



## 1D Magnetotelluric Forward Modelling Web App

**Andrew Pethick**

Curtin University  
Level 4, 26 Dick Perry Ave, Kensington  
Andrew.Pethick@curtin.edu.au

**Brett Harris**

Curtin University  
Level 4, 26 Dick Perry Ave, Kensington  
B.Harris@curtin.edu.au

### SUMMARY

The practice of geophysical computation on solely personal computers is fast becoming outdated in this modern web-app era. The instantaneous, on demand nature of modern life is now firmly established commercially. We present one approach for developing an integrated 1D magnetotelluric forward modelling web application. This basic web application combines a scientific python back end and a front end built upon PHP and HTML5 web technologies. It has also been packaged as easy to install plugin for the popular Wordpress framework. The application simulates the 1D magnetotelluric response over any isotropic geo-electrical earth. With our WebMT application, MT forward modelling can be performed on any internet enabled device containing a HTML5 compliant browser. This includes mobile phones, tablets and desktop PC's. This research demonstrates one approach to geophysical web application development and encourages future development and innovation within the wider community.

**Key words:** Web Application, Magnetotelluric, Forward Modelling, Software, EM, PHP, Python

### INTRODUCTION

Interactions between the users and geophysical software have to this day have been typically performed through personal computers and laptops. This approach is arguably the most efficient at handling geophysical data, but other approaches may have their place in the geophysical community at large. The world of enterprise software is beginning to transition from client software towards web hosted applications due to piracy, explosion of desktop and mobile operating systems, increasing public online presence, compute power requirements and the blurring of computational genres of televisions, mobile phones, desktop computers and laptops (Rosenbush 2014).

Web applications are not new to geophysics. There have been several successful web applications developed for various purposes including seismic and electromagnetic forward modelling and inversion. For example Modelr (Hall and Bianco 2014) is an open source seismic web application which rapidly generates a number of AVO and AVA plots and synthetically generated seismic cross sections from any given

2D starting impedance model. The Virtual Geophysics Laboratory (VGL) (Fraser et al. 2013) has developed a scientific workflow portal providing Geophysicists with access to an integrated working environment exploiting eResearch tools and Cloud computing technology. The online VGL web application is a collaboration between CSIRO, Geoscience Australia (GA), and the National Computational Infrastructure (NCI). The portal provides access to UBC EM and potential field inversion routines. Web applications are an innovative approach to developing easily accessible geophysical routines. The following paper presents the steps and open source code to build a basic geophysical forward modelling web application from the ground up. Our software named WebMT is open source licensed under the WTFPL license and is freely downloadable and modifiable from Bitbucket.com (See Pethick (2014a)).

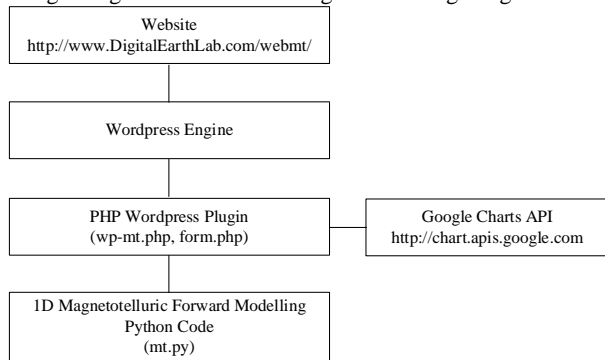
### METHOD AND RESULTS

The steps undertaken to develop the 1D magnetotelluric (MT) forward modelling web application, albeit not exhaustive, consists of,

1. Develop the scientific 1D MT forward modelling routines using Python
2. Create PHP code to run a python script hosted on a web server
3. Create PHP code to format the HTML web form input into an input string for the Python script
4. Create the web interface using HTML which allows users to customise forward modelling input variables
5. Write PHP code to retrieve and format the Python script
6. Write PHP code to export the formatted Python output to a HTML table and with the Google Charts API and generate output apparent resistivity and phase versus frequency curves as a PNG image
7. Insert Wordpress initialisation hooks and header information into the master PHP file and upload all files onto a hosted Wordpress site as a plugin.

The structure of the web app can be broken up into two major developmental components; the 1D magnetotelluric modelling python code and the PHP Wordpress plugin (See Figure 1). The structure of WebMT use the concept of a "Mashable Web" or "Programmable Web" principles. This means web applications can be built upon third party online API's. Our PHP Wordpress plugin calls the Google Charts API to convert the forward modelled data as graphs. That is, the charts are

generated by calling upon the Google Charts API web server using a single URL and retrieving the resulting image.



**Figure 1: The hierarchy of the WebMT the 1D forward modelling MT web application.**

The first step requires developing the 1D magnetotelluric code. The derivation of the code has been modified from several sources including Niwas, Gupta, and Gaur (2005) and Pedersen and Hermance (1986). The steps to reproduce the mathematical procedure as 1D MT forward modelling Python code have been presented as an online tutorial by Pethick (2014b). The code is surprisingly short, less than 50 lines of code and more importantly computes multi-frequency apparent resistivity and phase curves in less than 10 milliseconds on a single core using three arguments specifying frequencies, resistivities and thicknesses. The output is a string containing all the modelled apparent resistivity and phase values with the corresponding frequency.

