# pyMT

*Release 0.0.1*

**Eric Roots**

**Apr 23, 2020**

# GETTING STARTED

CHAPTER

ONE

# INSTALLATION

## 1.1 Dependencies

- Python=3.7.*
- Numpy
- SciPy
- matplotlib
- pyqt5=5.14.*
- colorcet
- pyshp
- pyproj
- pyvista=0.23.*
- pyvtk
- naturalneighbor

## 1.2 Installing pyMT

Clone the repository:

```
git clone https://github.com/eroots/pyMT.git
```

Install by navigating to the cloned pyMT directory and entering:

```
python setup.py install
```

Or if you intend to modify the code:

```
python setup.py develop
```

All dependencies will be installed during this process, however if you are coming from a clean python installation, it may be best to manually install certain packages with potentially complicated dependencies. These include:

- pip
- numpy
- scipy

- matplotlib
- pyqt
- colorcet

The tested method of installation involves installing each of these with Anaconda prior to installing pyMT.

Note, a previously required package naturalneighbor has been removed from the dependency list. *Data Plot - Main Window* will now instead offer other interpolation schemes based on scipy. If naturalneighbor is installed (it can be installed through pip, or using the wheel at https://www.lfd.uci.edu/~gohlke/pythonlibs/#naturalneighbor), the natural neighbor scheme will still be available.

As of the time of writing this (April 2020), there is an issue between versions of python and pyqt that may cause some GUIs to fail:

- Python needs to be version 3.7.*
- PyQT5 needs to be version 5.14.*

Note that as of now, Anaconda only seems to have access to pyqt5 version 5.9.*. You will therefore have to install pyqt5 using pip, and ensure this version supercedes any other version of pyqt installed in your environment.

An update to pyvista version 0.24.* has also caused an error in setting up the *Model Viewer - Main Window*, so pyvista version 0.23.* is required for now.

CHAPTER

TWO

# TESTING INSTALLATION

Test data is included in the test_data folder. From that folder, try launching each of the GUIs from the command line:

***Data Plot - Main Window*:**: data_plot -n pytest

***Mesh Designer - Main Window*:**: mesh_designer testfile_ModEM.dat

***Model Viewer - Main Window*:**: model_viewer testfile_ModEM.dat testfile_ModEM.model

If they all launch, you should be good to go. If not, most likely there is a dependency issue somewhere.

CHAPTER

THREE

# API

## 3.1 data_structures

### 3.1.1 Dataset

### 3.1.2 Data

### 3.1.3 Error Floors

Error floors may be set in a Data object by setting the corresponding value in the error_floors attribute. Data.error_floors initialized as:

```
Data.error_floors = {'Off-Diagonal Impedance': 0.05,
                     'Diagonal Impedance': 0.075,
                     'Tipper': 0.05,
                     'Rho': 0.05,
                     'Phase': 0.03}
```

The corresponding entries may be modified as any dictionary and then accesing the apply_error_floor method. For instance, to change the error floor of the ZXY and ZYX components to 7.5%, use:

```
data.error_floors['Off-Diagonal Impedance'] = 0.075
data.apply_error_floor()
```

### 3.1.4 RawData

### 3.1.5 Response

### 3.1.6 Site

### 3.1.7 Model

## 3.2 Input File Types

### 3.2.1 Startup File

The startup file specifies the relevant paths and files, as well as the roles of those files. The startup file is broken into seperate data sets, with a % denoting the beginning of a new data set specification, and the subsequent lines giving the

role and paths of the files to be used. Lines starting with a # will be ignored, which allows the insertion of comments. A data set can contain one *List File*, *Data File*, and *Response File*, or any permutation thereof (as long as only one of each is specified per data set). Additionally, you can specify a common path to each file, as well as a separate path which points to the location of the raw data files (EDI or j-format files), although this has not been fully tested, and so specifying relative or absolute paths is the safer option for now. An example startup file is as follows:

```
# This specification assumes allsites.lst, inv.dat, and the EDI/j-format files are␣
↪all in the same folder as pystart
% data_set1
list allsites.lst
data inv.dat
# The list and EDI files are in a folder ./EDIs
% data_set2
list EDIs/broadband.lst
# The list file and EDIs are in ./EDIs and the data and response files are in ./
↪inversion2
% data_set3
list EDIs/broadband.lst
data inversion2/broadband.dat
resp inversion2/inv_response.dat
```

Assuming this file is called pystart and the terminal is in the same folder, the GUI can be launched with (for example):

```
data_plot -n pystart -c data_set1:data_set2
```

### 3.2.2 Data File

The data files are those that are used as input to your inversions. Currently implemented formats are:

- ModEM (2-D)
- ModEM (3-D)
- WSINV3DMT
- Occam2D
- MARE2DEM

Thus far, only ModEM and WSINV3DMT file formats have been used extensively. Other formats, while implemented, have not been thoroughly tested.

Data file handling is implemented through the *Data* class.

### 3.2.3 Response File

The response file output from an inversion. If the format of data and response files is the same for your given inversion code (as is the case for ModEM), then data and response files may be used interchangeably.

Currently implemented formats are the same as for the *Data File* Response file handling is implemented through the *Response* class, which is largely just a subclass of *Data*.

### 3.2.4 List File

A list file specifies the EDI or j-format files you would like to import. The first line specifies the number of stations contained in the file, and each subsequent line is the name of a station. The names can specify .EDI or .dat to

specifically read in EDI or j-format files. The file-reader will look for both if no format is specified, preferentially selecting j-format files if both formats are present.

List files can be used to assign station names to when read into a *Data* object (for instance, a normal WSINV3DMT data file does not contain station names), or to specify the files to be read into a *RawData* object.

### 3.2.5 Model File

The model files that are used as input to and output from the inversions. Currently implemented formats are:

- ModEM (2-D)
- ModEM (3-D)
- WSINV3DMT

Thus far, only ModEM and WSINV3DMT file formats have been used extensively. Other formats, while implemented, have not been thoroughly tested. Furthermore, 2-D ModEM model files have not been used much, and so may result in some unpredictable behavior.

Model file handling is implemented through the *Model* class.

CHAPTER

FOUR

# GUIS

## 4.1 Data Plot

### 4.1.1 Getting Started

#### Launching the GUI

The data plotting GUI is launched from the command line. A number of options exist on exactly how to specify which files you would like to use, which can be accessed using:

```
data_plot -h
```

which will output:

```
Options include:
 -d : Use default startup file "pystart"
 -b : Browse for start file or data files (Not yet implemented)
 -n : Specify the start file you wish to use
 -l : List the dataset names present in the start file you have chosen
 -c : Choose a specific dataset(s) listed within the start file you have chosen
         For multiple datasets, separate names with a colon (:)
```

Options can (and generally should) be used in conjuction. For example, the most common method of launching the GUI is:

```
data_plot -n <startup_file> -c <dataset_name>
```

This will open the *Startup File* and initialize the GUI with the specified dataset. If no dataset is give (i.e., the -c flag is not used), all datasets within <startup_file> will be loaded.

#### Required Files

The data plotting GUI requires at minimum a *Startup File* containing at least one of the following:

- *List File*
- *Data File*
- *Response File*

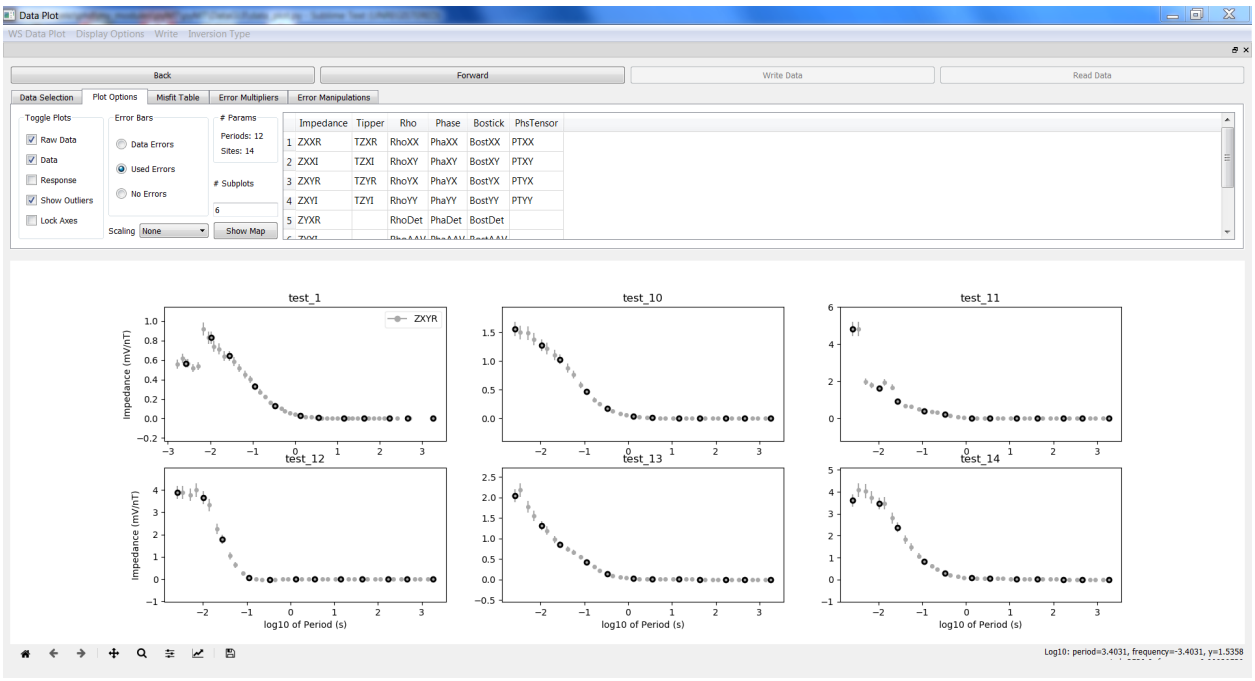
### Default Behavior

A few things to note about the default behavior of data_plot:

- Raw data (data read in from EDI / j-format files indicated in a list file) are shown by filled circles
- Inversion data (data read in or created for use in inversion) are indicated by filled circles with a black outline
- Response data (data read in from an inversion response file) is indicated by a solid line
- By default, data is displayed as is. Particularly for impedance data, it is useful to view it multiplied the periods or square root of the periods to be able to visualize both short and long periods equally well. This can be done through the *Scaling* drop-down menu.
- Some of the features of the Data Plot GUI and the *Data Plot - Map Viewer* require raw data in order to operate (e.g., addition of periods). Therefore it is generally best to include a *List File* in all datasets to be plotted.
- The *Data Plot - Map Viewer* plot is updated any time a contained element is changed. In general this operation is fast. However, if a pseudosection is being plotted in the *Map Window*, any operation involving a new plot (including changing the viewed sites using the *Forward and Back Buttons*) may become quite slow.
- **When only a *List File* is specified for the loaded dataset, a *Data* object will be initialized by taking logarithmically spaced p**
  - In this instance, the *Data Plot - Map Viewer* is not initialized properly, and no site locations are shown. The workaround is to re-sort the station locations (e.g., sort by west-east then re-sort back to default) and click the forward or back button. These operations should get the stations plotting.
- When multiple datasets are loaded in, the misfit table may not be properly updated to reflect the currently selected dataset.

