatic. Summing the run times for the different frequencies of the original figure comes to a total computation

time of $3\,\mathrm{h}$ $47\,\mathrm{min}$ $12\,\mathrm{s}$; $0.01\,\mathrm{Hz}$ was the slowest run with $31\,\mathrm{min}$ $19\,\mathrm{s}$, and $2.37\,\mathrm{Hz}$ was the fastest run with $2\,\mathrm{min}$ $54\,\mathrm{s}$. Our example, on the other hand, took less than two minutes in total, where the individual frequencies took between 3 and $15\,\mathrm{s}$ to run.

This massive speed-up has a couple of reasons. Computers have become more powerful in the last twelve years, and the codes were run on different computers. A quick test with the old scripts on our test machine shows that it would roughly run 2–3 times faster, therefore somewhere between 1 and 2 hours. The more important facts besides different hardware are: (1) we only used 14 frequencies instead of the 26 frequencies between 0.01 and $100\,\mathrm{Hz}$ of the original; (2) our adaptive gridding used significantly less cells (f-dependent) in comparison to the fixed $2\,097\,152$ cells (128^3) used in the original example. We did not see a significant difference in the speed of the actual codes, where the kernel-algorithm of the two implementations is the same, but in the original example it is implemented in Matlab/C, whereas `emg3d` is written in Python/Numba (Numba is a just-in-time compiler for Python code, Lam et al., 2015).

1D Model

The second example is a shallow marine, layered model with $200\,\mathrm{m}$ of seawater ($3\,\mathrm{S/m}$) above a halfspace of $1\,\mathrm{S/m}$, an embedded target layer at $2\,\mathrm{km}$ depth, $100\,\mathrm{m}$ thick, with a conductivity of $0.02\,\mathrm{S/m}$. The source is located $20\,\mathrm{m}$ above the seafloor and the receivers are on the seafloor. We chose the frequency range such that we can model offsets from 3 to 7 kilometers, with $f_{\min} = 0.007\,\mathrm{Hz}$ and $f_{\max} = 32\,\mathrm{Hz}$, using FFTLog with 5 frequencies per decade. This results in computations for 19 frequencies from $0.008\,\mathrm{Hz}$ to $31.8\,\mathrm{Hz}$. The complete frequency range for the transform includes 35 frequencies from $2\,10^{-5}$ to $126.4\,\mathrm{Hz}$.

For the adaptive gridding, we used a cell width of 100 m in the core domain and stretching outside up to a factor 1.5, where the computation domain extends up to 50 km in each direction. This yielded grids between 204 800 (higher frequencies) and 245 760 (lower frequencies) cells. The run times for each frequency and their corresponding parameters are listed in Table 2.

Figure 6 shows the result for an offset of 5 km, in (a) the frequency and (b) the time domain. The recovered response with the 3D code captures the airwave (first peak) and the subsurface (second peak) very accurately. At later times the error starts to increase. We would need to compute a few additional lower frequencies if we want to improve it.   In the frequency-domain plot, it can be seen that the high frequencies are not computed very accurately, but without too much influence on the time-domain response. These frequencies could be left out if an offset of 5 km is the only objective. However, we also want to retrieve shorter offsets from the same computation, for which these frequencies are required.

Figure 7 shows the time-domain responses of the same model for offsets of 3, 5, and 7 km, all obtained with the same frequency-domain computations and the same frequencies for the Fourier transform. The computation of these frequencies took less than 13 minutes and it handles any offset between 3 and 7 km. It can be seen that the chosen frequency selection is sufficient for this offset range; again, more low frequencies could be added to improve late-time values.

Horizontal extent of the computation domain

The skin-depth approach fails for the air layer, as explained in the Section *Gridding*. The reason is that the EM field in the air travels at the speed of light as a wave, and its amplitude

16

is only reduced through geometrical spreading. On land and in shallow marine scenarios one has therefore to include a sufficiently large computational domain, and the default in our scheme is 100 km. The important point is that this does not only apply to the upward z-direction, but also to the horizontal directions, as the airwave also bounces back horizontally and would continuously emit energy into the subsurface if the boundaries are not chosen far enough away from the receivers. If models are computed with very resistive layers or models with highly resistive basements, this can even apply to deep marine scenarios.

Figure 8 shows this effect. It is the same model as in the previous section; however, for the adaptive gridding in the horizontal directions $\rho_\mathrm{ave} = 1\,\Omega\,\mathrm{m}$ was used instead of $\rho_\mathrm{ave} = 10\,000\,\Omega\,\mathrm{m}$. Having the boundaries too near in the horizontal directions leads to worse results for most frequencies and entirely wrong results for high frequencies. Comparison with the 1D result in the time domain shows that it is the airwave whose amplitude is heavily overestimated. It can be difficult to spot these errors in the time-domain result, as the response looks plausible and only comparing it with the 1D result reveals that it is actually wrong. A possibility to detect such problems for complicated cases, where there is no semi-analytical result to compare with, is to compute two or more models, moving the boundary. When the responses stop to change, one can assume that the boundary is far enough. Another possibility is to look at the amplitudes close to the boundaries and ensure that they are small enough.