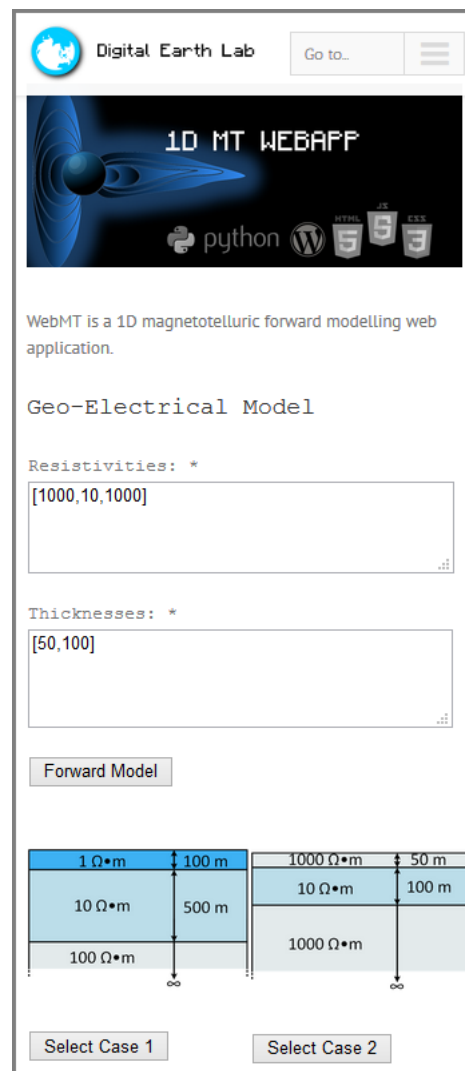
The PHP code is split into two parts the main controller (wp-mt.php) and the form input (form.php). The controller PHP file must enable the execution of the Python script with the appropriate geo-electrical model on the web server. A function was created to take in form input, run the script and export the appropriate data. The execution of the forward modelling script is performed using the “shell\_exec” PHP command which calls python on the server side. This means all computation is carried out on the web server and not on the user’s personal PC. To avoid unnecessarily writing the output data to the secondary storage (i.e., the server hard drive) the output from the Python script must somehow be piped to the PHP controller file. Unfortunately there is no native support for piping Python primitives other than by string so the PHP controller must format the piped output string from the Python script to three arrays of floating point numbers (i.e., arrays of frequency, apparent resistivity and phase datums). These PHP floating point arrays are then used to (i) generate Google chart images by formatting and passing the arrays into the google charts API and (ii) display the raw text output as a HTML table.

The PHP form file generates the input user interface, but unlike the control file, contains no business logic. The interface is composed of four user input controls the (i) layer resistivity input textbox, (ii) layer thickness input textbox, (iii) forward modelling button and (iv) pre-initialised geo-electrical case model selection buttons. The form is called within the master PHP control file initialise function via the PHP ‘include’ command.

Once all these components are developed the PHP file is ready to be inserted into a website, however we decided to test the

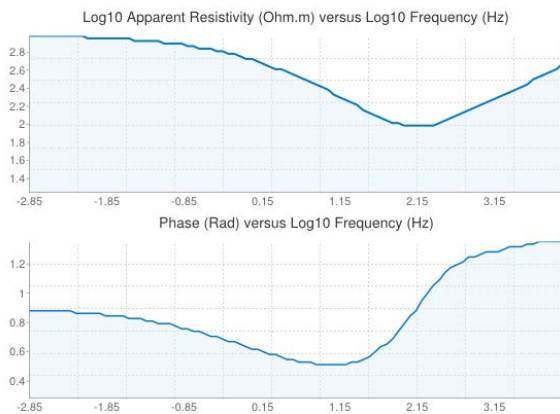
application within the Wordpress common website and blogging framework. Converting the PHP code into a Wordpress plugin required minor modifications within the control file. This included a text header containing software meta information and secondly initialising the shortcode hooks enabling wordpress to recognise the initialise function as the main PHP action. The final steps were to copy the development folder containing all the PHP, Python and case study images into the Wordpress plugin directory, enable the plugin via the Wordpress dashboard and call the shortcode within a page or post. The resulting web application can be seen in Figure 2 while the graphical display of the simulated data is shown in both the generated Google charts (Figure 3) and table output (Figure 4).

**FIGURES**



**Figure 2: A screenshot of the WebMT interface. The interface is integrated into a responsive Wordpress theme enabling automated layout reformatting for mobile, tablet and desktop devices.**

## Graphs



**Figure 3:** Example of synthetically generated magnetotelluric data represented within WebMT as a google chart. The chart is dynamically generated after executing the Python 1D MT forward code on the web server by sending a single URL containing data and plotting parameters to the Google Charts API.