## 4.1.2 Data Plot - Main Window

When the Data Plot GUI is launched, you will see a screen that looks something like:



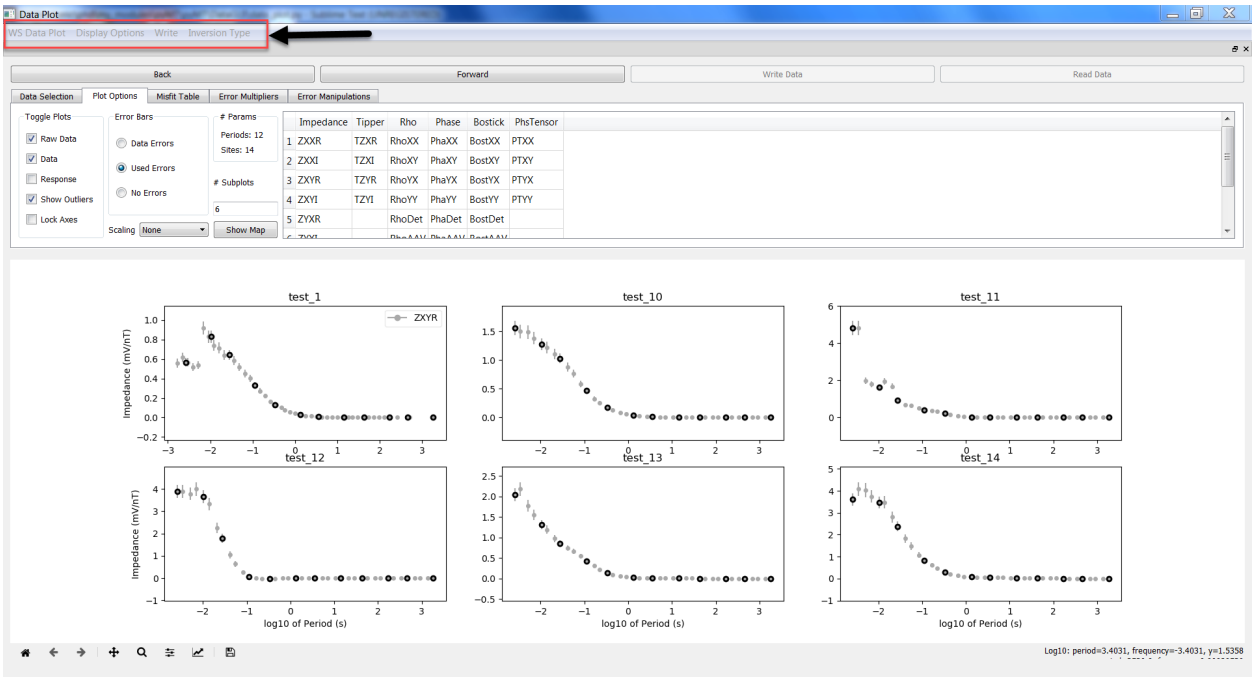

From here there are 5 main areas. From top to bottom:

- *Menu Bar*
- *Forward and Back Buttons*
- *Tabs Window*
- *Plot Window*
- *Toolbar*

The Forward and Back Buttons as well as the Tabs Window are contained within a dock that can removed from the main window and moved to a different location if desired.

WARNING: Do not exit the docked window (i.e., do not click the x contained in the gray bar of the dock), as there is no way to get it back and the Data Plot GUI will need to be closed and relaunched.

## Menu Bar




The menu bar offers a number of options from various drop down menus.

The WS Data Plot menu does nothing. It is a legacy menu that was never used for anything other than testing.

The Display Options menu, as the name implies, gives a few options for changing the display behavior of the main plots.

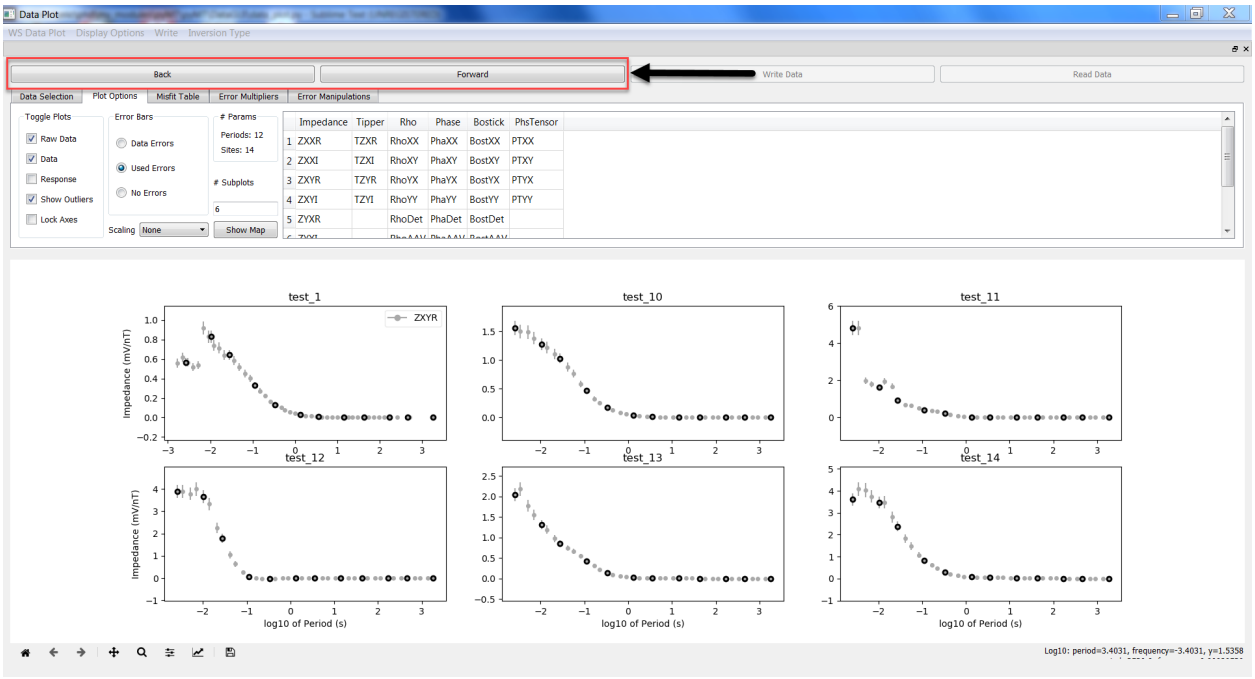
- Phase Wrap: The default behavior when plotting phase data is to have the XY mode in the 1st quadrant, and the YX mode in the 3rd quadrant. The Phase Wrap button forces both to be plotted into the 1st quadrant.
- Error Bars: The errors from both the raw data and inversion data files are plotted by default (if both are available). When creating data files for inversion, you may want to turn one or the other off. Hovering this menu will drop out options to turn the Raw or Data error bars on or off
- Write: This menu gives various output options.
  - List File: Writes out a new list file corresponding to any changes youve made in the GUI (e.g., order changes, addition or removal of stations)

  - Data File: Writes out a data file in the chosen format corresponding to the selected periods and inversion type
  - Current Plot: Writes out the current plot window to a PDF file. Note that a PNG version can also be saved using the *Toolbar*, although the resolution may not be as good.
  - All Plots: Iterates through all stations and writes them all to a single PDF file (using the number of subplots indicated in the *Plot Options* tab).
    * This has not been fully debugged, and may crash the GUI. Known instances of this are when you attempt to overwrite an open PDF file.

- **Inversion Type: Used to control which data components are written to a data file. If nothing is selected, it defaults to all co**

  - Note that some inversion types only work with certain file formats. Inversion types 1-5 work for both WSINV3DMT and ModEM, while inversion types 6-10 only work with ModEM.

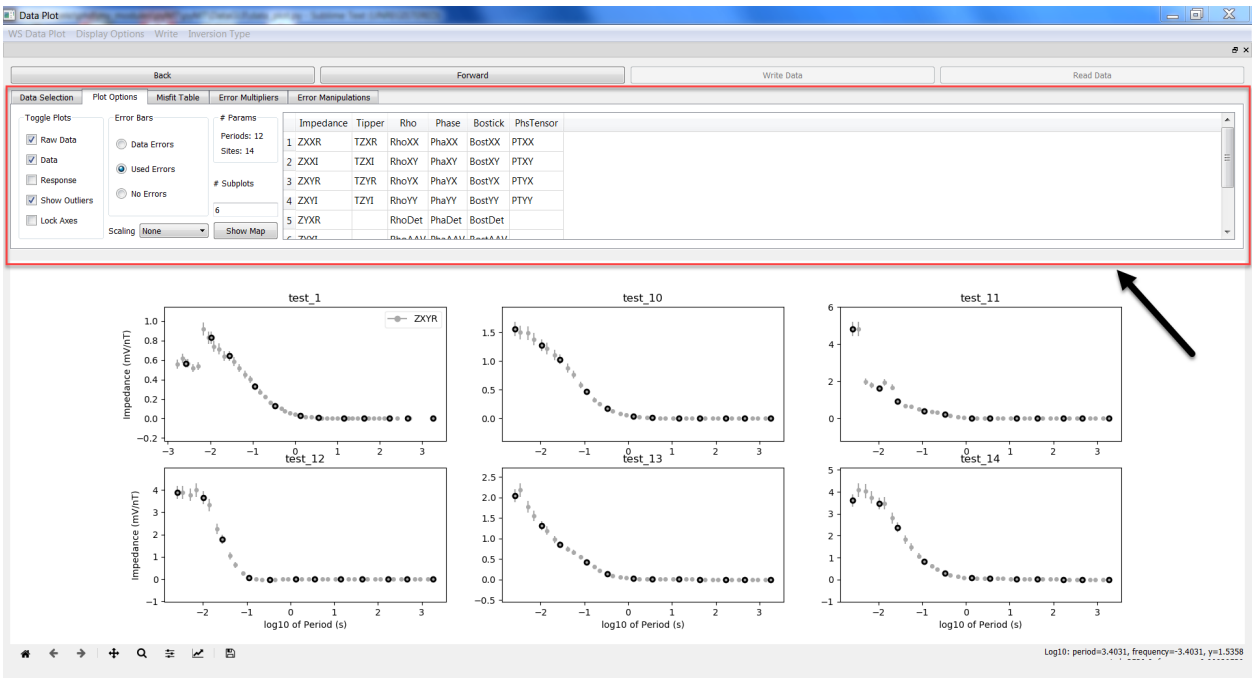
### Forward and Back Buttons

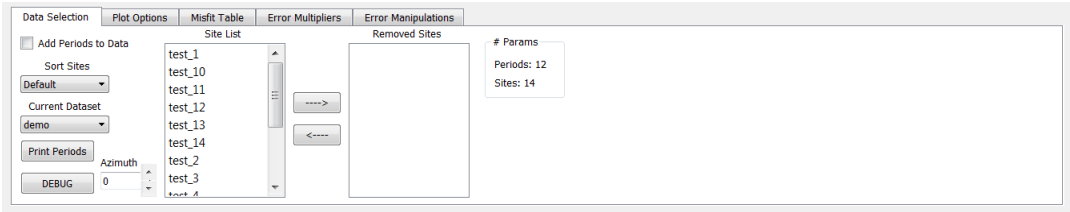



These are used to iterate through the stations to be plotted. There is a known bug here where if more subplots are used than there are stations available, using these buttons will result in one station being re-plotted in the remaining subplots.

