1DCSEMQWE:

An Open-Source 1D Forward Modeling Program for Controlled-Source Electromagnetic Method using Quadrature with Extrapolation technique

1. About the program

This manual proposes instructions for using 1DCSEMQWE code for 1D forward modeling. We also provide several Python examples for generating the input file and data postprocessing.

List of currently supported features:

- Cartesian coordinate system.
- Computes all electromagnetic fields' x, y, and z-components.
- Supports computation of both real and imaginary components of the fields.
- Transmitters and receivers can be located anywhere.
- Supports both electric and magnetic dipoles.
- Arbitrarily oriented transmitter.
- Isotropic resistivity.

1.1. License

1.2. Bugs, Fixes, Questions, and Comments

Send bug reports, fixes, questions, and comments to email: phamngockien@snu.ac.kr. We'll try our best to answer your questions, but don't expect an immediate reply.

2. Compiling and running the source code

For instructions on compiling and running on Unix and Linux machines, see the file "README" located in the /1D_CSEM_QWE folder.

3. Model parameters and transmitter-receiver configuration

3.1. Model parameters

We use a right-handed Cartesian geometry with the z-axis pointing down, as shown in Figure 1. All positions are in units of meters and angles are in degrees. N-layers are indexing from 0 to N-1. Each layer has a resistivity of ρ_i in units of Ohm-meter. The top depths of the layers are used to define the model geometry. Receivers and transmitters can be located anywhere in the model. Since the depths are free of choice, you will need to describe the parameters of the model in the "1D_CSEM_INPUT.txt" file.

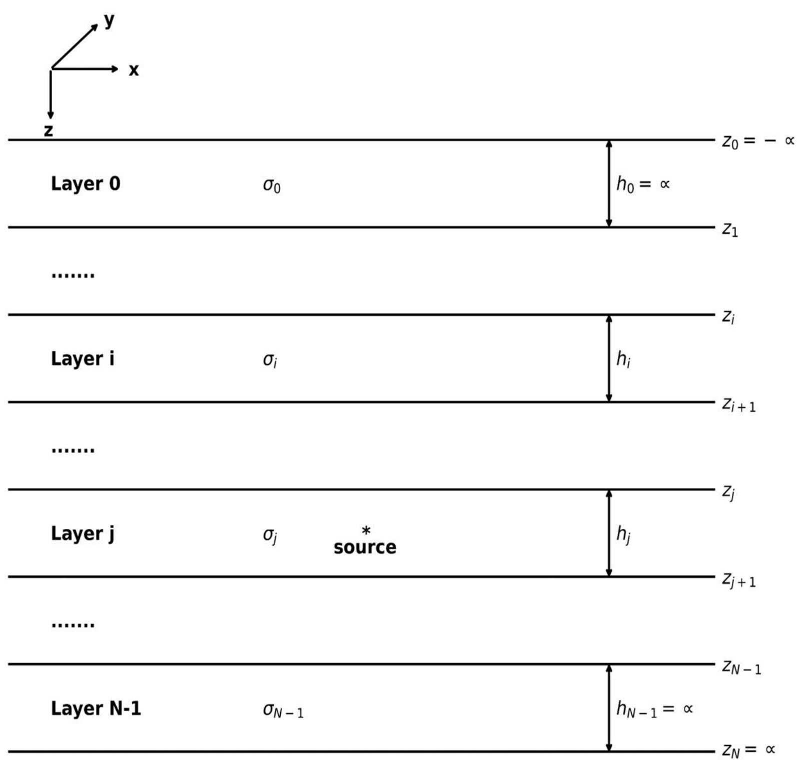

Figure 1. N-horizontally layered model (modified from Key, 2009). Each layer has a conductivity of σ_i and a thickness of h_i. The source is located in the j^{th} layer and the receiver is in the i^{th} layer. The dipole source and receiver can be placed anywhere in the model.

3.2. Transmitter-receiver configuration

The EM field of an arbitrarily oriented dipole source is the superposition of the fields excited by its horizontal and vertical components, as shown in Figure 2a. In 1DCSEMQWE, the azimuth of the transmitter is defined as its horizontal rotation from the x-axis, which is positive toward the y-axis (Figure 2b). Thus, an azimuth of 0 degrees means the transmitter points in the x-direction, whereas an angle of 90 degrees means the transmitter is y-oriented. The transmitter dip is positive in the downward direction, where a dip of 0 degrees means the transmitter only has a horizontal component.

