The MTPy software package for magnetotelluric data analysis and visualisation

Alison Louise Kirkby[1], **Fei Zhang**[1], **Jared Peacock**[2], **Rakib Hassan**[1], and **Jingming Duan**[1]

1 Geoscience Australia **2** United States Geological Survey

Introduction

The magnetotelluric (MT) method is increasingly being applied to a wide variety of geoscience problems. However, the software available for MT data analysis and interpretation is still very limited in comparison to many of the more mature geophysical methods such as the gravity, magnetic or seismic reflection methods.

MTPy is an open source Python package to assist with MT data processing, analysis, modelling, visualization and interpretation. It was initiated at the University of Adelaide in 2013 as a means to store and share Python code amongst the MT community (Krieger & Peacock, 2014). Here we provide an overview of the software and describe recent developments to MTPy. These include new functionality and a clean up and standardisation of the source code, as well as the addition of an integrated testing suite, documentation, and examples in order to facilitate the use of MT in the wider geophysics community.

Key Functionality

MTPy contains modules to carry out much of the processing and analysis that can be applied to MT data. MTPy works primarily with the industry standard Electronic Data Interchange (EDI) file format however it is compatible with other formats including the Jfile format exported by the Bounded Influence, Remote Reference Processing (BIRRP) code (Chave & Thomson, 2004; Chave, Thomson, & Ander, 1987), and xml. The MTPy structure is based on key steps in working with MT data, namely processing, analysis, modelling, and imaging. These sub-packages are based on modules within the core sub-package (Figure 1, Krieger & Peacock, 2014).

The core sub-package contains functionality to read and write MT data from industry standard formats such as EDI. When data are read in, an MT object is created. The MT object contains site metadata from the header of the EDI file, including the location. It also contains the impedance tensor (Z) and the vertical transfer functions (Tipper).

The processing sub-package is designed to facilitate working with time series data and generating inputs for existing third-party processing codes, for example the BIRRP code (Chave & Thomson, 2004; Chave et al., 1987).

The analysis sub-package includes functions that are commonly used to analyse MT data prior to modelling and inversion. These include: dimensionality analysis, strike angle calculation, phase tensor analysis (T. G. Caldwell, Bibby, & Brown, 2004), and calculation and removal of distortion (H. M. Bibby, Caldwell, & Brown, 2005). In addition, there
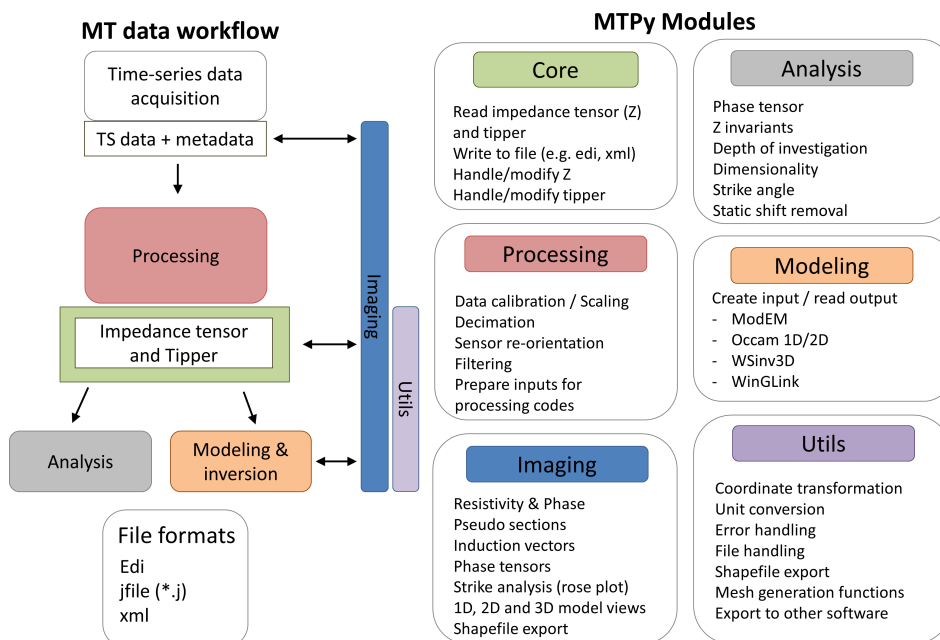
Figure 1: MTPy package structure showing the key modules that are under active development. The colours in the workflow diagram represent which parts of MTPy are used at each step. For example, modules in the analysis sub-package are used on the impedance tensor data, while visualisation is carried out on time series data, processed data, and the results of analysis and modelling or inversion. This diagram is modified after Krieger & Peacock (2014).
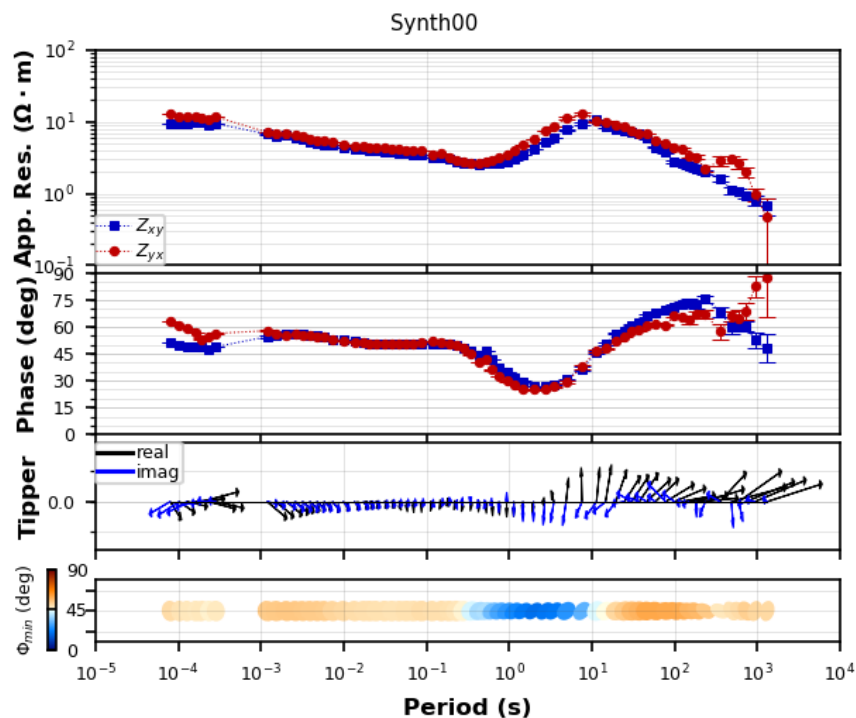
Figure 2: Example of a single station plot generated using the plotresponse module within the imaging sub-package in MTPy. The plot shows Z_{XY} and Z_{YX} resistivity and phase, induction vectors (Tipper; Parkinson, 1962), and phase tensors (T. G. Caldwell et al., 2004) as a function of period.