### Tabs Window

The Tabs Window is the main control window for the Data Plot GUI. There are (currently) 5 tabs, with each giving different options for either the manipulation or plotting of the data.


Data Plot
Back
Forward
Write Data
Read Data
Data Selection
Plot Options
Misfit Table
Error Multipliers
Error Manipulations
Toggle Plots
Raw Data
Data
Response
Show Outliers
Lock Axes
Error Bars
Data Errors
Used Errors
No Errors
# Params
Periods: 12
Sites: 14
# Subplots
Scaling
None
Show Map
Impedance
Tipper
Rho
Phase
Bostick
PhsTensor
test_1
test_10
test_11
test_12
test_13
test_14
ZXYR
Impedance (mV/nT)
log10 of Period (s)


Data Selection
Plot Options
Misfit Table
Error Multipliers
Error Manipulations
Add Periods to Data
Sort Sites
Default
Current Dataset
demo
Print Periods
DEBUG
Azimuth
Site List
test_1
test_10
test_11
test_12
test_13
test_14
test_2
test_3
Removed Sites
# Params
Periods: 12
Sites: 14

### Data Selection

The main controls within the Data Selection tab are the Add Periods to Data checkbox, and the Site List / Removed Sites tables.

### Adding and Removing Periods

By default, clicking on the plots within the *Plot Window* gives behavior similar to what you would expect from a normal plot window (e.g., you can use the *Toolbar* options to pan, zoom, etc.) To add and remove periods from the inversion, you must first override these controls by checking the Add Periods to Data checkbox.

Once checked, you can add periods to the inversion by left clicking on a data point that is not already included in the inversion (i.e., a filled, non-outlined circle within a plot). Doing so will add the selected period to the stored data across all sites, and should result in the data point becoming highlighted across all plots.

Periods can be removed from the inversion data by right clicking on an active data point (i.e., one that is outlined in black). This will remove the selected point from the inversion data, resulting in the points outline being removed from all plots.

When you are done selecting periods, uncheck the Add Periods to Data checkbox to resume normal plot behavior.

Note that if a period is selected that is not available at all stations, the data from the closest available period will be used instead at those stations. Be careful when adding periods that are not available at all stations. In extreme cases (e.g., when mixing AMT and broadband data), the closest available period at some sites may be too far from the selected period to be reliably substituted, and essentially becomes dummy data. The error bars on any dummy data must be increased (either within the Data Plot GUI, or programatically), or the data point itself must be manually edited (either programatically or within a data file).

### Sorting Sites

By default, the order of the stations is the same as the order in which they were read in (i.e., the order in which they appear within a *List File* or *Data File*).

The Sort Sites drop down menu can be used to reorder the stations by location. Available options are west-to-east, south-to-north, and Clustering. The Clustering option attempts to sort the stations by proximity to each other, so nearby sites are listed next to each other.

At any time you can restore the original ordering by selecting Default.

Reording the stations results in the order being changed within the Site List table, which gives the order in which the stations will be displayed when iterating using the *Forward and Back Buttons*

### Current Dataset

The Current Dataset drop-down menu controls which dataset is being displayed. If only one dataset was read in from the used *Startup File*, then only one option will be available.

Note: This feature has not been extensively tested, and at this point is mainly for comparing different inversion results within the same Data Plot window. A known bug exists where changing between different datasets does not update the *Misfit Table*.

### Print Periods

This button will print diagnostic information about the periods contained in the dataset into the Debug Window (the far right window within the Data Selection tab).

The printed periods correspond to those periods that are available at least 50% of the stations. The first column gives the period values. Negative values indicate frequencies instead (1 / period). The second column gives log10(period).

The last column gives the fraction of stations at which the period is available. Stars indicate that the periods in that row are currently included in the inversion data.

### Azimuth

The azimuth spinbox controls the rotation of the stations (both locations and data coordinate system). The azimuth is always set to a positive number between 0-359.

Data and station rotations are currently set up for 3-D inversion, in that a positive azimuth results in a clockwise rotation (from north) of the station locations, and a counter-clockwise rotation of the data. As a result, the rotating the data within the Data Plot GUI is not suitable to rotation and projection into 2-D.

This feature has been tested, but not used extensively. Therefore, always double check that the location and data rotations are correct. Inverting incorrectly rotated data leads to incorrect models.

### Site List

The Site List and Remove Sites indicate which sites are currently included in the inversion data, and which have been removed, respectively.

Sites can be removed by selecting them within the Site List and clicking the right arrow (to the right of the list). Sites that have been removed can be added back in by selecting them within the Removed Sites table and clicking the left arrow.

Currently there is no way to add new sites to an existing *Data File* from within the Data Plot GUI. Such an operation can be done programatically through the API. See the *Recipes* section for more details.

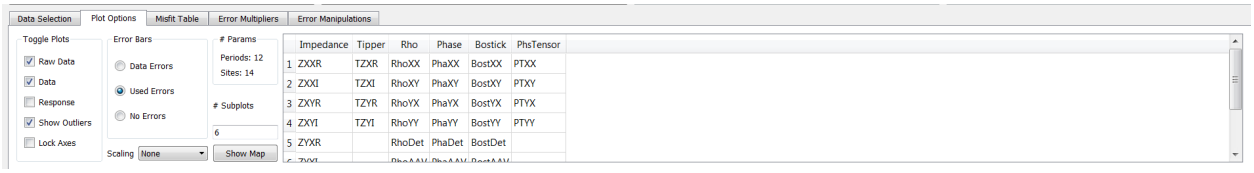
### Diagnostic Windows

There are 2 additional widgets within the Data Selection tab that are non-interactive and only for display purposes.

The first is the # Params box, which simply displays the number of periods and number of sites currently included in the inversion data.

The second is an empty white box that displays diagnostic information. Since the terminal used to launch the Data Plot GUI is inaccessible during operation of the GUI, information will be printed here instead (e.g., information printed out from the *Print Periods* button)

### Plot Options



The Plot Options tab controls what is plotted within the *Plot Window*. Additionaly, the *Data Plot - Map Viewer* is accessed from within this tab.

### Toggle Plots

The Toggle Plots groupbox contains checkboxes for toggling elements to be plotted, as well as some aspects of the plots themselves.

The Raw Data, Data, and Response checkboxes toggle on and off the plotting of their respective components. If a particular data type is not available, the corresponding button *should* be inactive.

The Show Outliers checkbox adds or removes outliers from the *Plot Window*. This does not remove them from the inversion data or the raw data, but rather trims the axis limits to not include such data within the plots.

It is worth mentioning that the algorithm used to detect outliers is very basic and not particularly robust.

The Lock Axes checkbox is used to force all plots to have the same axis limits. Useful when flipping through the stations to check, e.g., data quality.

### Error Bars

The Error Bars groupbox controls which error bars are plotted.

When Data Errors is selected, the errors from the original data file are used.

When Used Errors is selected, the errors that will be used in the inversion are shown, i.e., the original data errors multiplied by the error map.

The No Errors button removes all error bars from the plots.

### Scaling

The Scaling drop-down menu controls how the plotted data is scaled when viewed. None indicates that the unaltered data is shown. Periods multiplies each data point by its period. Only useful when viewing short and long period data similtaneously.

Sqrt(Periods) multiples each data point by the square root of its period. This is the most common option when viewing impedance data, as it allows both short and long period data to be viewed equally well.

The chosen scale factor is only applied to impedance and tipper data, and not for derived data types such as the apparent resistivity and phase.

### Subplots

The # Subplots edit line controls how many subplots are shown in the *Plot Window*. The default value is 6. Any positive value works here, however larger values will increase the time it takes to change the plots as the program will have to iterate through more stations to generate each new set of plots.

### Show Map

The Show Map button is used to launch the *Data Plot - Map Viewer*.

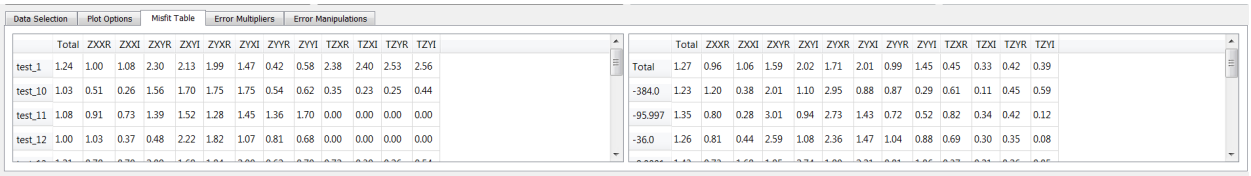
### Data Component Table

The final and largest widget in the Tabs Window is the Data Component Table. This controls which components are plotted into the *Plot Window*.

Any number of components can be plotted at the same time by using CTRL+click, SHFT+click, or click+drag provided they are contained in the same column of the table. If components from different columns are selected (e.g., an ZXXR and RhoXY), a pop-up window will appear with a warning and the selected components will be reset.

There is a known bug where the blank items of the table are selectable. However, this results in the first available component being plotted instead, and so should not break the GUI.

### Misfit Table




The Misfit Table tab shows information regarding the misfits (calculated as the RMS differences) between the data and response. The tab is separated into two tables.

The left hand table shows the misfit on a per site basis, with each row representing a single site and each column representing a given data component.

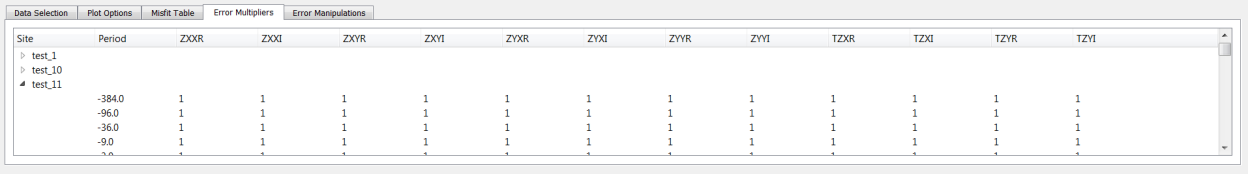
The right hand table gives the misfits on a per period basis, again with each row representing a single period and each column representing a single data component.

