THE UNIVERSITY of ADELAIDE

SOUTH AUSTRALIAN CENTRE FOR GEOTHERMAL ENERGY RESEARCH

MTpy

a Python Toolbox for Magnetotellurics

software for MT data processing and analysis

Lars Krieger[1,2,4]
lars.krieger@adelaide.edu.au

Jared Peacock[3]
jpeacock@usgs.gov

Stephan Thiel[1,2]

Kent Inverarity[2]

Kate Robertson[2]

[1]South Australian Centre for Geothermal Energy Research
[2]School of Earth and Environmental Sciences – Geophysics & Geology

University of Adelaide,
Adelaide 5005, Australia

[3]USGS
Geology & Geophysics
Menlo Park
94025, CA (USA)

USGS science for a changing world

[4]NICTA – Australia's Information and Communications Technology Research Centre of Excellence

Acknowledgements:

This work is partly funded by the South Australian Center for Geothermal Energy Research and Australia's Information and Communications Technology Research Centre of Excellence (NICTA). NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

Data from the AFAR geothermal site were provided by Yohannes Lemma.

First presentation of this poster:
2013 AGU Fall Meeting (San Francisco, USA), 13.12.2013

Abstract

We have developed the software package *MTpy*, which allows the handling, processing, and imaging of magnetotelluric data sets. It is written in Python and the code is open-source. The setup of this package follows the modular approach of successful software packages like GMT or Obspy. It contains sub-packages and modules for supporting processing, handling, and analysis of MT data.

Within the geophysical techniques, magnetotellurics (MT) is a relatively immature method. It is still not as widely spread as other methods like seismology, and as a result, the file handling and processing software within the academic community is mainly based on a loose collection of codes, often highly adapted to the respective local specifications. Although tools for the estimation of the impedance tensor, as well as inversion and modelling codes, are generally available, the standards and software for handling MT data are not unified throughout the community. We aim to overcome problems that arise from this lack of standards, and to simplify the general handling of MT data with the help of MTpy.

Our goal is not to produce a static collection of software, we rather introduce MTpy as a flexible framework, which will be dynamically extended into a versatile supplement for existing algorithms in the future. We introduce the concept and structure of MTpy, and we illustrate the workflow of MT data processing utilising MTpy on example data sets from geothermal exploration sites in South Australia and Ethiopia.

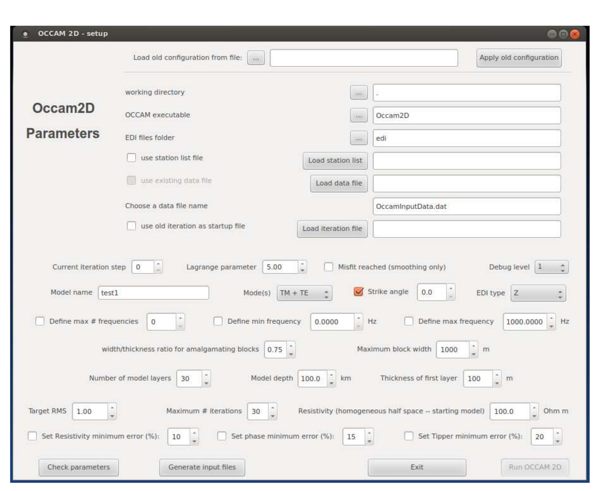
Implementation

- Python
- Platform independent
- Numpy, Scipy, Matplotlib
- Structured as sub-packages
- Modules and scripts
- Open source
- GPLv3 license
- Public GIT repository
- Freely accessible
- Simple installation with *Distutils*

Try it!

https://github.com/geophysics/mtpy

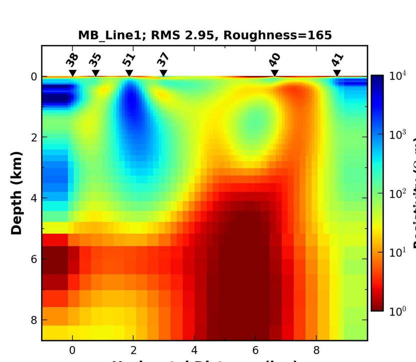
Modelling



Occam2D - generate input files

MTpy...