are functions to calculate and visualize depth of investigation using the Niblett-Bostick transform (Bostick, 1977; Jones, 1983; Niblett & Sayn-Wittgenstein, 1960).

MTPy also contains the modeling sub-package, which allows users to create inputs to, and visualize outputs from, several of the commonly-used MT inversion codes. These include the Occam 1D and 2D inversion codes (S. C. Constable, Parker, & Constable, 1987; DeGroot-Hedlin & Constable, 1990), and the ModEM 3D inversion code (Egbert & Kelbert, 2012; Kelbert, Meqbel, Egbert, & Tandon, 2014).

The imaging sub-package contains functionality to visualise MT data, as well as many of the outputs from the analysis and modelling modules. These include simple plots of resistivity and phase at a single station (e.g. Figure 2), resistivity and phase pseudosections and maps, phase tensor sections and maps (e.g. Figure 3), penetration depth and geoelectric strike plots, and images of resistivity models and the model responses.

The functions in utils underpin these modules, by handling basic underlying functionality such as coordinate transformations, common filehandling operations and calculations that are commonly applied to MT data, such as conversion from impedance tensor (Z) to apparent resistivity and phase.

Recent development

In order to make MTPy a comprehensive toolkit that can be easily installed and used by the wider geophysics community, we are applying software engineering best practices and techniques to improve software quality and usability. New developments include incorporation of a unit-testing suite, continuous build, automatic documentation generation
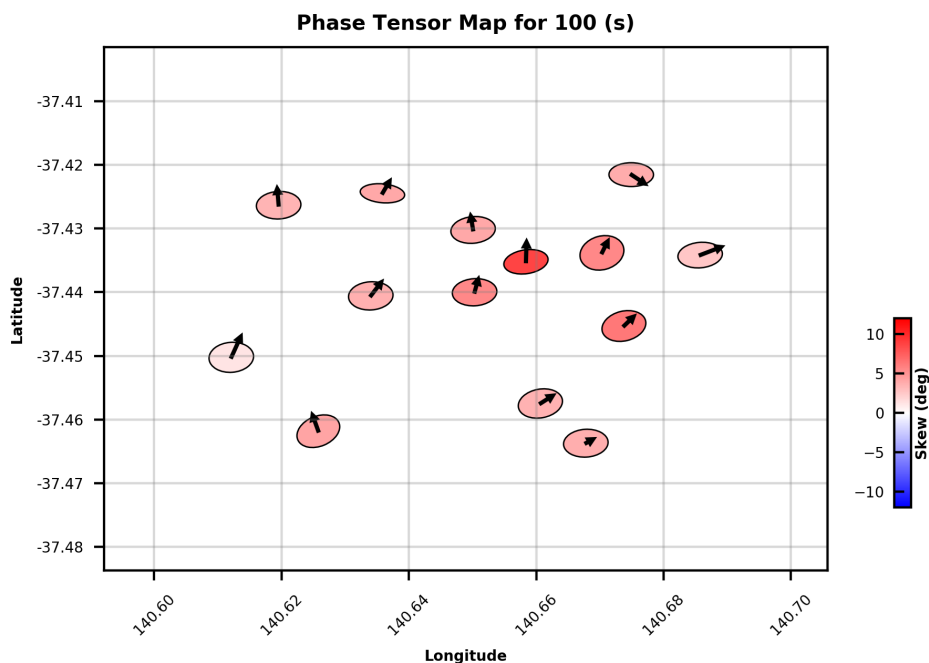
Phase Tensor Map for 100 (s)

Figure 3: Example of a phase tensor and induction vector map created by the phase_tensor_maps module in the imaging sub-package in MTPy. Phase tensor ellipses are calculated using the method detailed by T. G. Caldwell et al. (2004) and coloured by skew angle in degree.