3D Model

The final example consists of a resistive, three-dimensional block embedded in the lower of two halfspaces, as depicted in Figure 9. The target has resistivity $\rho_\mathrm{tg} = 100\,\Omega\,\mathrm{m}$, the

upper halfspace corresponds to seawater with $\rho_{\text{sea}} = 0.3\,\Omega\,\text{m}$, and the lower halfspace is the background with $\rho_{\text{bg}} = 1\,\Omega\,\text{m}$. The source is a 100-m long, x-directed dipole at the origin, $50\,\text{m}$ above the seafloor, and we are using a step-off source function. The x-directed inline receiver is at an offset of $2\,\text{km}$. The dimension of the target cube is $1.1 \times 1.0 \times 0.4\,\text{km}$, located $300\,\text{m}$ below the seafloor in the centre between source and receiver.

For the comparison we use the open-source code `SimPEG` (Cockett et al., 2015). `SimPEG` is a framework for modelling and inversion of geophysical data such as gravity, magnetics, and CSEM. It has Maxwell's equations implemented in both the frequency and time domain. As such we can compare our result computed in the frequency domain followed by a Fourier transform to a result computed directly in the time domain. A principal difference between `SimPEG` and `emg3d` is that the former has various direct solvers implemented, whereas the latter is an iterative multigrid solver. The 3D model is therefore a rather small example in order to be able to run it on our test machine, as the memory requirement by the direct solver would otherwise be too high. There are not many options out there of open-source time-domain 3D codes, `SimPEG` being the one we found to be suitable. A step-off response was chosen as this is the response currently implemented in `SimPEG`.

The model was discretised with $100 \times 100 \times 100\,\text{m}$ cells in the survey domain D_{s}. For the `SimPEG` model, 14 cells in x-direction and 12 cells in y- and z-directions were used on both sides with a stretching of 1.3 for the total computation domain D_{c}, which yields a mesh of $58\,344$ cells. The time-steps start at $0.1\,\text{s}$ and are: $21 \times 0.01\,\text{s}$, $23 \times 0.03\,\text{s}$, $21 \times 0.1\,\text{s}$, $23 \times 0.3\,\text{s}$, covering exactly the desired range of 0.1–$10\,\text{s}$. For `emg3d`, the mesh is generated frequency-dependent as in the previous examples, with a maximum stretching of $\alpha_{\text{c}} = 1.5$. This results in meshes between $18\,432$ cells for the highest frequencies and $76\,800$ cells for the lowest frequencies. The required frequencies were obtained by using the FFTLog with

five points per decade, which results in 20 frequencies between 0.001 Hz and 8 Hz. The actual transform was carried out with the 201-point sine-cosine filter from Key (2009).

The results are shown in Figure 10: In (a) the 1D background responses and the relative error using the semi-analytical result obtained from `empymod`, and in (b) the responses including the target. The background comparison shows that both 3D codes do an acceptable job having a relative error of a few percents at most; the result for `emg3d` seems to be better at early times. We cannot compare the errors for the response that includes the target for lack of an analytical solution. The background model from `empymod` is only included to show that there is a significant response from the target. We therefore show the normalized difference (ND) between the two 3D codes as a percentage, where $\text{ND} = 200|\texttt{SimPEG} - \texttt{emg3d}|/(|\texttt{SimPEG}| + |\texttt{emg3d}|)$. The ND between the two codes is below 1 % everywhere except for early times.

Both codes took a little over 7 minutes to compute the two models (single thread). However, in this particular comparison, the main difference in runtime is not frequency-domain computation vs. time-domain computation, but iterative solver vs. direct solver. For a test we doubled the number of cells in each direction by using 50 m for the minimum cell width instead of 100 m, and taking also half of the stretching, which results then in a mesh of 466 752 cells for `SimPEG` and meshes between 102 400 and 614 400 for `emg3d`. Because the memory of our test machine was not large enough for the direct solver of `SimPEG`, the last model had to be run on a cluster. In this scenario, `emg3d` was about eight times faster (single thread).

CONCLUSIONS

We have shown a method to minimize the required frequencies and their range for the computation of time-domain CSEM data with a frequency-domain code. This can significantly reduce the computation time and makes time-domain CSEM modelling with a frequency-domain code competitive given a robust frequency-domain solver, a frequency-dependent gridding function that minimizes the required cells, and a Fourier transform that works on a logarithmic scale. Fast layered modelling can be used to design the required frequency range, as the Fourier transform does not know about the dimensionality of the underlying model. Twenty frequencies or less are usually sufficient for a wide range of offsets. We believe that our proposed improvements to the previously published methods make time-domain CSEM modelling with a frequency-domain code even more attractive than it was already before. We have demonstrated the idea of our Fourier transform method on CSEM data transformed from the frequency to the time domain. However, it could equally be applied to the transform from the time to the frequency domain and to other methods with similar characteristics.

ACKNOWLEDGMENT

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available at Zenodo at doi: 10.5281/zenodo.???????. The data includes the scripts and instructions to reproduce all results and figures.

Note to editors and reviewers: The data are NOT YET on Zenodo, we will upload there the final versions and link the right DOI for the final manuscript. Until then everything can be found on GitHub at github.com/empymod/article-TDEM.