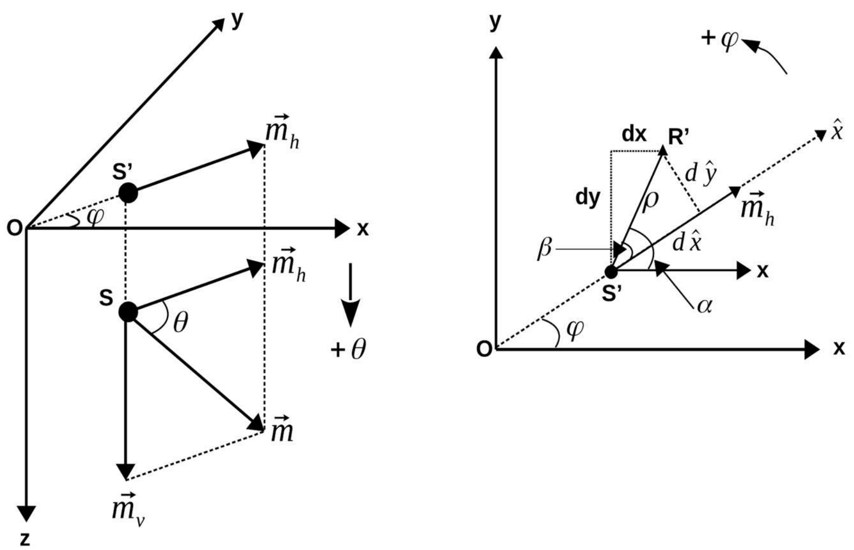

Figure 2. (a) An arbitrary oriented dipole source with moment $\vec{m}$ that can be decomposed into horizontal moment $\overrightarrow{m_h}$ and vertical moment $\overrightarrow{m_v}$. (b) various rotation parameters when computing the EM fields excited by a horizontal dipole that orients in the $\overrightarrow{O\hat{x}}$ direction. S is the position of the dipole source. S' and R' are the projections of the source and the receiver's positions onto the (Oxy) plane. The azimuth (φ) is the horizontal angle of the dipole source from the x-axis. $\varphi = 0$ for x-oriented dipole, and the positive direction of φ is counter clock-wise. The dip angle (θ) of the dipole is positive downward. dx and dy are the horizontal differences in x- and y-direction between the receiver and the source, respectively. d$\hat{x}$ and d$\hat{y}$ are those when we compute the EM fields of the dipole source in the rotated coordinates ($O\hat{x}\hat{y}$). ρ is the horizontal distance of the receiver from the source. α is the angle between $\overrightarrow{S'x}$ and $\overrightarrow{S'R'}$. β is that between $\overrightarrow{S'\hat{x}}$ and $\overrightarrow{S'R'}$.

4. Input file

1DCSEMQWE requires an input file named "1D_CSEM_INPUT.txt". This file should be copied into the /src folder of the program before carrying out the modeling. The parameters in this file are defined as follows.

+ Dipole type: a number that defines the type of the dipole source. Data type: integer. Value 0 is for electric dipole, and 1 is for magnetic dipole. Other values will return an error message.

+ NQUAD: number of Gauss quadrature points for evaluating the integrals. Data type: integer. Value: any positive integer.

+ RTOL: relative tolerance for the Quadrature With Extrapolation technique. Data type: double.Value: any positive value. It is noted that you should input a value that is smaller than 1 to control the accuracy of the integral evaluation.

+ ATOL: absolute tolerance for the Quadrature With Extrapolation technique. Data type: double.Value: any positive value. It is noted that you should input a value that is smaller than 1 to control the accuracy of the integral evaluation.

+ TRANSMITTERS: number of transmitters. Data type: integer. Value: any positive integer. NOTE: below this parameter, you have to describe the transmitters' parameters, including position (x, y, z), moment, azimuth, and dip.

+ FREQUENCIES: number of observation frequencies. Data type: integer. Value: any positive integer.
NOTE: below this parameter, you have to define the frequencies' values in the unit of Hz.

+ LAYERS: number of layers in the model. Data type: integer. Value: any positive integer.
NOTE: below this parameter, you have to describe the top depths (meters) and each layer's resistivities (Ohm-meters).

