

Getting started with controlled-source electromagnetic 1D modeling

Dieter Werthmüller¹

Forward modeling is an important part of understanding controlled-source electromagnetic (CSEM) responses. The diffusive term in the electromagnetic wave equation is dominant over the displacement term at these frequencies. It is the diffusive behavior that makes it difficult to imagine the actual propagation of the signal. An important tool in gaining experience therefore is forward modeling, and lots of it. The advantage of one-dimensional (1D) forward modeling, besides its speed, is to study isolated effects (see for instance Key, 2009): What is the influence of resistivity anisotropy, or of fine-scale resistivity variations? What is the influence of the airwave? With 1D modeling you can quickly study these effects in isolation before you go on to more complex models in higher dimensions. For an introduction to CSEM for hydrocarbon exploration see, for instance, Constable (2010).

Luckily the EM community can count on some freely available, stable, high-quality, and open-source CSEM modeling codes. Well-known ones are the codes from the Marine EM Lab at Scripps: *DIPOLE1D* (Key, 2009) and *MARE2DEM* (Key and Owall, 2011) for 1D and 2D modeling. *DIPOLE1D* is an isotropic EM modeler for a stratified model using the diffusive approximation valid for low frequencies such as used in CSEM (neglecting displacement currents). *MARE2DEM* is a 2.5D CSEM and magnetotelluric (MT) modeling and inversion code, including resistivity anisotropy. More recently, Cockett et al. (2015) published *SimPEG*, and Hunziker et al. (2015) *EMmod*. *SimPEG* is a framework for 3D simulation and modeling of all sorts of geophysical methods, among them CSEM. *EMmod* is a 1D electromagnetic modeler including anisotropy in the form of vertical transverse isotropic (VTI) resistivity as well as VTI electric permittivity and VTI magnetic permeability, therefore modeling the whole wavefield for electric and magnetic sources and receivers.

In this tutorial, I use the 1D electromagnetic forward modeling code *empymod*. The code is published under a permissive open-source license and available at github.com/prisae/empymod. It is based on Hunziker et al. (2015) for the wavenumber-domain calculation, and on Key (2012) for the Hankel and Fourier transforms. The electromagnetic Python modeler *empymod* can model electric or magnetic responses due to a three-dimensional electric or magnetic source in a layered earth model (Figure 1) with n layers, each with electric vertical isotropy (ρ_h, λ), horizontal and vertical electric permittivity (ϵ_h, ϵ_v), and horizontal and vertical magnetic permeability (μ_h, μ_v), from very low frequencies ($f \sim 0$ Hz) to very high frequencies ($f \sim$ GHz). The layer parameters all default to 1 if not provided (ρ_h is mandatory).

Have a look at the accompanying notebook at <http://github.com/seg> to work through the code yourself and to see more examples and more instructions.

A first example

We start with a simple model of a half-space below air:

- air/half-space interface at 0 m;
- electric, x -directed source at $x = 0$ m, $y = 0$ m, $z = 1$ m;
- electric receiver at $x = 1000$ m, $y = 0$ m, $z = 1$ m, with azimuth = 45° and dip = 10°; and
- frequency of 2 Hz.

The function *bipole* takes up to 24 arguments, but 19 of those have default values. We only have to provide the five mandatory parameters for this simple example: sources, receivers, depths of the interfaces, resistivities, and frequencies:

```
>>> from empymod import bipole
>>> inp1 = {'src': [0, 0, 1, 0, 0],
>>>         'rec': [1000, 0, 1, 45, 10],
>>>         'depth': 0,
>>>         'res': [2e14, 100],
>>>         'freqtime': 2} # Frequency
>>> bipole(**inp1)
empymod END; runtime=0:00:00.003869; 3 kernel calls
array((2.20420888976397e-08 -7.1538671654131e-10j))
```

From this example, you can infer that an infinitesimally small *dipole* is defined as $[x, y, z, \text{azimuth}, \text{dip}]$, using a left-handed coordinate system with positive z pointing downward. The azimuth is the anti-clockwise deviation from the x -axis, and the dip is the deviation downward from the xy -plane. A finite length *bipole*, on the other hand, would be defined as $[x_1, x_2, y_1, y_2, z_1, z_2]$, where

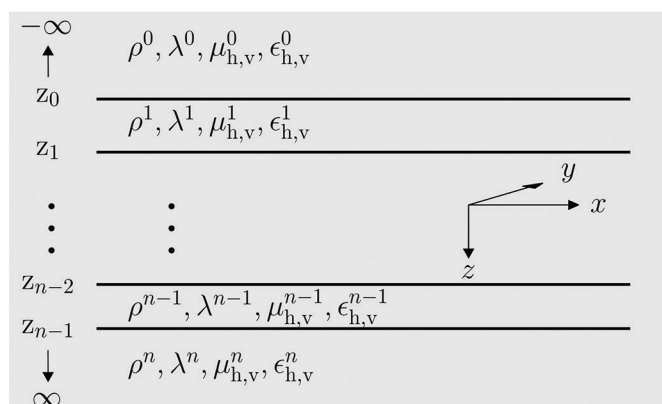


Figure 1. A layered earth model. Each horizontal layer is parameterized by horizontal resistivity and anisotropy, horizontal and vertical magnetic permeability, and horizontal and vertical electric permittivity.