REFERENCES

Andréis, D., and L. MacGregor, 2007, Time-domain versus frequency-domain CSEM in shallow water: SEG Technical Program Expanded Abstracts, 643–647; doi: 10.1190/1.2792500.

Avdeev, D. B., 2005, Three-dimensional electromagnetic modelling and inversion from theory to application: Surveys in Geophysics, **26**, no. 6, 767–799; doi: 10.1007/s10712-005-1836-x.

Avdeeva, A., M. Commer, and G. A. Newman, 2007, Hydrocarbon reservoir detectability study for marine CSEM methods: Time domain versus frequency domain: SEG Technical Program Expanded Abstracts, 628–632; doi: 10.1190/1.2792497.

Börner, R.-U., 2010, Numerical modelling in geo-electromagnetics: Advances and challenges: Surveys in Geophysics, **31**, no. 2, 225–245; doi: 10.1007/s10712-009-9087-x.

Cai, H., X. Hu, J. Li, M. Endo, and B. Xiong, 2017, Parallelized 3D CSEM modeling using edge-based finite element with total field formulation and unstructured mesh: Computers & Geosciences, **99**, 125–134; doi: 10.1016/j.cageo.2016.11.009.

Cockett, R., S. Kang, L. J. Heagy, A. Pidlisecky, and D. W. Oldenburg, 2015, SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications: Computers & Geosciences, **85**, 142–154; doi: 10.1016/j.cageo.2015.09.015.

Connell, D., and K. Key, 2013, A numerical comparison of time and frequency-domain marine electromagnetic methods for hydrocarbon exploration in shallow water: Geophysical Prospecting, **61**, 187–199; doi: 10.1111/j.1365-2478.2012.01037.x.

Constable, S., 2010, Ten years of marine CSEM for hydrocarbon exploration: Geophysics, **75**, no. 5, 75A67–75A81; doi: 10.1190/1.3483451.

Davydycheva, S., V. Druskin, and T. Habashy, 2003, An efficient finite-difference scheme for electromagnetic logging in 3D anisotropic inhomogeneous media: Geophysics, **68**, no. 5, 1525–1536; doi: 10.1190/1.1620626.

Fritsch, F., and R. Carlson, 1980, Monotone piecewise cubic interpolation: SIAM Journal on Numerical Analysis, **17**, no. 2, 238–246; doi: 10.1137/0717021.

Gehrmann, R. A. S., L. J. North, S. Graber, F. Szitkar, S. Petersen, T. A. Minshull, and B. J. Murton, 2019, Marine mineral exploration with controlled source electromagnetics at the tag hydrothermal field, 26°n mid-atlantic ridge: Geophysical Research Letters, **46**, no. 11, 5808–5816; doi: 10.1029/2019GL082928.

Ghosh, D. P., 1971, The application of linear filter theory to the direct interpretation of geoelectrical resistivity sounding measurements: Geophysical Prospecting, **19**, no. 2, 192–217; doi: 10.1111/j.1365-2478.1971.tb00593.x.

Girard, J.-F., N. Coppo, P. Wawrzyniak, B. Bourgeois, J.-M. Baltassat, and A. Gadalia, 2015, Resistivity imaging for geothermal exploration, using controlled-source em where magneto-telluric is not applicable: Model and field study.: Proceedings World Geothermal Congress, 1–8.

Grayver, A. V., and T. V. Kolev, 2015, Large-scale 3D geoelectromagnetic modeling using parallel adaptive high-order finite element method: Geophysics, **80**, no. 6, E277–E291; doi: 10.1190/geo2015-0013.1.

Grayver, A. V., R. Streich, and O. Ritter, 2013, Three-dimensional parallel distributed inversion of CSEM data using a direct forward solver: Geophysical Journal International, **193**, no. 3, 1432–1446; doi: 10.1093/gji/ggt055.

Hamilton, A. J. S., 2000, Uncorrelated modes of the non-linear power spectrum: Monthly Notices of the Royal Astronomical Society, **312**, no. 2, 257–284; doi: 10.1046/j.1365-

8711.2000.03071.x.

Hunziker, J., J. Thorbecke, and E. Slob, 2015, The electromagnetic response in a layered vertical transverse isotropic medium: A new look at an old problem: Geophysics, **80**, no. 1, F1–F18; doi: 10.1190/geo2013-0411.1.

Jahandari, H., and C. G. Farquharson, 2014, A finite-volume solution to the geophysical electromagnetic forward problem using unstructured grids: Geophysics, **79**, no. 6, E287–E302; doi: 10.1190/geo2013-0312.1.

Jaysaval, P., D. V. Shantsev, and S. de la Kethulle de Ryhove, 2015, Efficient 3-D controlled-source electromagnetic modelling using an exponential finite-difference method: Geophysical Journal International, **203**, no. 3, 1541–1574; doi: 10.1093/gji/ggv377.

Johansen, S. E., M. Panzner, R. Mittet, H. E. F. Amundsen, A. Lim, E. Vik, M. Landrø, and B. Arntsen, 2019, Deep electrical imaging of the ultraslow-spreading Mohns Ridge: Nature, **567**, no. 7748, 379–383; doi: 10.1038/s41586-019-1010-0.