- ...does not model (yet)
- ...uses external software (e.g. OCCAM)
- ...generates input files
- ...validates setup parameters
- ...converts output files
- ...visualises models



Occam2D - visualise output

Current status

Occam2D determine parameters, generate input, visualise output

WS3Dinv generate input, convert output (vtk)

WinGLink read models and station files

ModEM convert output (vtk)

References

MTpy - a Python Toolbox for Magnetotellurics; L. Krieger, J. Peacock; Computers & Geosciences; *accepted with revisions*; 2013

The magnetotelluric phase tensor; T.G. Caldwell, H.M. Bibby, C. Brown; GJI, 158, pp. 457-469; 2004;

On the robust estimation of power spectra, coherences, and transfer functions; A.D. Chave; D.J. Thomson, M.E. Ander; JGR, 92/B1, pp. 633-648; 1987

Robust multiple-station magnetotelluric data processing; Geophysical Journal International; G.D. Egbert 1997, GJI, 130/2, pp. 475-496; 1997

Three-dimensional magnetotelluric inversion: data-space method; W. Siripunvaraporn, G. Egbert, Y. Lenbury, M. Uyeshima; PEPI, 150(1), pp. 3-14; 2005

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

Magnetotelluric monitoring of a fluid injection: Example from an enhanced geothermal system; J.R. Peacock, S. Thiel, P. Reid, G. Heinson; GRL, 39(18); 2012

Snuffler - A nifty seismogram browser; S. Heimann;
`http://emolch.github.io/pyrocko/`

Packages

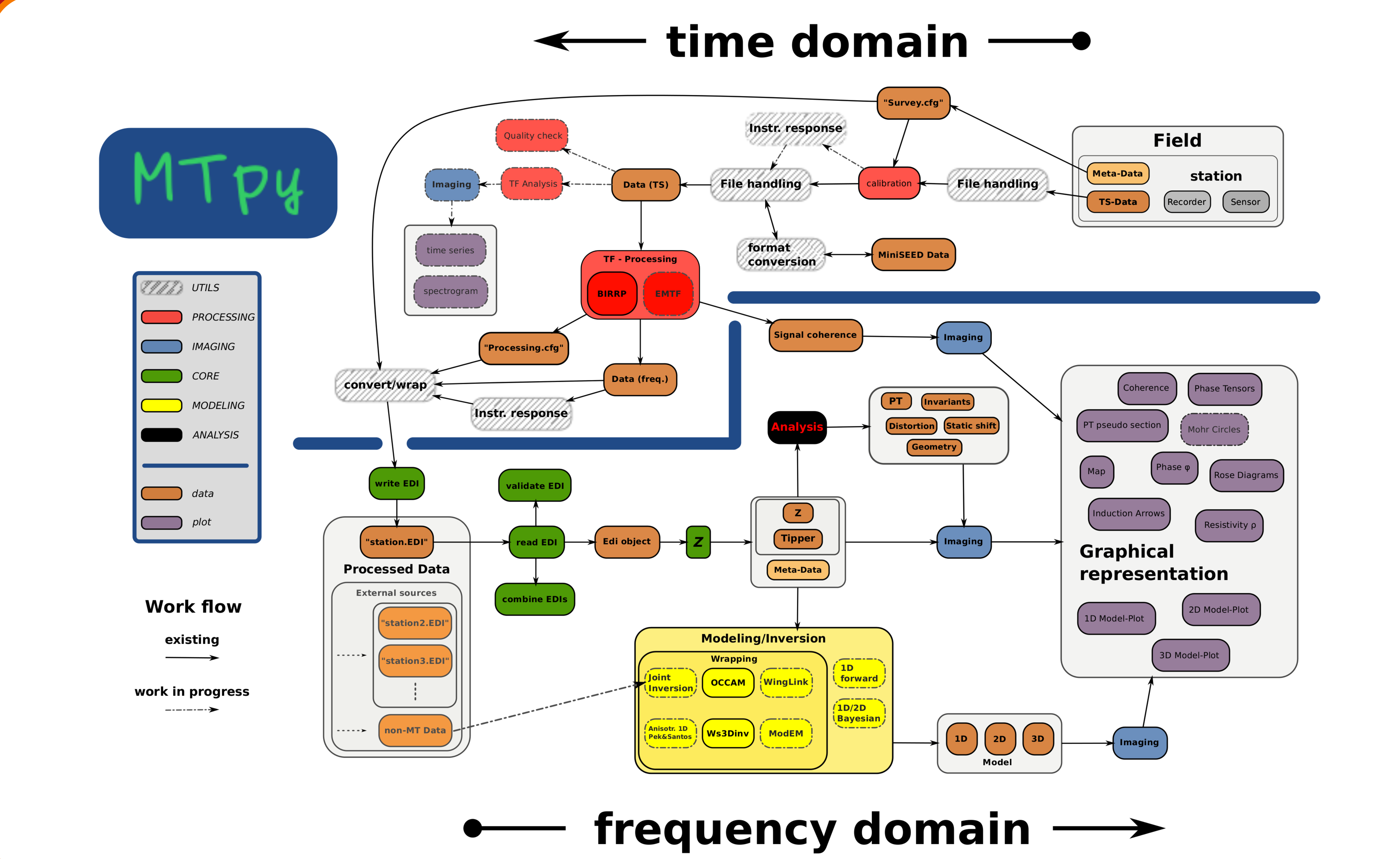
The variety of structurally different tasks within the standard processing of MT data are reflected by the structure of *MTpy*. Six standard sub-packages are defined, which comprise the handling of: 1) raw TS data, 2) core information (impedance tensor and EDI files), 3) inversion and modelling, 4) frequency domain data analysis, 5) visualisations, 6) data structure and files in general.