Both tables have an additional column for the total misfit across each row. For the left table, this means the total misfit per site (averaged across all components and periods), and in the right table it gives the total misfit per period (averaged across all sites and components).

The right hand table also contains an additional row at the top that gives the total misfit across all periods and sites for each column. As such, the top left entry of the right table gives the total misfit of the inversion.

Note, the RMS misfits in this table may differ slightly from what logged by ModEM as pyMT always applies an error floor to any data read in. As ModEM does not explicitly use (or store) an applied error floor, the hard-coded error floors of pyMT may differ from those used in the inversion. As a result, if you used a lower error floor than those coded into pyMT, the misfits shown in this table will be lower.

### Error Multipliers




The Error Multipliers tab is the main control window for setting the data errors. In order to compatible with both WSINV3DMT and ModEM, the error control takes form of integer multiples. The multipliers shown in a Tree Widget. Site names are given in the left-most column with a small arrow beside them. Clicking on the arrow will collapse or expand the error multiplier tree for that site. Changing the errors is done by double clicking on the multiplier value

for the desired site, period, and component and changing it to the desired value. Assuming you edit the multiplier for a site and component that is currently being plotted, the new errors should be displayed immediately after the edit is completed.

As manually editing many sites, periods, and components is *incredibly* tedious, a few keyboard shortcuts have been made available.

- Holding SHFT while double clicking to edit a multiplier will result in all periods for the chosen site and component being changed.
- Holding CTRL while double clicking will edit the multiplier for all components for the chosen site and period
- Holding ALT while double clicking will edit the multiplier for all sites for the chosen component and period

The keyboard shortcuts may be used in conjunction with one another as well. For instance, holding SHFT+CTRL while double clicking to edit will change the multipliers for all periods and all components across a single station.

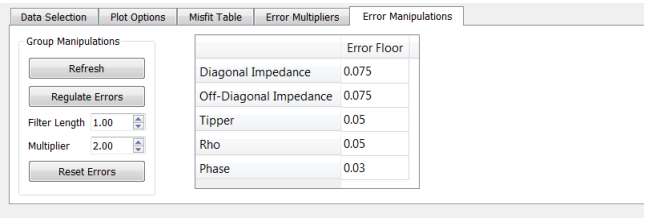
Note, double clicking to edit a multiplier and then entering the same value will result in no changes being made, regardless of any keyboard shortcuts being used.

Default behavior of the Error Multiplier tree widget is to have the stations automatically expand and scroll to the sites currently displayed in the *Plot Window*.

The Error Multiplier tree currently only accepts integer values. However, negative values may be used to decrease the applied errors. For example, setting a multiplier to 5 increases the error by, and subsequently setting the multiplier to -2 would divide the error by 2, resulting in a total error of 2.5 times the original error.

Entering a negative value does however result in the multiplier value being reset to 1, in order to ensure consistent use between ModEM and WSINV3DMT data formats.

### Error Manipulations




The Error Manipulations tab allows for more broad-stroke error setting compared. Within the Group Manipulations box are 3 buttons. The Refresh button is a debug button. All it does is force the *Error Multipliers* tree to reset. This button should not be needed, but does nothing to the data itself.

The Regulate Errors button automatically sets errors for the entire dataset in the following manner. A smoothed version of the raw data is calculated using a smoothing filter.

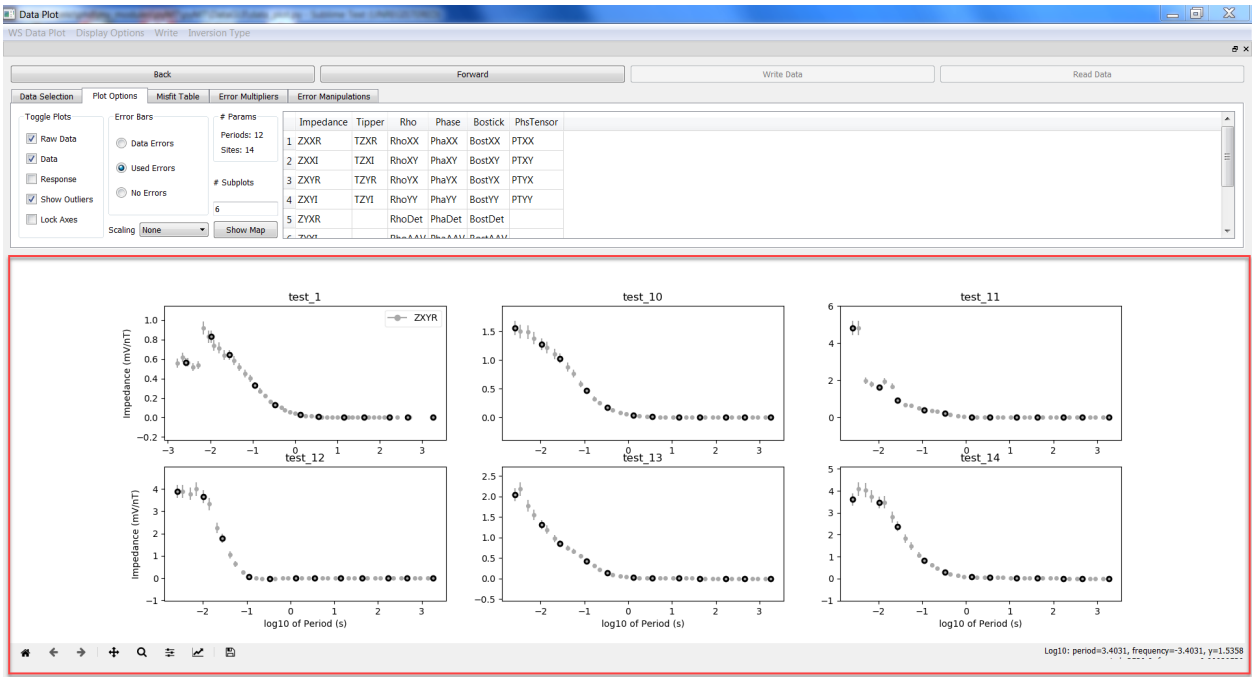
The filter length is given by the value in the Filter Length spinbox. The higher the value, the more smoothing is applied (values between 0.7 and 1.2 have are generally good).

The distance between the data points and the smooth curve is then calculated, and multiplied by the value in the Multiplier spinbox. This value is then set as the error.

The result of this process is that data that is already nearly smoothly varying have relatively low errors, while outliers jittery data will have higher error.

The Reset Errors button simply resets all errors to the error floor. Floor values are given for each set of components in the table to the right of the Group Manipulations groupbox.

### Plot Window




The Plot Window contains all the plots of the data. Nearly all plotting options can be found either within the *Plot Options* tab, or the *Menu Bar*. The currently displayed data components will be given in a legend within the 1st (top left corner) plot.

### Toolbar

The Toolbar is the basic Matplotlib toolbar. It contains buttons which may be activated to interact with the plots within the *Plot Window*.

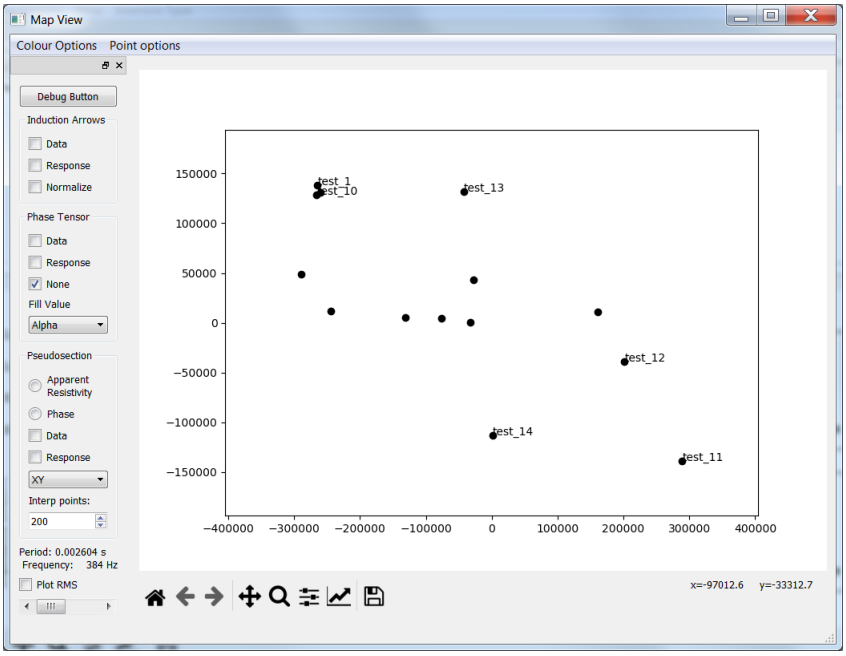
It contains, from left to right:

- Home: Resets the views for all plots
- Back: Returns to the previous view
- Forward: Goes to the next stored view
- Zoom: Activates a zoom cursor. Drawing a rectangle in any subplot will zoom to that locations within that plot
- Subplot Options: Offers options for customizing the margins and general layout of the subplots. The most useful option here is the tight layout button, which will force the subplots to fill the Plot Window
- Axis Options: Contains options for customizing individual axes
- Save: Save the current Plot Window to a PNG file.

In addition to the Toolbar buttons, hovering over any of the plots will show the x-y coordinates of the cursor in terms of the frequency and period.

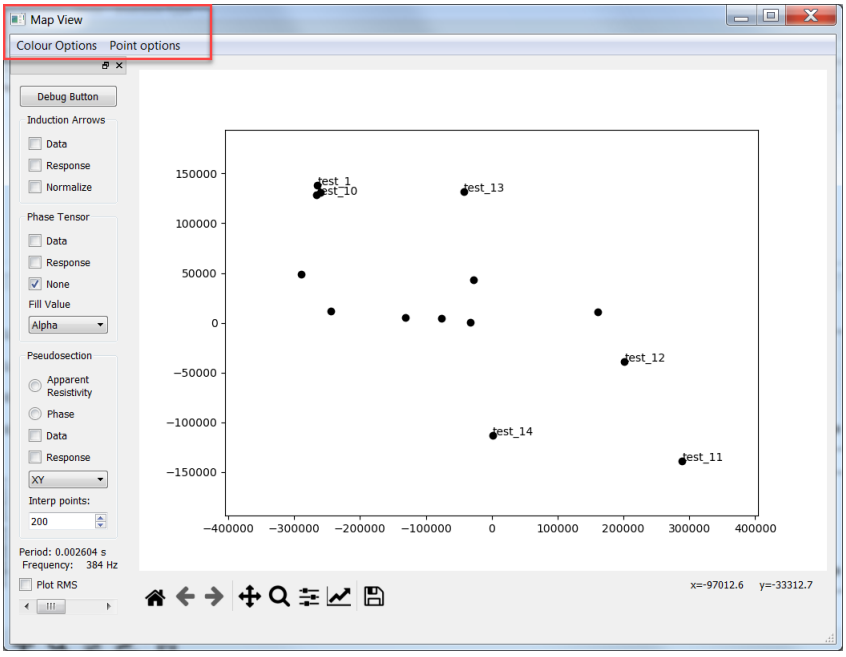
## 4.1.3 Data Plot - Map Viewer

The Map Viewer is launched by clicking the *Show Map* button within the *Plot Options* tab. This window is used to view the locations of the stations, as well as plot various data types in map view. The window is broken into a *Menu*




*Bar*, *Control Dock*, and the *Map Window*.