Jönsthövel, T. B., C. W. Oosterlee, and W. A. Mulder, 2006, Improving multigrid for 3-D electro-magnetic diffusion on stretched grids: European Conference on Computational Fluid Dynamics; UUID: df65da5c-e43f-47ab-b80d-2f8ee7f35464.

Key, K., 2009, 1D inversion of multicomponent, multifrequency marine CSEM data: Methodology and synthetic studies for resolving thin resistive layers: Geophysics, **74**, no. 2, F9–F20; doi: 10.1190/1.3058434.

Kruglyakov, M., and L. Bloshanskaya, 2017, High-performance parallel solver for integral equations of electromagnetics based on Galerkin method: Mathematical Geosciences, **49**, no. 6, 751–776; doi: 10.1007/s11004-017-9677-y.

Lam, S. K., A. Pitrou, and S. Seibert, 2015, Numba: A LLVM-based Python JIT Compiler: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC,

ACM, 7:1–7:6; doi: 10.1145/2833157.2833162.

Lebedev, V. I., 1964, Difference analogues of orthogonal decompositions, basic differential operators and some boundary problems of mathematical physics. I: USSR Computational Mathematics and Mathematical Physics, **4**, no. 3, 69–92; doi: 10.1016/0041-5553(64)90240-X.

Liu, R., R. Guo, J. Liu, C. Ma, and Z. Guo, 2018, A hybrid solver based on the integral equation method and vector finite-element method for 3D controlled-source electromagnetic method modeling: Geophysics, **83**, no. 5, E319–E333; doi: 10.1190/geo2017-0502.1.

Mulder, W. A., 2006, A multigrid solver for 3D electromagnetic diffusion: Geophysical Prospecting, **54**, no. 5, 633–649; doi: 10.1111/j.1365-2478.2006.00558.x.

——, 2020, *in* Numerical Methods, Multigrid: Springer International Publishing, 1–6; doi: 978-3-030-10475-7_153-1.

Mulder, W. A., M. Wirianto, and E. Slob, 2008, Time-domain modeling of electromagnetic diffusion with a frequency-domain code: Geophysics, **73**, no. 1, F1–F8; doi: 10.1190/1.2799093.

Oristaglio, M., and B. Spies, 1999, Three-dimensional electromagnetics: SEG, volume **7** *of* Geophysical Developments; doi: 10.1190/1.9781560802154.

Pedersen, L. B., M. Bastani, and L. Dynesius, 2005, Groundwater exploration using combined controlled-source and radiomagnetotelluric techniques: Geophysics, **70**, no. 1, G8–G15; doi: 10.1190/1.1852774.

Plessix, R.-E., M. Darnet, and W. A. Mulder, 2007, An approach for 3D multi-source, multifrequency CSEM modeling: Geophysics, **72**, no. 5, SM177–SM184; doi: 10.1190/1.2744234.

Rochlitz, R., N. Skibbe, and T. Günther, 2019, custEM: customizable finite element simula-

tion of complex controlled-source electromagnetic data: Geophysics, **84**, no. 2, F17–F33; doi: 10.1190/geo2018-0208.1.

Sommer, M., S. Hölz, M. Moorkamp, A. Swidinsky, B. Heincke, C. Scholl, and M. Jegen, 2013, GPU parallelization of a three dimensional marine CSEM code: Computers & Geosciences, **58**, 91–99; doi: 10.1016/j.cageo.2013.04.004.

Ward, S. H., and G. W. Hohmann, 1988, Electromagnetic theory for geophysical applications: SEG, Investigations in Geophysics, 4, 130–311; doi: 10.1190/1.9781560802631.ch4.

Werthmüller, D., 2017, An open-source full 3D electromagnetic modeler for 1D VTI media in Python: empymod: Geophysics, **82**, no. 6, WB9–WB19; doi: 10.1190/geo2016-0626.1.

Werthmüller, D., K. Key, and E. Slob, 2019a, A tool for designing digital filters for the Hankel and Fourier transforms in potential, diffusive, and wavefield modeling: Geophysics, **84**, no. 2, F47–F56; doi: 10.1190/geo2018-0069.1.

Werthmüller, D., W. A. Mulder, and E. C. Slob, 2019b, emg3d: A multigrid solver for 3D electromagnetic diffusion: Journal of Open Source Software, **4**, no. 39, 1463; doi: 10.21105/joss.01463.

———, 2020, Time-domain CSEM modelling using frequency- and Laplace-domain computations: EAGE Technical Program Expanded Abstracts, 319.

Werthmüller, D., A. Ziolkowski, and D. Wright, 2014, Predicting controlled-source electromagnetic responses from seismic velocities: Interpretation, **2**, no. 3, SH115–SH131; doi: 10.1190/INT-2013-0153.1.

Yee, K., 1966, Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media: IEEE Transactions on Antennas and Propagation, **14**, no. 3, 302–307; doi: 10.1109/TAP.1966.1138693.

Ziolkowski, A., B. Hobbs, and D. Wright, 2007, Multitransient electromagnetic demonstra-

tion survey in france: Geophysics, **72**, F197–F209; doi: 10.1190/1.2735802.

LIST OF TABLES

LIST OF FIGURES

in Figure 6.