Sub-package	Task	Functionalities
core	EDI files, impedance tensor	reading, writing, handling, modifying
processing	Time series data	frequency-domain transformation, calibration, re-orientation (, tf-analysis, filtering)
analysis	Phase tensors, geometry, dimensionality	evaluation of Z and Tipper information
imaging	Graphical representations	visualisation of (analysed) data, maps
modeling	In- and output handling for (external) modelling and inversion tools	generation and conversion of in- and outputs for running modelling software
utils	Supplementary scripts and tools	numerical calculations, coordinate transformations, shell scripts, file handling, format conversions

Sub-packages can be independently imported into Python scripts.

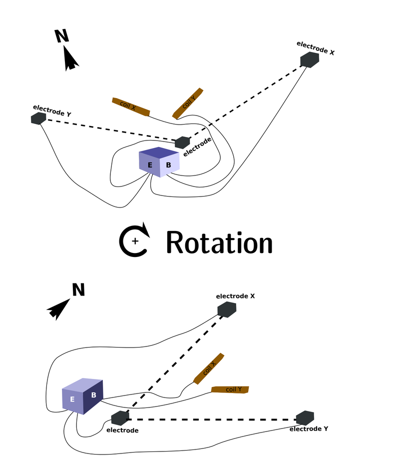
The *MTpy* package structure can be adapted or extended in the future. Additional sub-packages can be defined locally if required.

The modules provide basic functionality. Handling of multiple data files/folders use loops over the functions for single files (exception: visualisation). These are carried out by scripts.



Processing

The term *processing* refers to the handling of time series (TS) data.



***MTpy* TS data**

- simplifies re-processing
- ASCII – 1 file per component
- basic header & comments

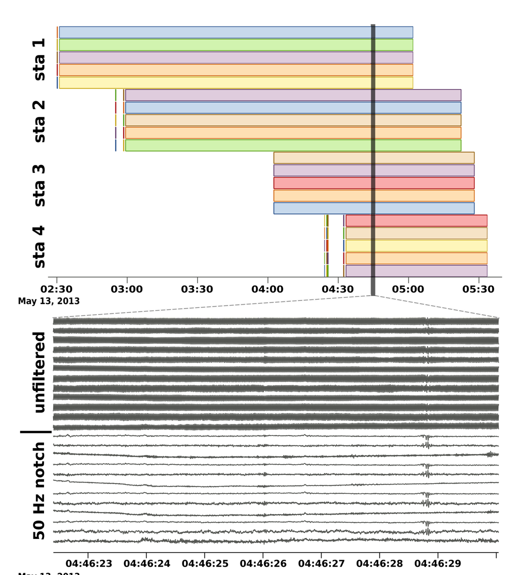
raw TS data sort, calibrate, re-orientate, add header