¹Geophysical consultant.

$[x_1, y_1, z_1]$ is the first pole location, and $[x_2, y_2, z_2]$ is the second pole location. We will see in the next example why the frequency parameter is called `freqtime`.

The default verbosity reveals that `empymod` took some milliseconds to run on my machine, and three kernel calls were required. This is because our receiver has an arbitrary rotation, so it had to calculate the x -, y -, and z -fields due to an x -directed source. And it returns a complex number, the result.

Frequency and time domain

Let's look at a four-layer isotropic model (air, overburden, target, underburden), with and without hydrocarbons. The resistivities in the two models are:

```
>>> bri = [2e14, 20, 20, 20]
>>> oil = [2e14, 20, 500, 20]
```

We will keep it simple in terms of source and receiver, with an x -directed source and an x -directed inline receiver at the same offset as before. Note that I shift the source and receiver to $z = 0.01$ m, because `empymod` places electrodes on a layer interface in the upper layer.

```
>>> inp2 = {'src': [0, 0, 0.01, 0, 0],
>>>         'rec': [1000, 0, 0.01, 0, 0],
>>>         'depth': [0, 500, 525]}

>>> f = np.logspace(-1, 3, 100)
>>> fbres = bipole(res=bri, freqtime=f, **inp2)
>>> fores = bipole(res=oil, freqtime=f, **inp2)
```

For the time domain, we have to provide a signal in order to calculate time-domain results instead of frequency-domain results: 0 denotes an impulse response; 1 denotes switch-on; -1 denotes switch-off.

```
>>> t = np.linspace(0, 0.06, 100)
>>> tbres = bipole(res=bri, freqtime=t, signal=-1,
>>>               **inp2)
>>> tores = bipole(res=oil, freqtime=t, signal=-1,
>>>               **inp2)
```

Figure 2 shows the comparison of frequency-domain and time-domain responses for the same model. The jump in the time-domain response is the so-called airwave. Now you see why the frequency-parameter is called `freqtime`: this parameter takes frequencies if `signal=None` (default) or times if `signal` is -1, 0, or 1.

Anisotropy

Anisotropic CSEM modeling, more specifically vertical transverse isotropic (VTI) modeling, is the normal case these days (unlike 10 years ago). Resistivity anisotropy λ and mean resistivity ρ_m are commonly defined in terms of horizontal (ρ_h) and vertical (ρ_v) resistivity as:

$$\lambda = \sqrt{\frac{\rho_v}{\rho_h}}, \quad \text{and} \quad \rho_m = \sqrt{\rho_h \rho_v}.$$

The three resistivities ρ_h , ρ_m , and ρ_v are therefore related through λ :

$$\lambda \rho_h = \rho_m = \frac{1}{\lambda} \rho_v.$$

A common mistake is to only speak about *increasing anisotropy*, without further specification. As can be seen from the above relationship, increasing anisotropy is ambiguous. The relationship between the three resistivities is elaborated in detail in Werthmüller (2009). An example of this ambiguity is given in Figure 3; the code to produce it is in the accompanying notebook.

These results show that stating “increasing anisotropy” can mean quite different things, and it is crucial to specify it more, for instance “increasing anisotropy by keeping the mean resistivity constant.” The isotropic case, the black line, is the same in all three plots. We can draw the following conclusions:

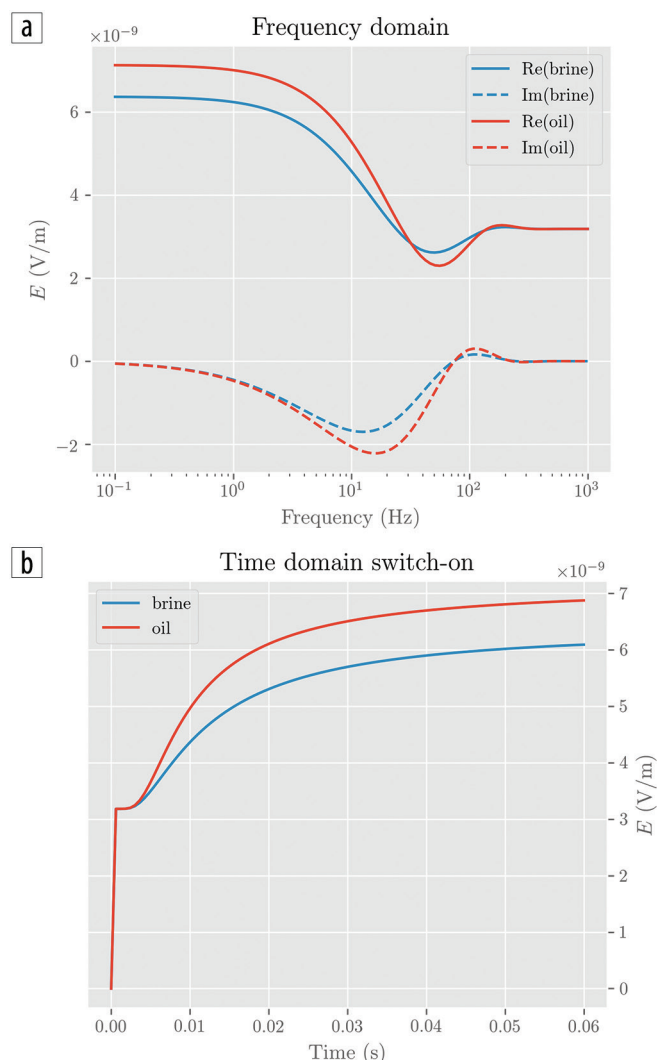


Figure 2. A simple land CSEM example. (a) Frequency- and (b) time-domain responses for the four-layer model, with brine (blue) and oil (red) in the third layer. The offset distance is 1 km.

- 1) The initial step, the airwave, depends only on ρ_h .
- 2) The DC value, hence time $t \rightarrow \infty$, depends only on ρ_m .

These insights can be used to calculate apparent anisotropy values of the subsurface from early and late time CSEM responses.

Interactive modeling

Jupyter and the `ipywidgets` make it incredibly easy to create interactive plots, which I use in the notebook to create an interactive model for `empymod`. All there is to do is to load the widgets, define the desired sliders, create a plot function as we have done for the examples before, and subsequently call `interact`. In the example, I include three sliders: one for the resistivity of the target layer, one for the depth of the target layer, and one for the thickness of the target layer. Five additional lines of code are all that is required to generate this interactive modeler: one line to import the required functions, three lines to define the three sliders, and a last line to activate the interaction. The result is shown in Figure 4.

The interactive example should give you a good idea how to write your own interactive modeler. With a few changes, you could instead have a slider for frequencies, anisotropies, source/receiver azimuth and dip, or any other parameter that goes into bipole.

More examples

The notebook contains, besides much more details, the following four additional examples:

- comparison of bipole versus dipole;
- calculate the amplitude and phase of the entire x, y -plane;
- calculate offset versus frequency crossplot, a common tool for feasibility studies; and
- common-source gather of time-domain responses.

The eight complete examples in the notebook should get you up and running with CSEM modeling. I am open to suggestions

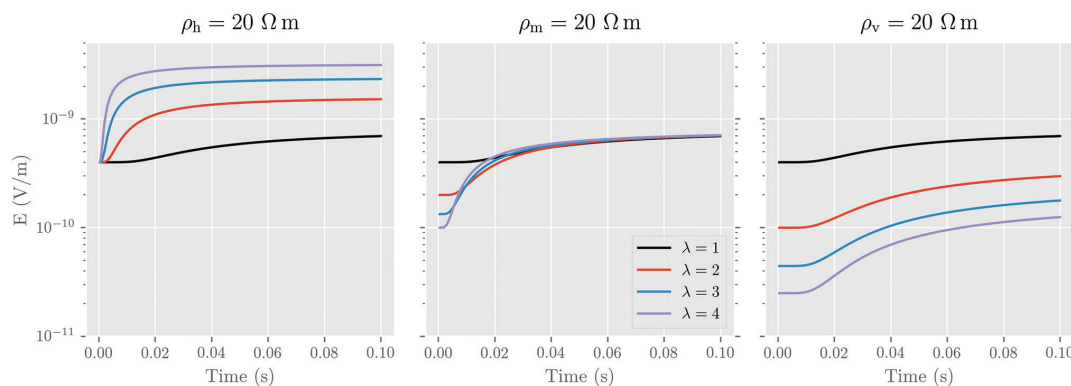


Figure 3. The effect of increasing resistivity anisotropy λ on the time-domain response of the model. The offset distance is 2 km.