8   Same model as used for Figure 6, but with the horizontal boundaries not far enough. Although the resulting time-domain result looks plausible, the comparison with the 1D result shows that it significantly overestimates the amplitude of the airwave.

9   Three-dimensional block embedded in the lower of two halfspaces. The 100-m long, x-directed dipole source is located $50\,$m above the seafloor at the origin, and the receiver is on the seafloor at an offset of $2\,$km.

10   Responses for the model outlined in Figure 9 using `SimPEG` and `emg3d`, and for the layered background also `empymod`. In the lower plot of (a) the relative error (%) in comparison to `empymod` is shown, and in (b) the normalized difference (%) between the two 3D codes. The results of these plots are clipped for values smaller than $0.1\,$% and bigger than $10\,$%.
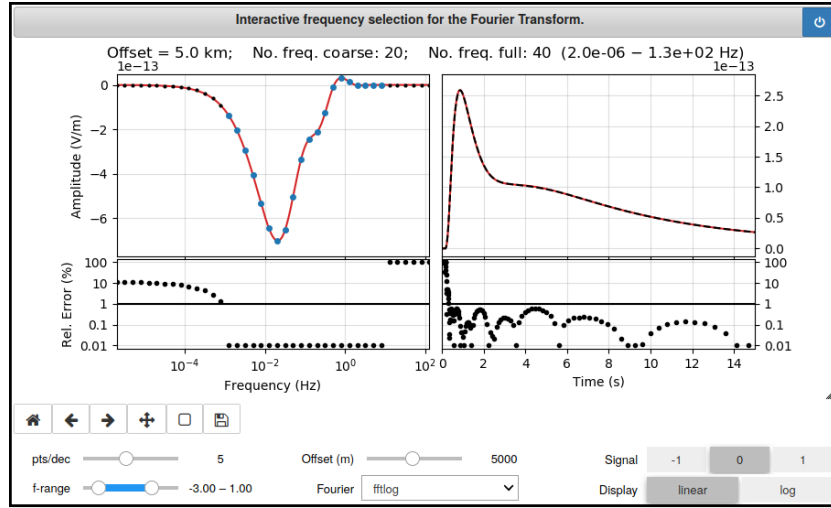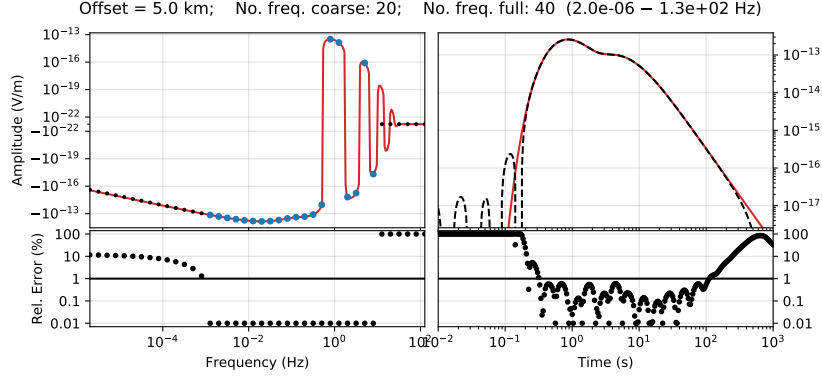
Figure 1: Interactive frequency selection for a user-provided layered model (the shown model parameters are described in the text). This example shows the impulse response at an offset of 5 km, for which it uses FFTLog from 0.001 Hz to 10 Hz with five frequencies per decade. Note that the error is clipped for values smaller than 0.01 % and bigger than 100 %.

Werthmüller et al. –

(a)
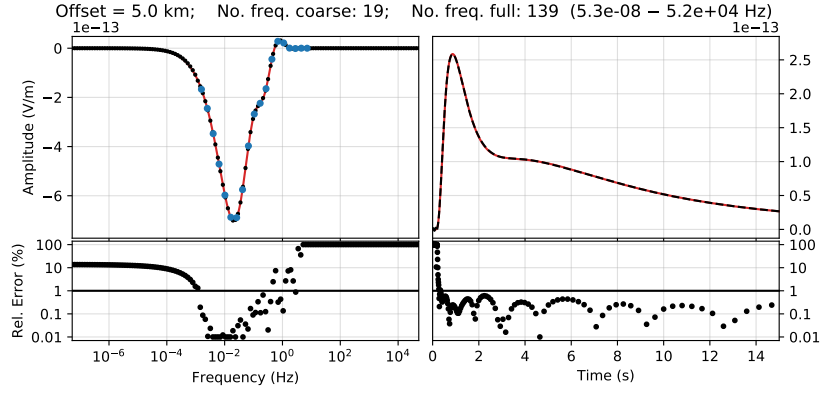
(b)