### Menu Bar




The Menu Bar has a number of options to customize the plotting within the *Map Window*.

### Colour Options

The Colour Options menu contains options for colour map and colour limit selection.

Use the Colour Map menu to select the colour map to be used within the Map Window.

The Color Limits menu is used to customize the lower and upper data limits to be plotted. Separate colour limits may be used for apparent resistivity, phase, and difference pseudosections.

The colour limits for fill values when plotting *Phase Tensor* data is currently hard-coded. This will likely be changed in a future release.

### Point Options

The Point Options menu has options for customizing the appearance of the point related data.

The Annotate sub-menu has options for controlling how stations are annotated. By default, only the sites currently active within the *Plot Window* are annotated. Annotations can also be turned off completely, or set to have all sites labelled.

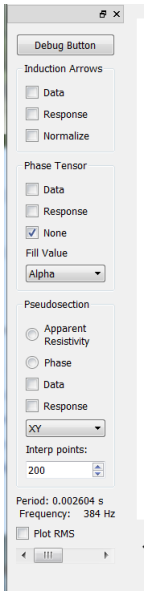
The Marker sub-menu contains further menus for controlling the appearance of the site markers.

The Phase Tensor sub-menu is used the control size of the phase tensor ellipses.

The Induction Arrow sub-menu is used to change the relative length of the induction arrows.

Note that the scaling of the phase tensor ellipses and induction arrows has not be tested on all survey sizes, and so will likely need to be modified. The phase tensor ellipses are pre-normalized and so should generally plot well, however the induction arrows are plotted as is by default. In most cases, it is best to normalize their length (see the following section)

### Control Dock




The Control Dock is the main control panel for the Map Viewer. The various group boxes give options for plotting induction arrows, phase tensor ellipses, and apparent resistivity and phase pseudosection.

The currently plotted period / frequency is seen near the buttom of the Control dock, and can be changed using the nearby horizontal slider bar.

### Induction Arrows

The Induction Arrows group box allows for plotting of the induction arrows (in Parkinson convention) within the *Map Window*.

The Data and Response buttons plot the the induction arrows from the inversion data and response files in black and red, respectively.

The Normalize button scales the arrows so that they all have the same length. This is nearly always required in the current release, as a single noisy high amplitude induction arrow will tend to drown out all the others if the lengths are not normalized.

### Phase Tensor

The Phase Tensor groupbox is used to plot phase tensor ellipses in the *Map Window*. The fill value of the ellipses is controlled by the contained drop-down menu.

The Data and Response checkboxes plot the phase tensor ellipses from the inversion data and response files, respectively. If both checkboxes are selected, the phase tensor misfit tensor is plotted, as defined in Heise et al. (2007):

$$\mathbf{\Delta} = \mathbf{I} - \frac{1}{2}(\mathbf{\Phi}^{-1}\mathbf{\Phi} + \mathbf{\Phi}\mathbf{\Phi}^{-1})$$

In this case, the colour of the ellispes may also be filled by the $\delta$ value give as a percentage, defined by:

$$\delta = 100 * |\mathbf{\Delta}/\mathbf{\Phi}|$$

All other phase tensor fill values are defined as in Cadwell et al., 2004. This includes the skew value $\beta$, and so the recommended upper limit for approximate two-dimensionality is $\beta \leq |3^{\circ}|$

### Pseudosection

The Pseudosection groupbox gives options for plotting map view sections of the apparent resistivity and phase. The sections are generated using the Natural Neighbor interpolation scheme of Sibson (1981), as implemented in the naturalneighbor python package (https://pypi.org/project/naturalneighbor/)

The first two radio buttons control whether the apparent resistivity or phase is plotted.

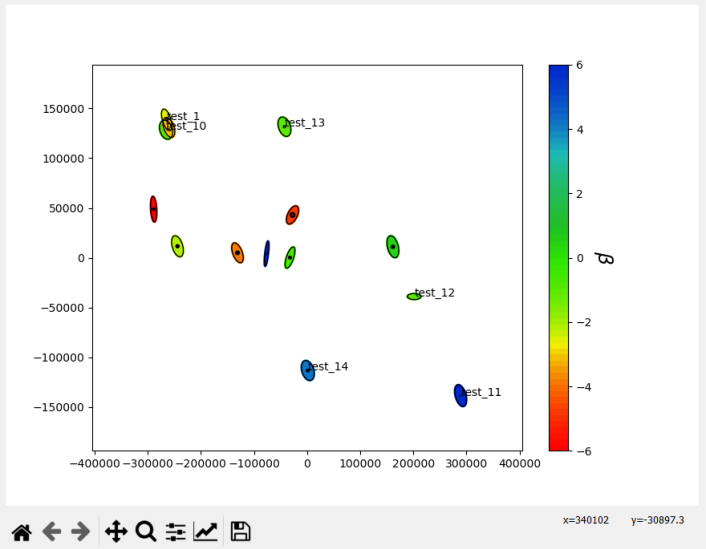
The Data and Response checkboxes plot the sections from the inversion data or response files, respectively, while checking both boxes will result in difference plots. The difference in apparent resistivities is expressed as a percent difference, while for phase it is given in degrees.

The XY, YX, and determinant resistivities and phases may be toggled through the drop-down menu.

The Interp points spinbox controls how many grid points are used in the section interpolation. The default is 200 points. Few points may be used to increase the responsiveness of the GUI, while more points may be used for a finer grid.

### Plot RMS

The relative RMS misfit of each station can be overlaid by checking the Plot RMS checkbox. Unlike the other data plotting options in this window, the RMS misfit overlay does not change per period, but reflects instead the total RMS misfit across all periods and components for each site.




### Map Window

The Map Window is the plot area of the Map Viewer. The site locations are plotted here, along with any data components that have been activated in the *Control Dock*.

By default, only the station locations are plotted along with the annotation style set in the *Point Options* menu. A colour bar will be added if necessary, e.g., when plotting phase tensor ellipses.

Note that the subplot used is set to fill the available space, and therefore does not have an equal aspect ratio.

The toolbar at the bottom of the Map Window may be used to pan, zoom, and customize the plot itself.

The Map Window is reset any time an element of the plot changes. As such, zoom and pans that are applied will not be remembered when the plot is changed. This may be changed in a future release.

## 4.2 Model Viewer

### 4.2.1 Getting Started

#### Launching the GUI

The model viewer GUI is launched from the command line via the command:

```
model_viewer <model_file> <data_file>
```

The order of the files does not matter.

It can take a while to load the GUI, particularly the first time you do so, due to the somewhat heavy dependencies required for 3-D viewing.

#### Required Files

The model plotting GUI requires at minimum a *Model File*, and an optional *Data File*, which is currently only used to plot station locations.

### Default Behavior

Assuming valid files are used, the GUI should launch into the 2-D view, with the slice locations set to X=1, Y=1, Z=1 (bottom, left, and surface slices, respectively)

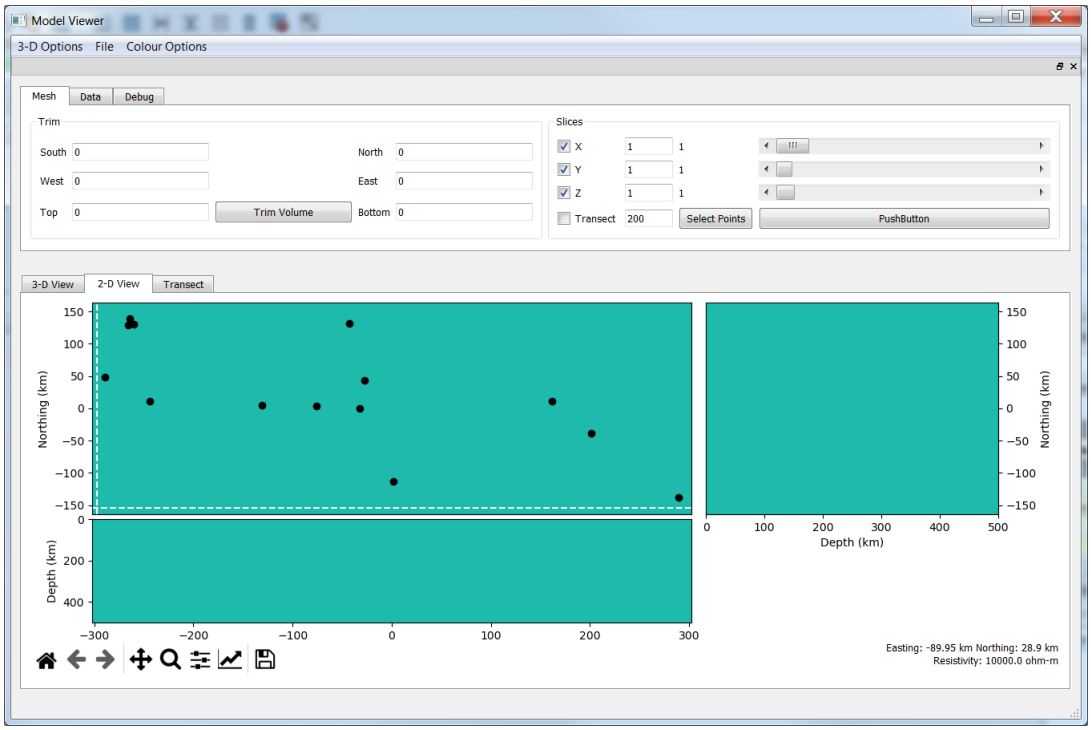
The 3-D view is initialized to a top-down (XY) view.

Default colour map is jet_plus, a modified version of the Matlab default jet with lower and upper colour map limits of 1 and 5 (log10 scale).

Note that currently there is no colour bar shown for the 2-D views.

Currently the data file is used only to plot the station locations over the model. Plotting of induction arrows and phase tensors will likely be added in the future.

## 4.2.2 Model Viewer - Main Window




The Model Viewer window is has 3 sections: The *Menu Bar*, *Control Dock*, and *Plot Window*.