(http://mtpy2.readthedocs.io/en/develop/) and addition of installation guide wiki pages (https://github.com/MTgeophysics/mtpy/wiki).

Recent functionality development to MTPy includes significant expansion of the modules that generate inputs to, and allow visualiation of outputs from, the ModEM 3D inversion code (Egbert & Kelbert, 2012; Kelbert et al., 2014). This includes new functionality to extract vertical slices from 3D models for display on profiles. Both of these support overlay of other datasets in both profile and plan view. These can be used with the seismic module in imaging for plotting 2D seismic reflection data (in profile) and the geology module for plotting shapefiles. We have now also implemented functions to export ModEM 3D models to GOCAD$^{\mathrm{TM}}$ sgrid format, and to export depth slices to a column based text format containing x, y, z resistivity.

The analysis and imaging sub-packages have been expanded through the addition of modules that calculate and visualise depth of investigation of MT measurements, for a single site, a profile, or a grid of sites.

We have also added an edi_collection module that works specifically with collections of edi files and enables phase tensors from a collection of stations to be exported to csv or ArcView shapefiles.

Projections and transformations are now handled using the open-source Pyproj library, or if users prefer, the more comprehensive GDAL library.

Finally, we have compiled a set of jupyter notebooks, example scripts, and workshop material within the examples folder in the repository, which demonstrate the key functionality contained in MTPy.

Availability

The software is distributed under a GNU General Public License v3.0 and is available from https://github.com/MTgeophysics/mtpy.

Acknowledgements

References

Bibby, H. M., Caldwell, T. G., & Brown, C. (2005). Determinable and non-determinable parameters of galvanic distortion in magnetotellurics. *Geophysical Journal International*, *163*(3), 915–930. doi:10.1111/j.1365-246x.2005.02779.x

Bostick, F. (1977). A simple almost exact method of MT analysis. In *Workshop on electrical methods in geothermal exploration, u.s. Geological survey, contract no 14080001-8-359*. Retrieved from http://www.complete-mt-solutions.com/mtnet/papers/ClassicPapers/Bostick_1977.pdf

Caldwell, T. G., Bibby, H. M., & Brown, C. (2004). The magnetotelluric phase tensor. *Geophysical Journal International*. doi:10.1111/j.1365-246X.2004.02281.x

Chave, A. D., & Thomson, D. J. (2004). Bounded influence magnetotelluric response function estimation. *Geophysical Journal International*, *157*(3), 988–1006. doi:10.1111/j.1365-246X.2004.02203.x

Chave, A. D., Thomson, D. J., & Ander, M. E. (1987). On the robust estimation of power spectra, coherences, and transfer functions. *Journal of Geophysical Research*, *92*(B1), 633. doi:10.1029/JB092iB01p00633

Constable, S. C., Parker, R. L., & Constable, C. G. (1987). Occam's inversion —- A practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics*, *52*(3), 289–300. Retrieved from http://marineemlab.ucsd.edu/Projects/Occam/1DCSEM/

DeGroot-Hedlin, C., & Constable, S. (1990). Occam's inversion to generate smooth, two-dimensional models from magnetotelluric data. *GEOPHYSICS*, *55*(12), 1613–1624. doi:10.1190/1.1442813

Egbert, G. D., & Kelbert, A. (2012). Computational recipes for electromagnetic inverse problems. *Geophysical Journal International*. doi:10.1111/j.1365-246X.2011.05347.x

Jones, A. G. (1983). On the Equivalence of the "Niblett" and "Bostick" Transformations in the Magnetotelluric Method. *Journal of Geophysics*, *53*, 72–73.

Kelbert, A., Meqbel, N., Egbert, G. D., & Tandon, K. (2014). ModEM: A modular system for inversion of electromagnetic geophysical data. *Computers and Geosciences*. doi:10.1016/j.cageo.2014.01.010

Krieger, L., & Peacock, J. R. (2014). MTpy: A Python toolbox for magnetotellurics. *Computers & Geosciences*, *72*, 167–175. doi:10.1016/J.CAGEO.2014.07.013

Niblett, E. R., & Sayn-Wittgenstein, C. (1960). Variation of electrical conductivity with depth by the magneto-telluric method. *Geophysics*, *25*(5), 998–1008. doi:10.1190/1.1438799

Parkinson, W. D. (1962). The Influence of Continents and Oceans on Geomagnetic Variations. *Geophysical Journal International*, *6*(4), 441–449. doi:10.1111/j.1365-246X.1962.tb02992.x