+ RECEIVERS: number of receivers. Data type: integer. Value: any positive integer.
NOTE: below this parameter, you have to describe the receivers' positions (x, y, z).

```
# DIPOLE TYPE: 0  ! Types: 0 for electric dipole source, 1 for magnetic dipole source.
# NQUAD: 101   ! Number of quadrature points to evaluate the integral.
# RTOL: 1e-12   ! Relative tolerance for convergence criteria of the QWE technique.
# ATOL: 1e-30  ! Absolute tolerance for convergence criteria of the QWE technique.
! This section lists the number of transmitters and then their parameters.
! The azimuth ranges from [0, 360] with a value of 0 in x-direction
! and has positive value w.r.t counter clockwise direction.
! The dip ranges from [-90, 90] degrees where -90 is upward direction and vise versa.
! X      Y        Z         Moment      Azimuth  Dip
# TRANSMITTERS: 11
 0       0        950       1           90        0
 0     1000       950       1           90        0
 0     2000       950       1           90        0
 0     3000       950       1           90        0
 0     4000       950       1           90        0
 0     5000       950       1           90        0
 0     6000       950       1           90        0
 0     7000       950       1           90        0
 0     8000       950       1           90        0
 0     9000       950       1           90        0
 0    10000       950       1           90        0
! This section lists the number of frequencies, and then their values.
# FREQUENCIES:    2
 0.25
 1.0
! This section gives the number of layers, their top depths, and resistivities (Ohmm).
# LAYERS:       5
-100000 1e12
 0    .30
1000    1.0
2000    100
2100    1.0
! This section lists the number of receivers, and then their positions in x, y, z coordinates.
# RECEIVERS:     1
 0       0      1000
```

Figure 3. An example of an input file to compute the electromagnetic field of the canonical model.

An example of the input file is shown in Figure 3. The symbol "!" is used before every comment in the file; hence, the comments are not loaded into the program's main functions. It is noted that there is no inline comment at the lines defining transmitters' parameters, frequency values, layers' parameters, and receivers' positions. If you write an inline comment on those lines, the program may return an error message.

5. Output files

1DCSEMQWE returns the following data file in Comma Separated Value (CSV) format.
+ TX_RX_configurations.csv file contains the transmitters' parameters and the positions of the receivers. An example of the configuration file is illustrated in Figure 4.

source_type	TX_x	TX_y	TX_z	moment	azimuth	dip	RX_x	RX_y	RX_z
Magnetic_Dipole_Source	0	0	2050	1	0	90	-3000	0	-1000
Magnetic_Dipole_Source	0	0	2050	1	0	90	-3000	0	-900
Magnetic_Dipole_Source	0	0	2050	1	0	90	-3000	0	-800
Magnetic_Dipole_Source	0	0	2050	1	0	90	-3000	0	-700

Figure 4. An example of the TX_RX_configurations.csv file.

+ model_parameters.csv file includes the 1D model described in the input file. The data structure of this file is shown in Figure 5. Here, the top of the first layer is set to negative infinity.

Top_depth(m)	resistivity(ohm-m)
-1.79769313486232E+308	1000000000000
0	0.3
1000	1
2000	100
2100	1

Figure 5. An example of the model_parameters.csv file.

+ n files of EM responses for the given transmitter-receiver configuration at each frequency. Here, n represents the number of frequencies used in the modeling. Each file contains data that is structured within 19 columns. The first column stores the frequency value in the unit of Hz. Columns 2 and 3 are for the real and imaginary components of the electric field in the x-direction (Ex); columns 4 to 7 are those of Ey and Ez, respectively. The rest columns contain the data for magnetic field (H) and magnetic flux density in the same order as the data for the electric field.

6. Examples

We provide several examples as a guideline for using the 1DCSEMQWE program. To reproduce the result, you should copy the "1D_CSEM_INPUT.txt" file into the /src folder and run the program. Copies of the output files are also added to the related example in the /doc/examples folder.

Although it is straightforward to create the "1D_CSEM_INPUT.txt" file by hand, we include a python code that illustrates how to generate this file automatically in the 1D_CSEM_QWE/doc/examples/Generating_input_file folder. Thus, it would be easier to generate an input file with a large number of transmitters and receivers.

Other examples for electric dipole and magnetic dipole sources are also included in the 1D_CSEM_QWE/doc/examples folder. In each example, there is a python code for generating the input file, another code is to implement data postprocessing which produces a figure in pdf format.

In this version of 1DCSEMQWE, we use the following version of python libraries to carry out the input file generation and data postprocessing:

- Python version 3.9.6
- NumPy version 1.19.5
- Pandas version 1.3.0

Matplotlib version 3.3.4