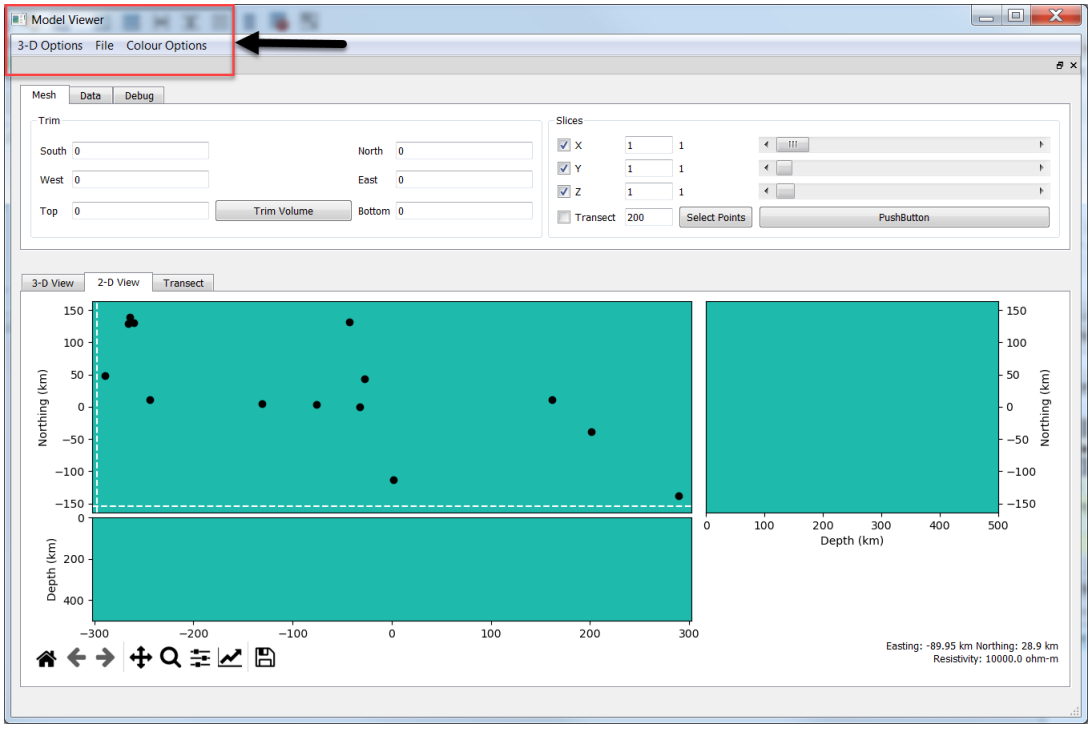
This GUI is still under development, so there may be some widgets in the GUI that have not been fully implemented yet.

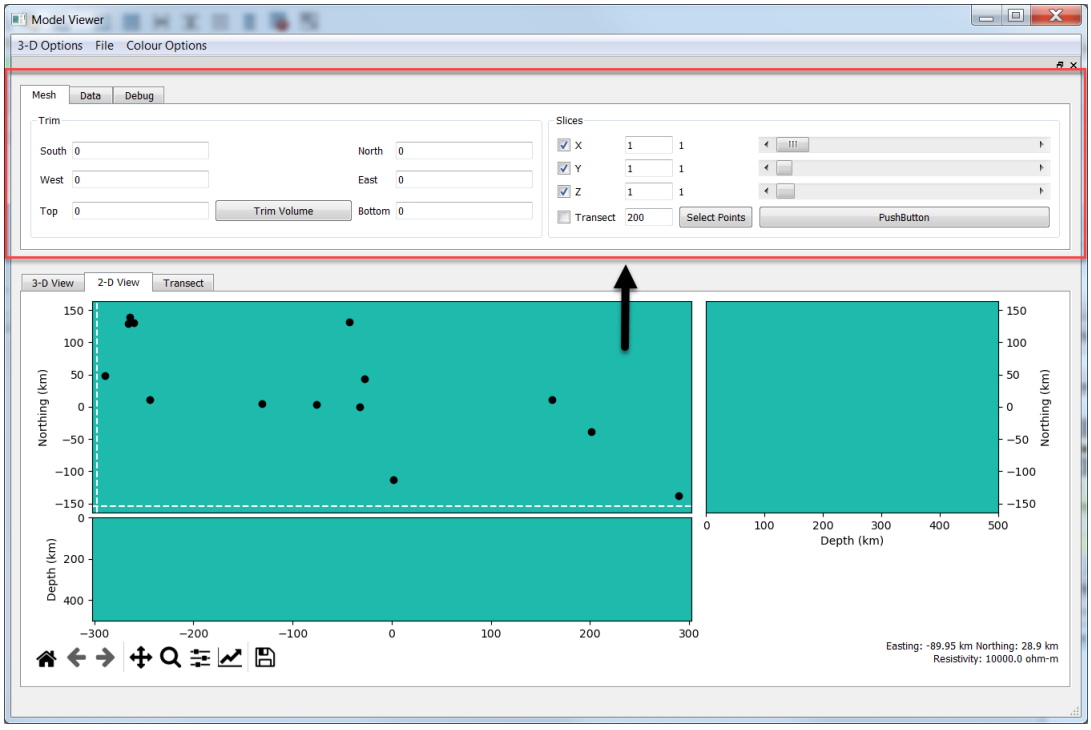
### Menu Bar

The Menu bar currently has two sub-menus.

The 3-D Options sub-menu has options for resetting the *3D View* to an XY, XZ, and YZ view.

The Colour Options sub-menu has options for changing the colour map, as well as the colour limits of all plots.


Model Viewer
3-D Options File Colour Options
Mesh Data Debug
Trim
South 0
North 0
West 0
East 0
Top 0
Trim Volume
Bottom 0
Slices
X 1 1
Y 1 1
Z 1 1
Transect 200
Select Points
PushButton
3-D View 2-D View Transect
Northing (km)
Depth (km)
Easting: -89.95 km Northing: 28.9 km
Resistivity: 10000.0 ohm-m


Model Viewer
3-D Options File Colour Options
Mesh Data Debug
Trim
South 0
North 0
West 0
East 0
Top 0
Trim Volume
Bottom 0
Slices
X 1 1
Y 1 1
Z 1 1
Transect 200
Select Points
PushButton
3-D View 2-D View Transect
Northing (km)
Depth (km)
Easting: -89.95 km Northing: 28.9 km
Resistivity: 10000.0 ohm-m

### Control Dock

The Control Dock contains options for navigating through the model. Only the Mesh tab has been implemented in this release. The Control Dock can be detached from the main window to make more space for the *Plot Window*.

The Mesh tab has two groupboxes: Trim and Slices

The Trim groupbox is used to adjust the bounds of the model that are plotted.

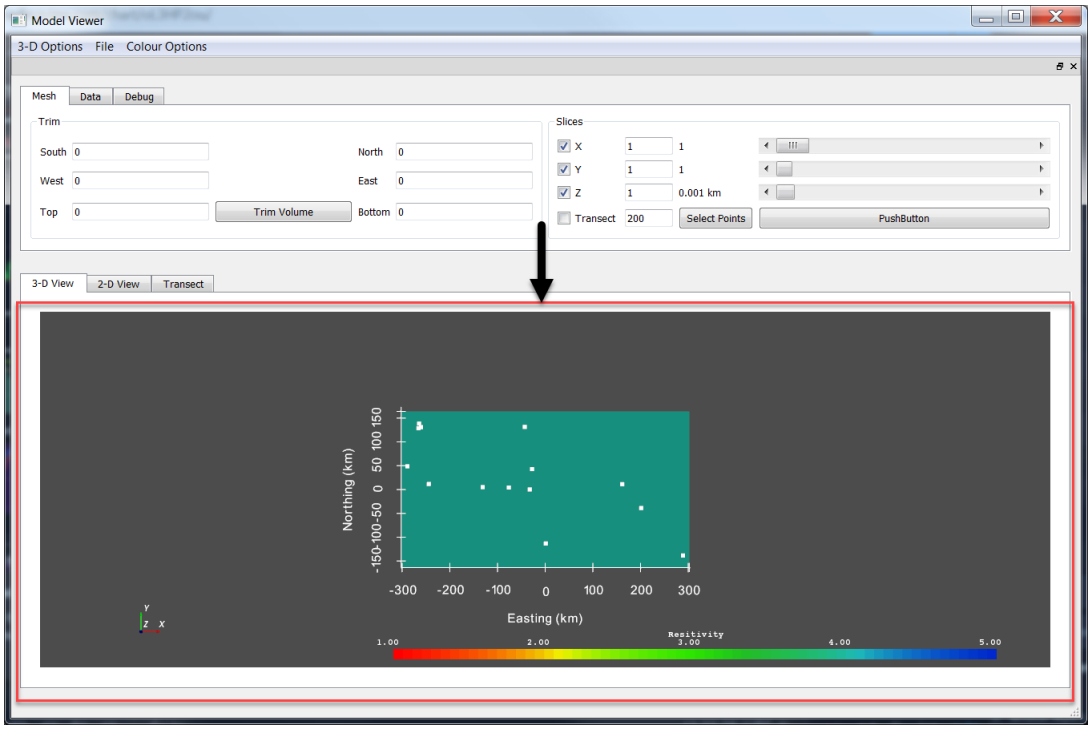
By default, the full model is shown. Edit each line beside the South, North, etc., labels to define how many cells to trim from the corresponding directions. Hit the Trim Volume button to perform the operation. This operation will generate new model plots for all of the subplots in the *2D View*, as well as for the *3D View*.

The Slices groupbox controls which slices are plotted. The X, Y, and Z checkboxes are used to turn off/on the corresponding slices in within the *3D View*. The nearby line edits and slider bars are used to control the location of the each plotted slice.

The last line in the Slices groupbox is used to generate a transect slice along a specified set of points.

- Hit the Select Points button to begin picking. Picking must be performed within the Plan View plot of the *2D View* (hitting the Select Points button will automatically switch the tab focus accordingly).
- Click within the Plan View window to assign points through which the slice will go through.
- When all the desired points have been selected, hit the Select Points button again.
- **A set of points will appear on the Plan View plot indicating your clicks, and the corresponding transect will be plotted wit**
  - The transect slice within the 3D View can be toggled with the Transect checkbox.

### Plot Window




The Plot window is where the model is plotted. It has 3 tabs: *3D View*, *2D View* and *Transect*

### 3D View

The 3D View tab shows the model in a rotatable, zoomable 3D view (similar to in Paraview).

Rotating of the model is done by left-clicking and dragging.

To zoom, either use the scroll wheel or click and hold the right mouse button and drag in or out.

