Development of the MTPy software package for magnetotelluric data analysis

A. Kirkby[1], J. Peacock[2], F. Zhang[3], R. Hassan[4] and J. Duan[5]
[1]Geoscience Australia, Alison.Kirkby@ga.gov.au
[2]United States Geological Survey, jpeacock@usgs.gov
[3]Geoscience Australia, Fei.Zhang@ga.gov.au
[4]Geoscience Australia, Rakib.Hassan@ga.gov.au
[4]Geoscience Australia, Jingming.Duan@ga.gov.au

SUMMARY

The MTPy Python library is open source software that aims to facilitate processing, analysis, modelling, and inversion of magnetotelluric (MT) data. Until recently, MTpy has contained bugs and gaps in both functionality and documentation, which have limited its use to date. We are developing MTPy to rectify these problems and expand functionality. Key improvements include adding new functions and classes to the modules that handle ModEM inputs and outputs, improving data and model visualization tools, refactoring the code to improve maintainability, quality, and consistency, and developing documentation.

Keywords: magnetotellurics, python, software, analysis, processing

INTRODUCTION

The magnetotelluric (MT) method is increasingly being applied to a wide variety of geoscience problems, as inversion tools and compute facilities become more sophisticated. However, the software available for MT data analysis and interpretation is still very limited in comparison to other geophysical methods such as gravity, magnetics, and seismic.

MTpy is an open source software package that has been developed 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 and Peacock, 2014).

Written in the Python programming language, MTpy contains modules to carry out much of the processing and analysis that needs to be applied to MT data, working with the industry standard EDI file format amongst other formats. MTPy also contains modules to create inputs to, and visualize outputs from the Occam 1D and 2D (Constable et al., 1987, deGroot-Hedlin and Constable, 1990), and ModEM 3D (Egbert and Kelbert, 2012, Kelbert et al., 2014) inversion codes. However, until recently, much of the code has lacked internal unit tests, and contained bugs and gaps in functionality. These factors have limited its use in the wider MT community to date.

Geoscience Australia in collaboration with developers at the University of Adelaide and the United States Geological Survey continue to develop MTpy to improve its functionality and provide increased user support. The aim is to make the software toolkit easier to use and thus facilitate the use of MT in the wider geophysics community.

FUNCTIONALITY DEVELOPMENT

MTpy is structured around key steps in working with MT data, namely processing, analysis, modelling, and imaging. These sub-packages are based upon the core modules, which deal with reading and writing impedance tensor data from industry standard formats such as EDI (Figure 1, Krieger and Peacock, 2014).

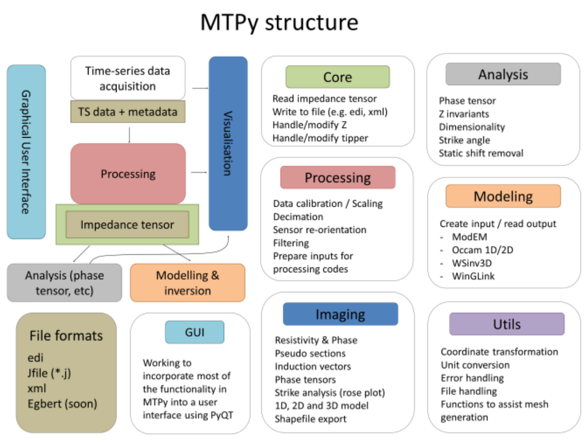

Figure 1. MTPy package structure, modified after Krieger and Peacock (2014).

The processing modules are designed to facilitate working with time series data and generating inputs for existing processing codes, for example, the processing code BIRRP (Chave et al 1987, Chave and Thomson, 2004).

The analysis modules include phase tensor and dimensionality analysis, calculation of impedance tensor invariants, and calculation and removal of distortion. Recent developments include addition of functions to calculate and visualize depth of investigation using the Niblett-Bostick transform (Jones, 1983).

A number of changes have been added to the modelling modules in MTPy. These modules cover creation of inputs and analysis of outputs for many of the widely used modelling codes used in the MT community. Improvements to the modelling modules over the last 12 months have focused mainly on creating inputs for and viewing outputs from the ModEM code (Egbert and Kelbert, 2012, Kelbert et al, 2014).

Recent developments to the ModEM modules in MTpy include the addition of modules to allow topography and bathymetry to be added to models, and to allow reading and writing of models to and from GOCAD™ sgrid format. GOCAD™ is a software package developed by Paradigm for 3D spatial analysis and geological modelling, and will allow for more detailed analysis of resistivity models produced by ModEM. We have also made developments to the handling of projections in ModEM, to facilitate the creation of large models that cross UTM zones. Finally, updates have been made to the visualization modules, allowing visualization of input data and modelled responses as plots showing resistivity and phase as a function of period, and as phase tensor maps. In addition, the modules allow plotting of resistivity models as depth slices and cross sections.

As well as adding to the functionality contained in MTpy, we are working to apply software engineering best practices and techniques to improve software quality and usability. In addition, a GUI is being developed (Figure 1) to assist with the use of all MTpy modules. Our aim is to make MTpy a comprehensive toolkit that can be easily installed and used by the wider geophysics community.

Conclusions

The MTpy Python library was created to assist with many aspects of MT data processing, analysis and modelling. Geoscience Australia in collaboration with developers at the University of Adelaide and the United States Geological Survey continues the development to improve functionality, remove duplication, and facilitate its use. Key developments include improvements to the processing modules, addition of functions to calculate and visualise penetration depth, and improvement to the modelling modules to assist with creation of input files, and visualisation and analysis of outputs. A GUI is also being developed to help with the easy use of the functions contained within MTpy. For more updated details please visit the open source code repository: https://github.com/MTgeophysics/mtpy.

References

Chave A.D., Thompson DJ and Ander, M.E. (1987) On the robust estimation of power spectra, coherences and transfer functions. Journal of Geophysical Research 92: 633–648.

Chave, A.D., and Thomson, D.J. (2004) Bounded influence magnetotelluric response function estimation. Geophysical Journal International 157: 988–1006.

Constable, S. C., R. L. Parker, and C. G. Constable, (1987) Occam's inversion — A practical algorithm for generating smooth models from electromagnetic sounding data, Geophysics, 52: 289–300.

deGroot-Hedlin, C., and S. Constable (1990) Occam's inversion to generate smooth two-dimensional models from magnetotelluric data, Geophysics, 55 (12), 1613–1624.

Egbert, G. D. and Kelbert, A. (2012) Computational recipes for electromagnetic inverse problems. Geophysical Journal International, 189 (1): 251-267.

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., and Tandon, K. (2014) ModEM: a modular system for inversion of electromagnetic geophysical data. Computers & Geosciences, 66: 40-53.

Krieger, L. and Peacock, J. (2014) MTPy: A Python toolbox for magnetotellurics. Computers and Geosciences, v. 72, pp. 167-175.