MTpy TS data write, parse

MiniSEED format conversion

BIRRP prepare input data, determine parameters, run, incl. remote referencing

EMTF ...in progress



MiniSEED allows easy storage and visual analysis, e.g. for data availability (top) and by frequency filtering (bottom) visualised using *Snuffler*

EDI file handling

EDI files are the most common way to store processed frequency domain MT data. *MTpy* provides tools for reading and writing.

Fundamental tasks

⋆ read (Z, spectra, ρ/φ)

⋆ write, merge

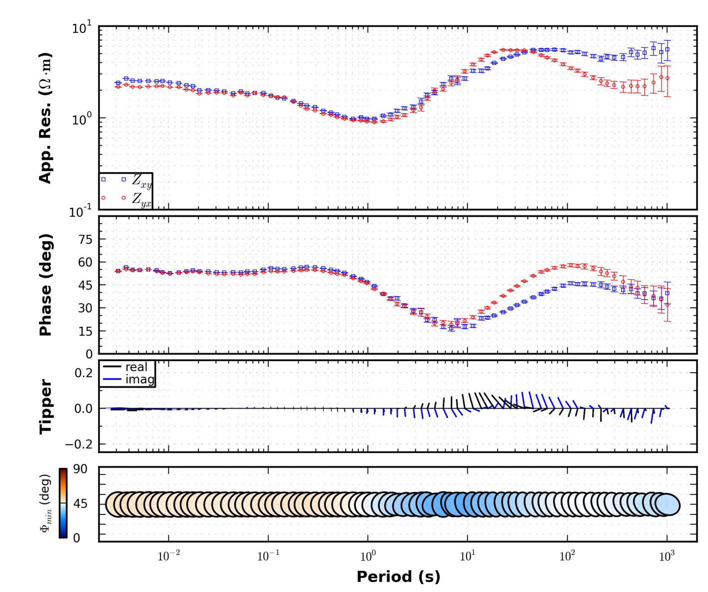
⋆ rotate, scale, average, analyse

- one file per station
- unifies EDI structure
- follows EDI standards
- parsable for *SPUD* database (`www.iris.edu/spud/emtf`)

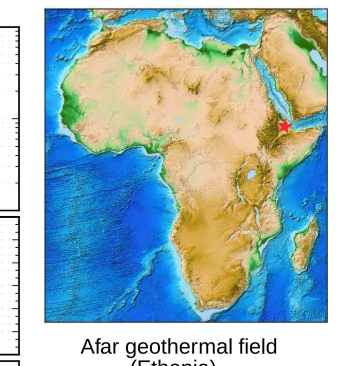
Examples

```
>>> import mtpy.core.edi as EDI
>>> edi_object = EDI.Edi()
>>> edi_object.readfile('filename.edi', data_type='spectra')
>>> edi_object.rotate(234)
>>> edi_object.writefile('newname.edi')
>>>
>>> res,phs,res_err,phs_err = edi_object.res_phase
>>> Z_invariants_dictionary = e.Z.invariants
>>> tipper_object = edi_object.Tipper
>>> meta_data = edi_object.head
>>> Z_1D_component = e.Z.only1d
```

ρ_a/φ plots – extended



Single station – ρ_a/φ / Tipper / PT curve plot – Afar data example



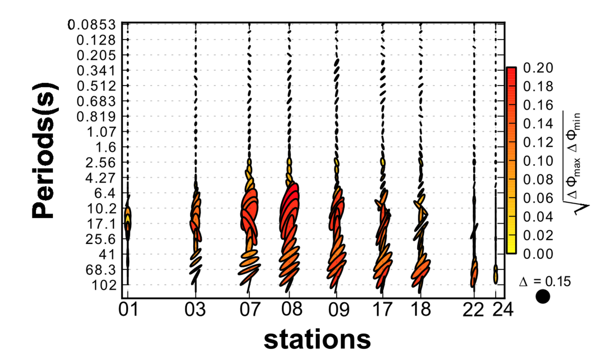
Afar geothermal field (Ethiopia)