**To pan, hold the middle mouse button and drag.**

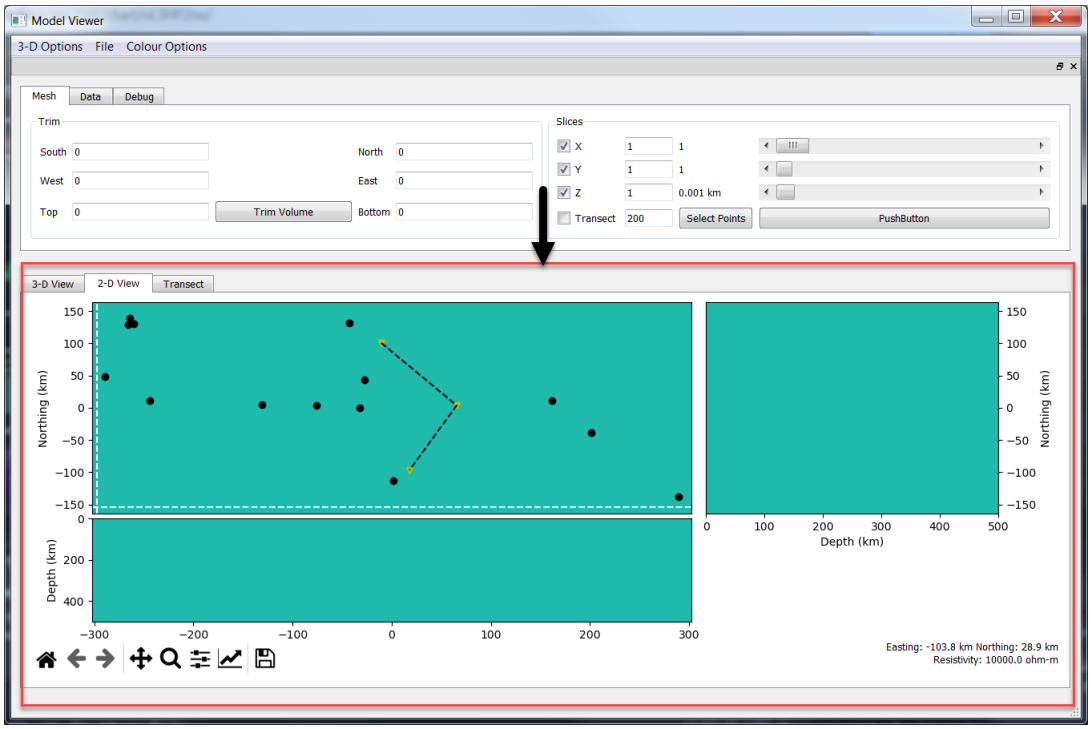
- Panning also changes the center point of the plot (i.e., the point around which rotations will be focused)
- To reset to the original center point, hit r

You can quick-zoom to certain angles using the options in the 3-D Options menu of the *Menu Bar*.

Change which slices are plotting using the controls within the Slices groupbox of the Mesh tab in the *Control Dock*.

Use the W and S keys to toggle between surface and wireframe representations of the model. The default view is surface. The wireframe represention can be used to see the model mesh.

### 2D View




The 2D View tab shows slices through the model in 2-D. It consists of:

- A Plan View plot, showing slices through the XY plane (top left plot)
- An XZ view, showing depth slices through the XZ plane (top right plot)
- A YZ view, showing depth slices through the YZ plane (bottom left plot)

The toolbar at the bottom can be used as in normal matplotlib figures.

The bottom right of the toolbar shows location and resistivity information when the mouse is hovered over any of the 2-D plots.

- Note: Still troubleshooting this feature a bit. It seems to work for the most part, but occasionally the displayed resistivity value seems to not match the feature seen in the plot.

#### Transect

Once a transect path is selected using the Select Points button, the corresponding slice through the model will be displayed here.

- **This plot does not currently properly update when changing certain plot options.**
  - Changes the colour map, colour limits, and Trim are not reflected in this plot.
  - The workaround until this is fixed is to change the plot options first, then reselect the transect plot points with the Select Points button.

## 4.3 Mesh Designer

### 4.3.1 Getting Started

#### Launching the GUI

The mesh designer GUI is launched from the command line via the command:

```
mesh_designer <model_file> <data_file>
```

The order of the files does not matter.

#### Required Files

The model plotting GUI requires one or both of the following:

- *Model File*
- *Data File*

#### Default Behavior

The default behavior is different depending on what files are input. If only a model file is used, the GUI initialized using the given model, and no station locations are plotted. In this case, the Regenerate Mesh button will not be functional.

If only a data file is used, an initial model will be created based on the bounds of the given stations. There is a known bug here where the inital view of the model cuts of the outer edges. Hitting the Add Pads a few times will extend the model out a bit to cover the whole area covered by the stations.

If both model and data files are given, the mesh will be shown as it is in the given model, with the station locations overlaid.

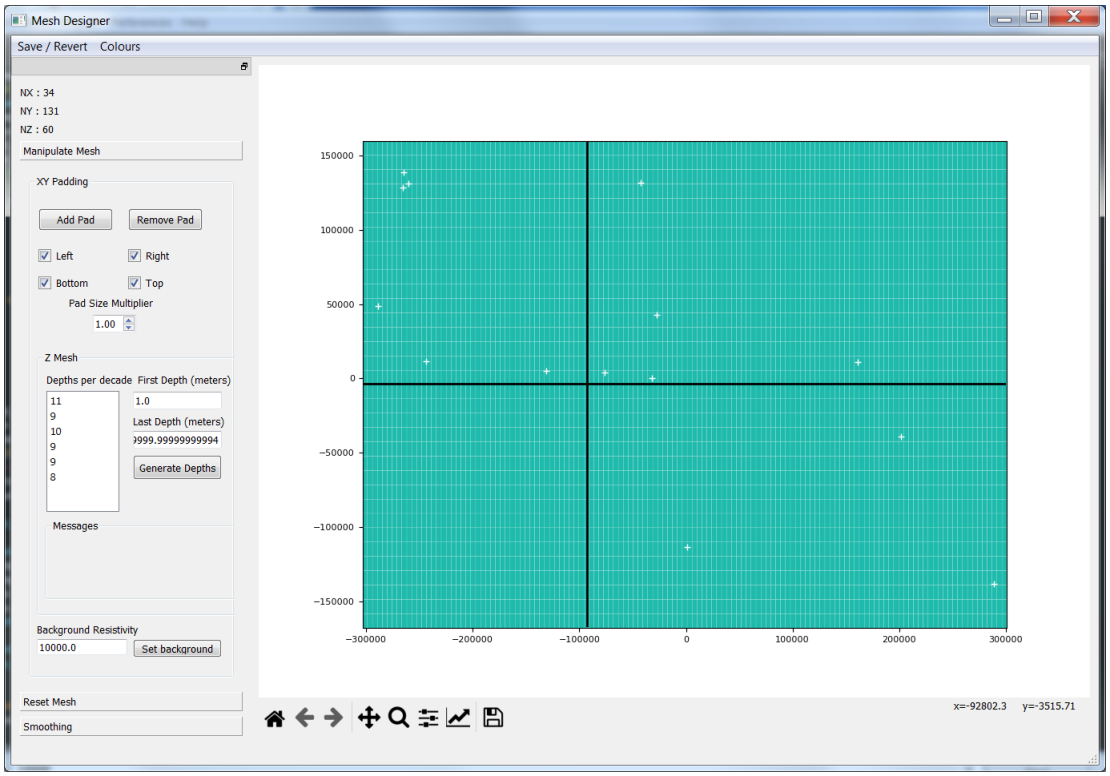
As this GUI is not meant for viewing the model, the slice shown is always the first depth slice.

The GUI works equally well for creating new meshes as it does for modifying existing ones.

- Note: As the definition of the mesh within the ModEM and WSINV3DMT file formats has no explicit origin, all models generated here will have their origins placed in the center of the mesh.

- **For this reason, it is important to ensure that the model is even on the left / right and top / bottom.**
  - If it is not, the definition of the mesh relative to the station locations may not be the same as appears while using this GUI.
- Always double check the output mesh and data files with another tool afterwards (e.g., with *Model Viewer - Main Window*

### 4.3.2 Mesh Designer - Main Window




The Mesh Designer has 3 areas: The *Menu Bar*, *Control Dock*, and *Plot Window*.

#### Menu Bar

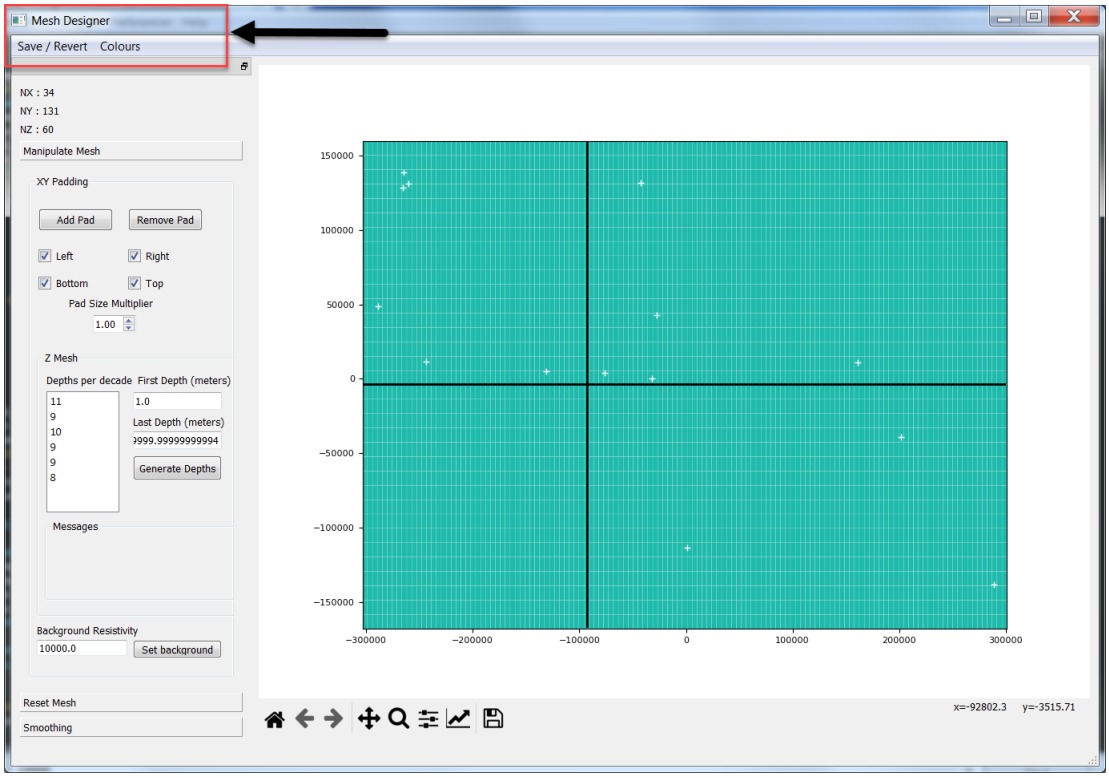
The Save / Revert menu has options for writing the model to a file, as well as saving and reverting progess within the Mesh Designer.

The Write Model sub-menu will open a dialog box that asks for the name of the file to be written to. You will be prompted again if this operation is going to overwrite an existing file. A second prompt will ask for a name to output an associated covariance file (hit cancel to skip).

- The only file format available from here is that of a ModEM 3-D file. This is the same as the WSINV3DMT file format, except that the resistivity values will be given as the natural logarithm.
- If you need the file in WSINV3DMT format, the workaround at present is to change the file format programmatically (see *Recipes*)

The Save Progress sub-menu internally saves any changes youve made to the mesh. This checkpoint can then be reverted to later by using the Revert Progress sub-menu.

The Colours menu is controls the colour map and colour limits used, as well as the colour of the mesh lines.