Figure 2: (a) Same as Figure 1, but on a logarithmic scale. (b) Using DLF instead of FFTLog, but computing the same frequencies as for the FFTLog. The resulting time domain response has comparable accuracy (compare the error to the error in Figure 1).
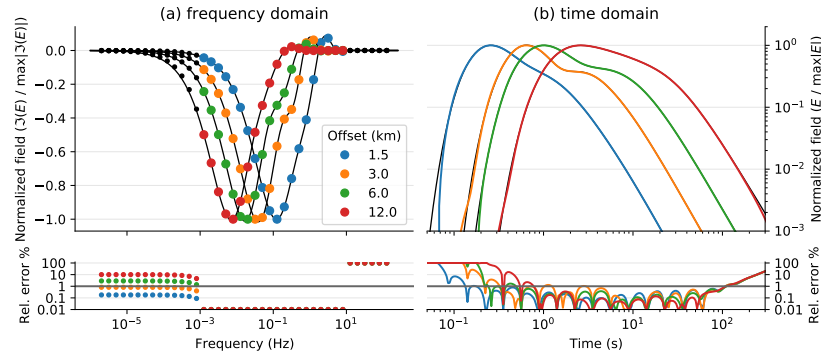
Werthmüller et al. –

Figure 3: Normalized (a) frequency- and (b) time-domain responses for offsets $r = 1.5, 3, 6,$ and $12\,\mathrm{km}$ using the parameters defined in Figure 1. The coloured circles are the actually computed responses, the black dots are the responses which are set to zero or interpolated. The black curves are the analytical responses.

Werthmüller et al. –

freq. (Hz)	time (s)	nx×ny×nz	Δ_{min} (m)	α_{c}
20.0	3	80×24×24	20	1.26
12.6	6	96×32×32	20	1.17
7.98	8	96×32×32	20	1.20
5.03	7	96×32×32	20	1.23
3.18	7	80×32×32	24	1.25
2.00	5	80×32×32	30	1.21
1.26	4	64×32×32	37	1.21
0.798	5	64×32×32	40	1.23
0.503	5	64×32×32	40	1.26
0.318	5	64×32×32	40	1.28
0.200	10	64×40×40	40	1.27
0.126	10	64×40×40	40	1.30
0.0798	14	80×40×40	40	1.26
0.0503	15	80×40×40	40	1.28

Table 1: Run times per frequency for the homogeneous space example, with the corresponding number of cells and minimum cell width as well as the stretching factor in the computation domain; $\alpha_{\mathrm{s}} = 1$ everywhere.
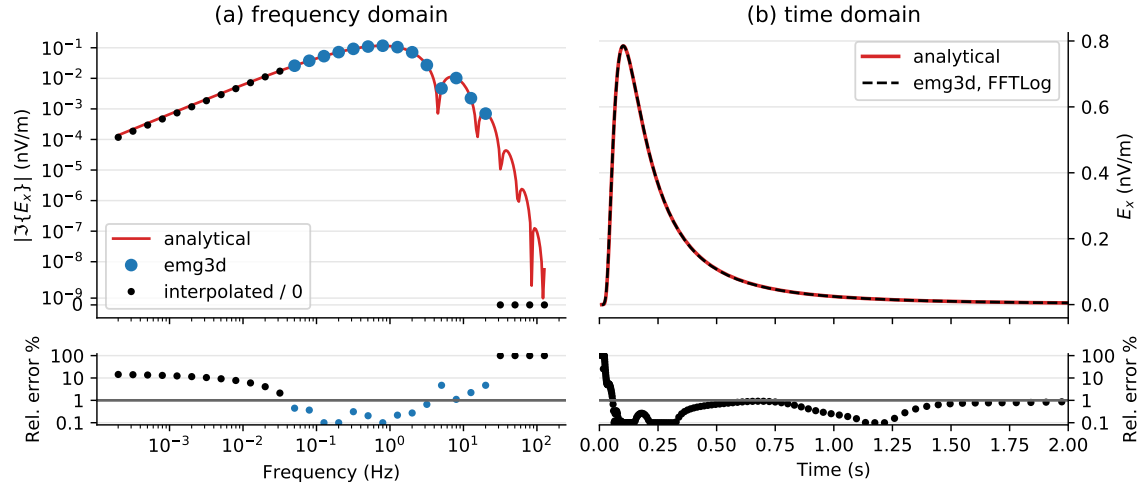
Figure 4: (a) Frequency- and (b) time-domain results for the homogeneous space model. The red lines are the analytical solutions, the blue circles are the actually computed responses with `emg3d`, the black dots are the interpolated responses, and the dashed black line the obtained time-domain response. The errors are clipped for values smaller than $0.1\,\%$ and bigger than $100\,\%$.
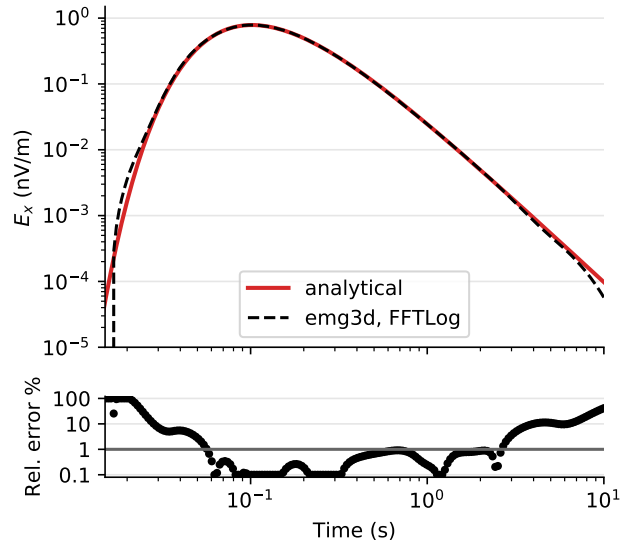
Werthmüller et al. –

Figure 5: Same as in Figure 4 (b), but on a logarithmic scale. To improve later times we would have to compute lower frequencies; to improve earlier times we would have to compute more frequencies per decade to get a better resolution.