## Raw Output Data

Frequency	A. Res.	Phase (Rad)
0.0014	954.53396685148	0.76267861562498
0.0017	950.02029370946	0.76042370068412
0.002	945.90983869471	0.75837016494382
0.0023	942.11448178769	0.75647397021704
0.0028	936.33386935348	0.75358572015112
0.0034	930.08549114951	0.75046338421941
0.004	924.40712736564	0.7476254600411
0.0046	919.17415876417	0.74500969881971
0.0055	911.980858746	0.74141323384696
0.0067	903.33415241072	0.73708863795497
0.0079	895.51108469001	0.73317436273369
0.0092	887.75545370713	0.72929208352803
0.011	877.97985027586	0.72439583712054
0.0134	866.27839429502	0.71853028780477

**Figure 4:** Small sub-sample of the output table data. The output string retrieved after running the Python code is parsed into three corresponding frequency, apparent resistivity and phase arrays and finally converted into a standard HTML table.

## CONCLUSIONS

The WebMT web application allows any individual with a smartphone, tablet or desktop PC to quickly model a synthetic 1D MT response over an isotropic layered earth. The application was developed using a range of web based and scientific technologies including PHP, HTML, Wordpress, Google API's and Python scripts. The application is currently hosted on a standard web server and is freely available to the public on the URL [www.DigitalEarthLab.com/WebMT](http://www.DigitalEarthLab.com/WebMT). The code is open source, including the 1D MT forward modelling Python and associated PHP code and has unlimited commercial, educational and personal use. This research undertaken hopefully provides a clear pathway for the creation of further and more complex geophysical web applications.

## APPENDIX: MT Modelling Code (mt.py)

```
import math
import cmath
import time
import sys
freqString = sys.argv[1][1:-1];
resString = sys.argv[2][1:-1];
thickString = sys.argv[3][1:-1];
resistivities = map(float,resString.split(','));
thicknesses = map(float,thickString.split(','));
frequencies = map(float,freqString.split(','));
freqOut = [];
aresOut = [];
phaseOut = [];
output = [freqOut,aresOut,phaseOut];
def fwd(frequency, resistivities, thicknesses):
    mu = 4*math.pi*1E-7;
    n = len(resistivities);
    w = 2*math.pi*frequency;
    impedances = list(range(n));
    impedances[n-1] =
cmath.sqrt(w*mu*resistivities[n-1]*1j);
    for j in range(n-2,-1,-1):
        resistivity = resistivities[j];
        thickness = thicknesses[j];
        dj=cmath.sqrt((w*mu *(1.0/resistivity))*1j);
        wj = dj * resistivity;
        ej = cmath.exp(-2*thickness*dj);
        belowZ = impedances[j + 1];
        rj = (wj - belowZ)/(wj + belowZ);
        re = rj*ej;
        Zj = wj * ((1 - re)/(1 + re));
        impedances[j] = Zj;
    Z = impedances[0];
    absZ = abs(Z);
    aRes = (absZ * absZ)/(mu * w);
    phase = math.atan2(Z.imag, Z.real);
    freqOut.append(math.log10(frequency));
    aresOut.append(math.log10(aRes));
    phaseOut.append(phase);
    return [frequency, aRes, phase];
for frequency in frequencies:
    fwd(frequency,resistivities,thicknesses);
print output;
```

## REFERENCES

- Fraser, Ryan, Josh Vote, Richard Goh, and Simon Cox. 2013. "Virtual Geophysics Laboratory: Exploiting the Cloud and Empowering Geophysicists" EGU General Assembly Conference Abstracts.
- Hall, Matt, and Evan Bianco. 2014. Modelr. Accessed 29/08/2014, <https://www.modelr.io/>.
- Niwas, Sri, PK Gupta, and VK Gaur. 2005. "Normalized Impedance Function and the Straightforward Inversion Scheme for Magnetotelluric Data." *Journal of earth system science* 114 (5): 523-531.
- Pedersen, Jens, and John F Hermance. 1986. "Least Squares Inversion of One-Dimensional Magnetotelluric Data: An Assessment of Procedures Employed by Brown University." *Surveys in Geophysics* 8 (2): 187-231.

Pethick, Andrew. 2014a. Digital Earth Lab Bitbucket Account. 29/08/2014 Accessed 29/08/2014, <https://bitbucket.org/apethick>.

Pethick, A. 2014b. Tutorial - 1d Forward Modelling (Magnetotelluric). Digital Earth Lab

Accessed 29/08/2014, <http://www.digitalearthlab.com/tutorial/tutorial-1d-mt-forward/>.

Rosenbush, Steve. 2014. The Morning Download: Activation of Corporate Mobile Apps Is Soaring. Accessed 12/08/2014, <http://blogs.wsj.com/cio/2014/08/12/the-morning-download-activation-of-corporate-mobile-apps-is-soaring/>.