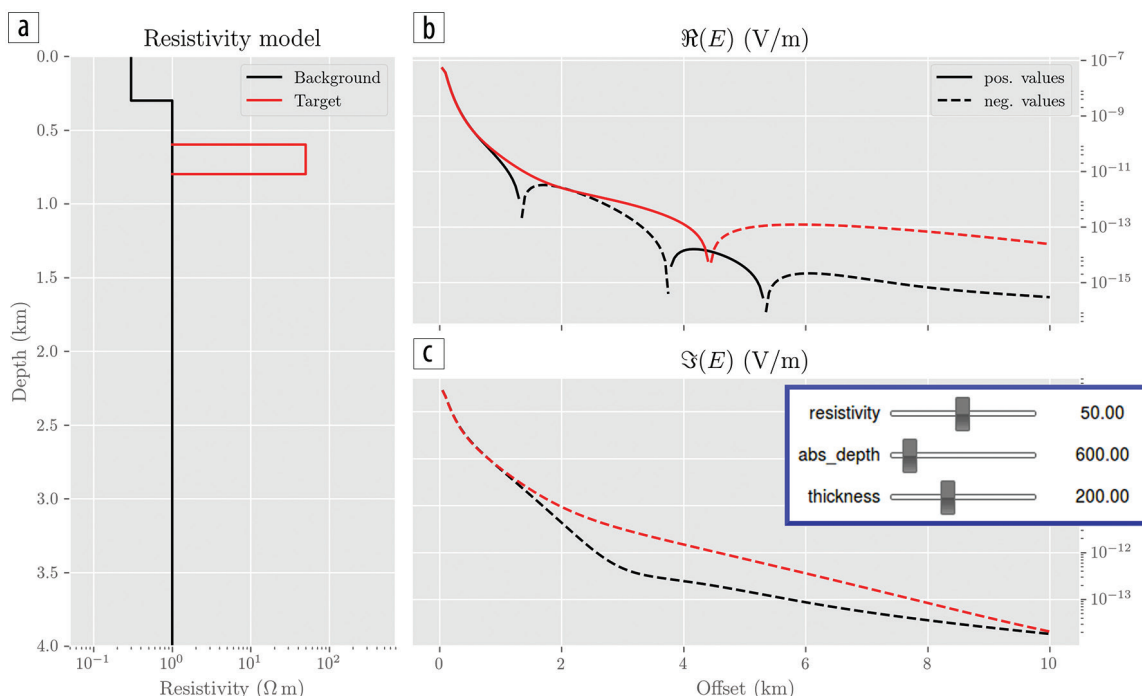



Figure 4. (a) A simple earth model and (b) the real and (c) imaginary responses for an x -directed, electric source at 250 m depth and x -directed, electric receivers at the sea bottom. Water depth is 300 m, and the background resistivity is $1 \Omega m$.

and pull requests, thankful for bug reports, and happy if I see that you fork it and create something awesome of your own. Happy EM modeling! 

Acknowledgments

I would like to thank Matt Hall, Jürg Hunziker, Rowan Cockett, and Lindsey Heagy for comments that greatly improved this tutorial.

Corresponding author: dieter@werthmuller.org

References

- 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, <http://dx.doi.org/10.1016/j.cageo.2015.09.015>. Code at <http://simpeg.xyz>.
- Constable, S., 2010, Ten years of marine CSEM for hydrocarbon exploration: *Geophysics*, **75**, no. 5, 75A67–75A81, <http://dx.doi.org/10.1190/1.3483451>.
- 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, <http://dx.doi.org/10.1190/geo2013-0411.1>. Code at <http://software.seg.org/2015/0001>.
- 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, <http://dx.doi.org/10.1190/1.3058434>. Code at <http://marineemlab.ucsd.edu/Projects/Occam/1DCSEM>.

Key, K., and J. Owall, 2011, A parallel goal-oriented adaptive finite element method for 2.5-D electromagnetic modelling: *Geophysical Journal International*, **186**, no. 1, 137–154, <http://dx.doi.org/10.1111/j.1365-246X.2011.05025.x>. Code at <http://mare-2dem.ucsd.edu>.

Key, K., 2012, Is the fast Hankel transform faster than quadrature?: *Geophysics*, **77**, no. 3, F21–F30, <http://dx.doi.org/10.1190/GEO2011-0237.1>. Code at <http://software.seg.org/2012/0003>.

Werthmüller, D., 2009, Inversion of multi-transient EM data from anisotropic media: M.Sc. thesis, TU Delft, UUID: <http://repository.tudelft.nl/view/ir/uuid:f4b071c1-8e55-4ec5-86c6-a2d54c3eda5a>.



© The Author(s). Published by the Society of Exploration Geophysicists. All article content, except where otherwise noted (including republished material), is licensed under a Creative Commons Attribution 3.0 Unported License (CC BY-SA). See <https://creativecommons.org/licenses/by-sa/3.0/> Distribution or reproduction of this work in whole or in part commercially or noncommercially requires full attribution of the original publication, including its digital object identifier (DOI). Derivatives of this work must carry the same license.