Werthmüller et al. –

freq. (Hz)	time (s)	nx×ny×nz	Δ_{min} (m)	α_c
31.8	10	128×40×40	100	1.36
20.0	9	128×40×40	100	1.36
12.6	9	128×40×40	100	1.36
7.98	17	128×40×40	100	1.36
5.03	21	128×40×40	100	1.44
3.18	20	128×40×40	100	1.48
2.00	18	128×40×40	100	1.49
1.26	22	128×40×48	100	1.36
0.798	26	128×40×48	100	1.36
0.503	29	128×40×48	100	1.36
0.318	42	128×40×48	100	1.36
0.200	41	128×40×48	100	1.38
0.126	48	128×40×48	100	1.40
0.0798	54	128×40×48	100	1.41
0.0503	59	128×40×48	100	1.44
0.0318	74	128×40×48	100	1.44
0.0200	81	128×40×48	100	1.47
0.0126	96	128×40×48	100	1.48
0.007 98	93	128×40×48	100	1.49

Table 2: Run times per frequency for the marine 1D example, with the corresponding number of cells and minimum cell width as well as the stretching factor in the computation domain; $\alpha_s = 1$ everywhere.
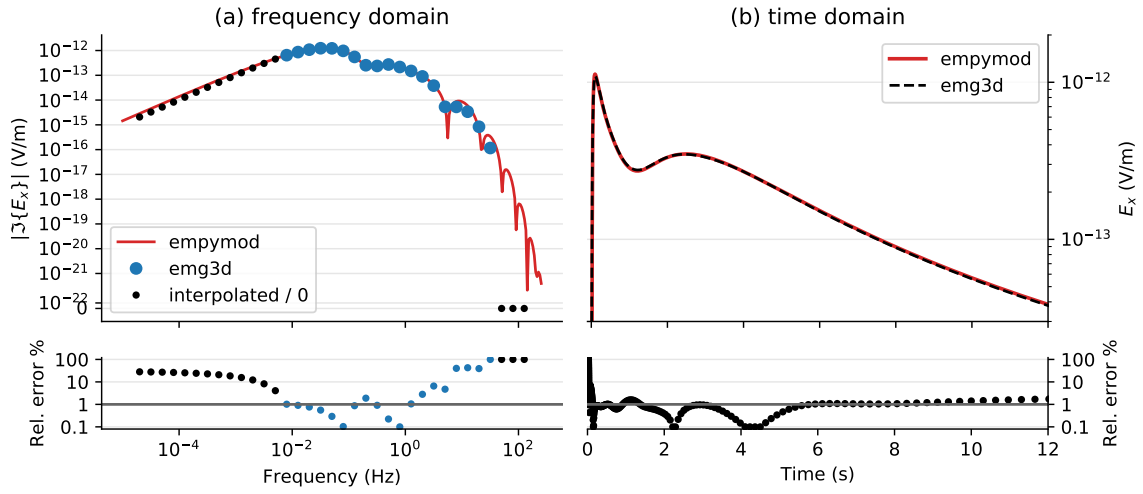
Figure 6: Electric inline response at an offset of 5 km for a shallow marine, layered scenario. (a) Frequency-domain, where the blue circles are computed responses and the black dots are interpolated responses or responses set to zero. (b) Time-domain response.
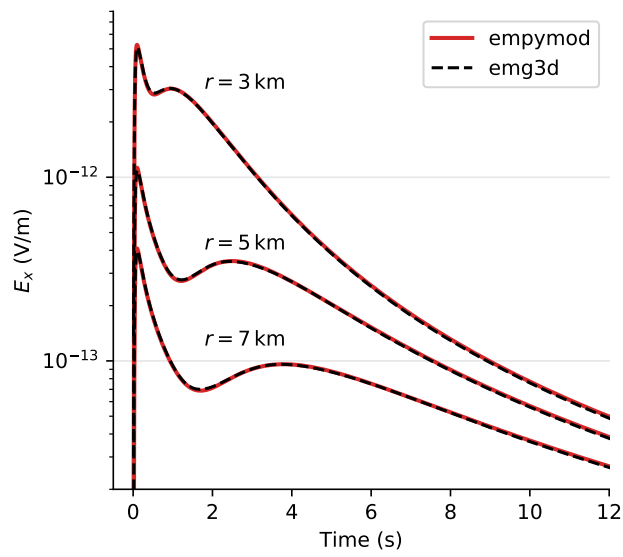
Werthmüller et al. –

