Practical integration of processing, inversion and visualization of magnetotelluric geophysical data

Gudni Rosenkjaer, Lindsey Heagy, Rowan Cockett, Seogi Kang, Doug Oldenburg

Short (400 char):

Having well contained software to perform essential tasks is beneficial in our geophysical workflow. We'll discuss the ongoing implementation of geophysical software for the Magnetotelluric problem that builds on the SimPEG framework (Simulation and Parameter Estimation in Geophysics), and show how we implement our code and workflows to deal with the overhead of processing data, running physical simulations, completing inversions, and visualizing the results.

Long (300 Words):

Geophysical inversions can produce insightful images of the subsurface, however, there is a significant amount of effort involved in pre-processing data as well as visualizing the inversion outputs. In the case magnetotelluric data, a multitude of file formats, data unit conversions, and coordinate systems need to be addressed before the inversion process can be started. Specific software solutions exist but require the user to use multiple packages that can be a headache to integrate and often need commercial licences. Our aim is to build an open-source software tool that is flexible, customizable, extensible and capable of dealing with all of our tasks in the same environment. Essential abilities are to be able to do the work both in a visual and programmable fashion, in order to have the interactive capabilities to interrogate and modify the logic/methodologies in the workflow. We set our work up as scripts and plugins that are easy to understand and can be reused, shared or adapted.

We have extended the SimPEG framework (http://simpeg.xyz) for application to the Magnetotelluric problem. We will show our programmatic workflow to progress from reading and extracting information from standardized EDI files, running a geophysical inversion, and show how we can integrate other spatial data, geological information and produce 3D visualizations of the results. We will highlight some of our implementations throughout this process, and how we utilize different Python packages such as GDAL (https://pypi.python.org/pypi/GDAL), numpy (http://www.numpy.org/) and VTK (www.vtk.org) to complete our work.