- Period dependent information
- Combined plots
- Resistivity/Phase
- Phase Tensors
- Induction arrows (real&imaginary)
- ...to be continued

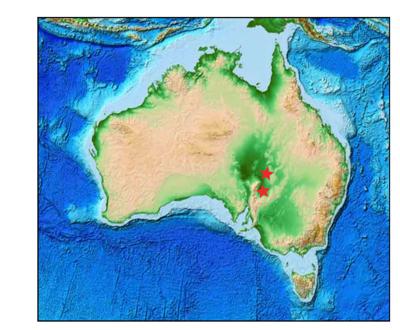
This visualisation allows an easy analysis of local vertical (ρ_a/φ) and regional (Tipper arrows) resistivity structure, as well as local dimensionality (PT).

Phase Tensors

Introduced 2004 by Caldwell et al., Phase Tensors (PT) have become a standard tool for the analysis of MT data. PT are represented by ellipses containing information on the geometrical structure of the Earth. They can be colour coded to represent further data, e.g. skew angles.

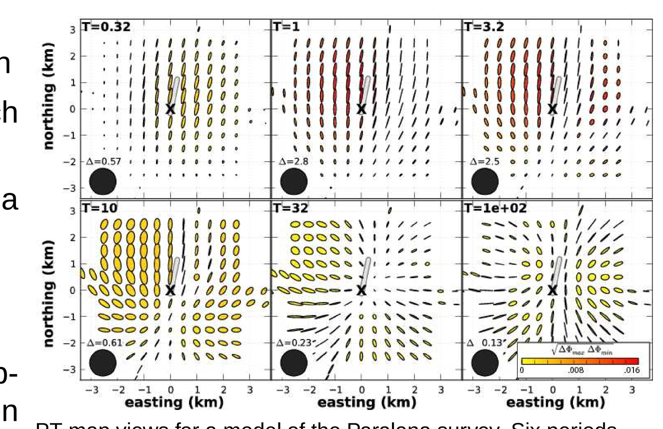


Residual Phase Tensor pseudo section for Habanero



Locations of South Australian EGS test sites: *Habanero* and *Paralana*

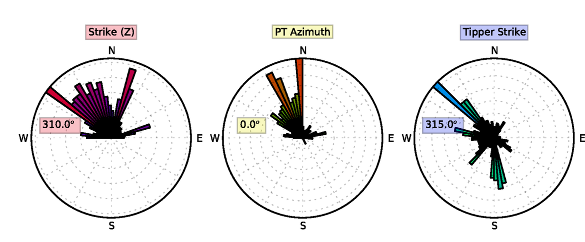
- Analysis of pure data - no modelling!
- No (galvanic) distortion
- Interpretation for each signal period
- Depth dependence via skin depth equation
- Define *Residual PT* $\overline{\Phi}_{01} = \mathbb{1} - \Phi_0^{-1}\Phi_1$
- $\overline{\Phi}_{01}$ can visualise subtle changes between states 0 and 1 in the subsurface, e.g. from fluid injections



PT map views for a model of the Paralana survey. Six periods cover skin depths 900 m - 16000 m. Injection location is marked by the black cross, fluid filled fault by the grey bar. Colours code the geometric means of $\Phi_{\min}$ and $\Phi_{\max}$. Black circles show the reference size for the noted change in percent.

Geometry

MT data contains information about local and regional geometry and dimensionality.



Rose diagrams, visualising statistical distribution of geometry parameters *strike* and *PT azimuth*.

- geoelectric strike & PT azimuth
- Z and Tipper information
- determined for each signal period
- distribution over depth (vertical structure)...
- ...or over stations (lateral structure)

Future

MTpy aims to support all MT data processing and analysis. However, the provided functionality is based on experience and local necessities. We hope that the software will grow by contributions of all kinds: new modules, feedback, ideas,.. .

Contribute!