Figure 7: Time-domain responses for offsets of 3, 5, and 7 km for the same model as shown in Figure 6.
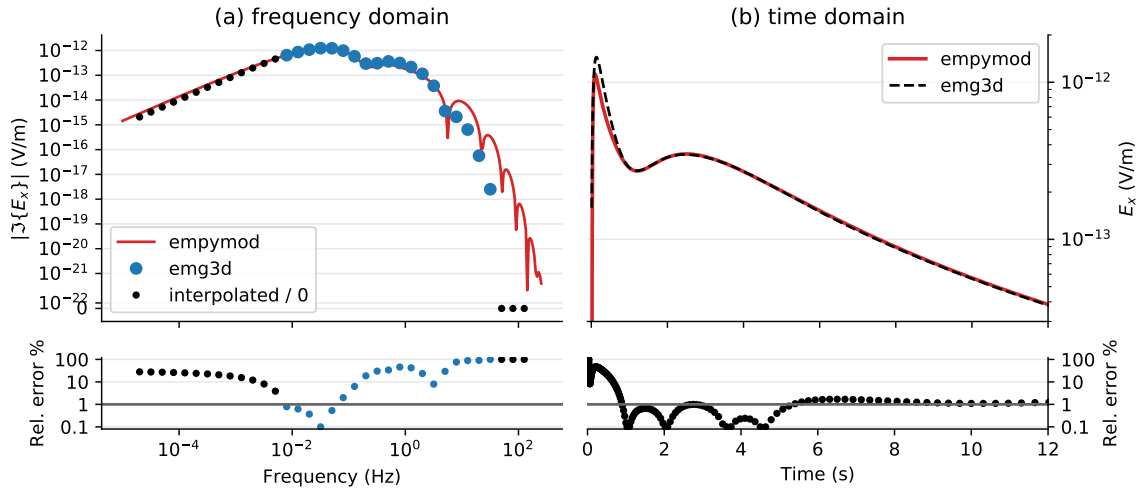
Werthmüller et al. –

Figure 8: Same model as used for Figure 6, but with the horizontal boundaries not far enough. Although the resulting time-domain result looks plausible, the comparison with the 1D result shows that it significantly overestimates the amplitude of the airwave.
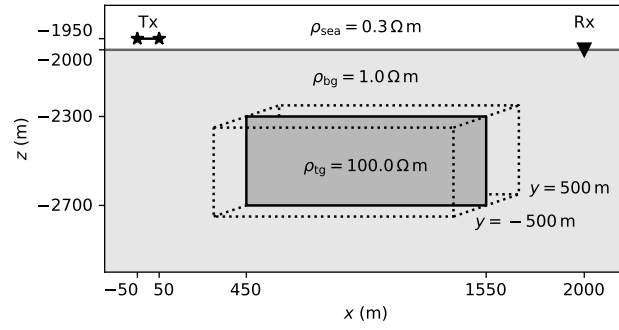
Werthmüller et al. –

Figure 9: Three-dimensional block embedded in the lower of two halfspaces. The 100-m long, x-directed dipole source is located $50\,\mathrm{m}$ above the seafloor at the origin, and the receiver is on the seafloor at an offset of $2\,\mathrm{km}$.
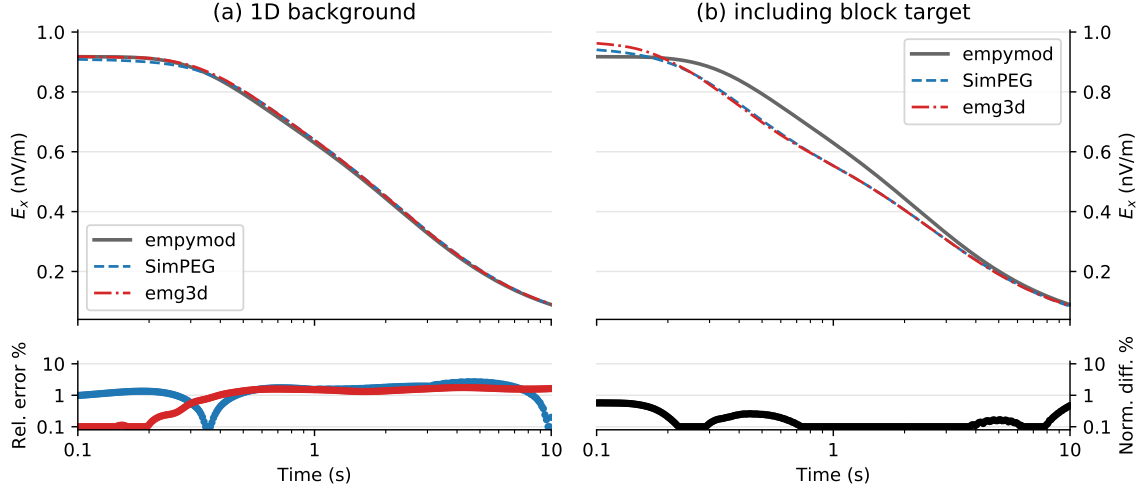
Werthmüller et al. –

Figure 10: Responses for the model outlined in Figure 9 using `SimPEG` and `emg3d`, and for the layered background also `empymod`. In the lower plot of (a) the relative error (%) in comparison to `empymod` is shown, and in (b) the normalized difference (%) between the two 3D codes. The results of these plots are clipped for values smaller than $0.1\,\%$ and bigger than $10\,\%$.

Werthmüller et al. –