- As this GUI is meant mainly for editing the mesh, these options are mainly to ensure good visibility of the mesh lines regardless of the resistivity used

### Control Dock

The Control Dock is the main control panel for manipulating the mesh and model. It is broken into 3 tabs:

- *Manipulate Mesh*
- *Reset Mesh*
- *Smoothing*

### Manipulate Mesh

The Manipulate Mesh tab is used to, as the name suggests, to modify and manipulate the mesh used.

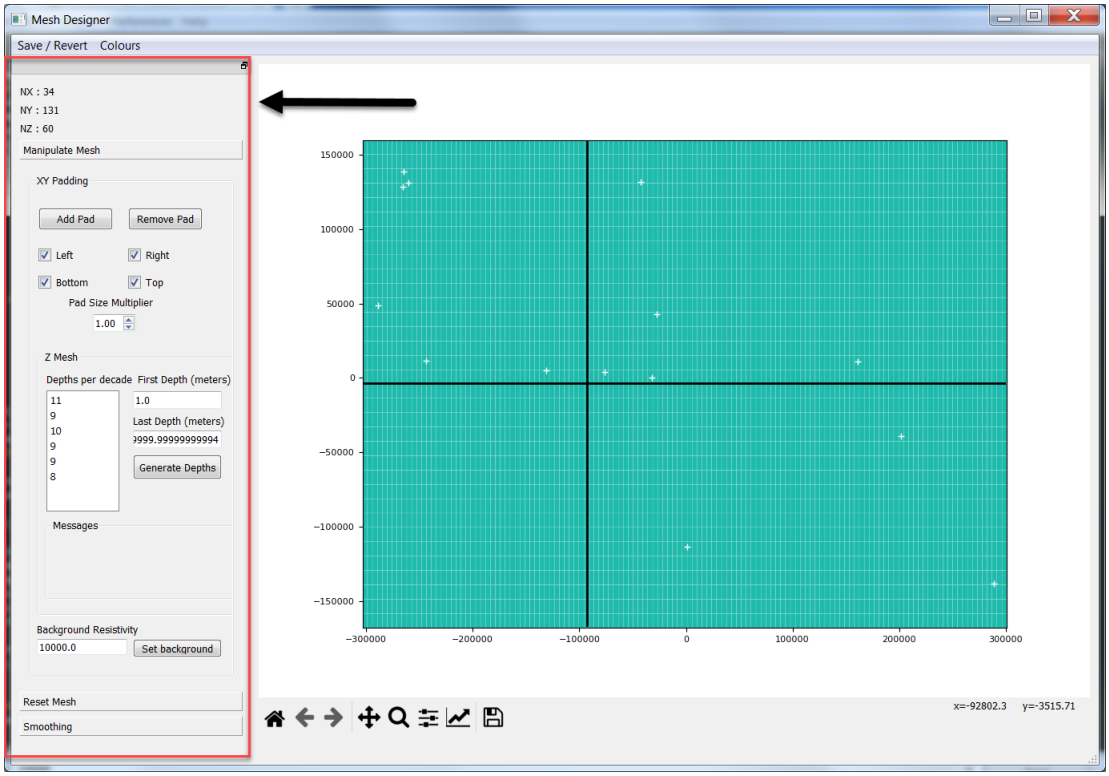
Specifically, there are 3 things that can be modified from this tab: The XY Padding, the Z mesh (or depth mesh), and the background resistivity.

The XY Padding groupbox is used to add and remove padding cells in the XY plane.

The Add Pad and Remove Pad buttons add and remove pads from the left / right / top bottom/ of the XY plane, depending on which of the corresponding checkboxes are selected.

When adding pads, the size of the new pad is determined by taking the size of the outer most cell and multiplying it by the value in the Pad Size Multiplier spinbox.

- Note: As the definition of the mesh within the ModEM and WSINV3DMT file formats has no explicit origin, all models generated here will have their origins placed in the center of the mesh.




- **For this reason, it is important to ensure that the model is even on the left / right and top / bottom.**
  - If it is not, the definition of the mesh relative to the station locations may not be the same as appears while using this GUI.
- Always double check the output mesh and data files with another tool afterwards (e.g., with *Model Viewer - Main Window*)

The depth mesh is controlled through the Z Mesh groupbox. The thickness of the first slice (in meters) is specified in the First Depth edit line. The final depth (i.e., the maximum depth to use in the mode) is specified in the Last Depth edit line.

Specification of the mesh between the first and last depth is controlled in the Depths per decade list. This list will be automatically populated with a list of values. The length of this list is such that there is one value per decade of depth.

Once each of these values has been specified, hit the Generate Depths button to generate the Z mesh.

For example, for a first depth of 1 m and a last depth of 500000 m, the Depths per will be populated with 6 values. From top to bottom, they correspond to the number of layers used between depths of:

- 1-10 m
- 10-100 m
- 100-1000 m
- 1000-10000 m
- 10000-100000 m
- 100000-500000 m

In this instance, each value corresponds to the number of logarithmically spaced layers to use within each decade.

As a general rule of thumb, it is best to ensure that the sizes of the layers are always increasing. In accordance with this, the program will check the 2nd derivative of the generated mesh. If the derivative is negative anywhere, a message will appear saying so.

The backround resistivity of the model may be changed by editing the corresponding line and clicking the Set Background button.

### Reset Mesh

The Reset Mesh tab is used to generate a new, uniformly spaced mesh from scratch.

Set the nominal cell spacing for the X and Y directions in the corresponding boxes, and hit Regenerate Mesh.

A new mesh will be generated using these spacings, extending to the bounds set by the station locations.

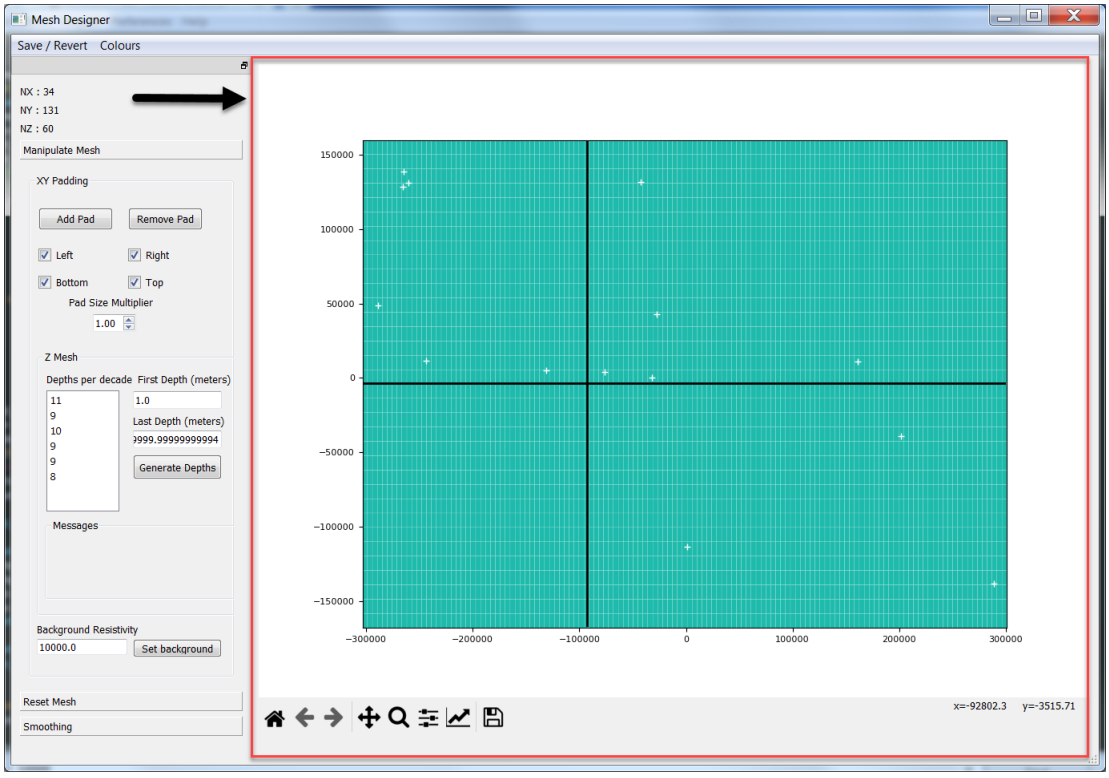
### Smoothing

The Smoothing tab is used to smooth the resistivity values of an inverted model.

- This tab is somewhat experimental at the moment *

Set the smoothing length in the X, Y, and Z direction using the corresponding spinboxes, and hit Smooth Model to apply a Gaussian smoother with those parameters.

### Plot Window




The Plot Window shows the current mesh, and if a data file was included, the station locations.

Fine grained modification of the mesh is done within this window.

Left click anywhere within the mesh to add a new vertical mesh line. Right click to add a horizontal mesh line.

Holding CTRL while left or right clicking will remove the nearest vertical / horizontal mesh line, respectively.

Refrain from double clicking within this window. Although some precautions have been implemented to avoid generated invalid meshes, double clicking can sometimes result in multiple mesh lines in the same location, i.e., a cell with 0 width, which will subsequently crash ModEM.

At the bottom of the Plot window is a toolbar. From here, you can pan and zoom into the plot, as well as return to the home view using the corresponding buttons.

Note that panning and zooming *is* stored within this window, which allows you to zoom into an area of high site density and add additional mesh:

- Click the zoom button and draw a rectangle around the area of interest.
- Unclick the zoom button to return the click functionality back to mesh modification.
- Modifiy the mesh as required.
- Hit the Home button in the toolbar to return to your original (un-zoomed) view.

CHAPTER

FIVE

# EXAMPLES

## 5.1 Recipes

All the following recipes assume pyMT modules have been imported as follows:

```
import pyMT.data_structures as ds
import pyMT.utils as utils
```

### 5.1.1 Adding Data to an Existing Data File

Adding new periods to an existing data file is supported through the *Data Plot - Main Window* GUI. However, it is assumed that the list file and data file used are consistent, i.e., that they have the same stations and in the same order. Therefore, adding new stations to an existing data file must be done using the pyMT API.

This example uses the *Data*, and *RawData* classes as well as the Data.get_data and Data.add_site methods.

Assuming the data file we want to modify is demo.dat, and the list file we want to pull in new sites from is raw/all.lst:

```
    data = ds.Data('demo.dat')  # Load in the data file
    raw_data = ds.RawData('raw/all.lst') # Load in the raw data
    # Get the new site to add with the same periods and components as 'data'
    to_add = raw_data.get_data(periods=data.periods,
                                                    components=data.components,
                                                    sites='test_1')
data.add_site(to_add['test_1']) # Add the new site to the data object
data.write('new_data', file_format='modem')  # Write the new data out.
```

Alternatively, multiple ModEM data files can be combined (e.g., append tipper data from one file into another), so long as resulting file still adheres to pyMTs requirements (i.e., uniform periods and data components for all sites).

### 5.1.2 Working with Inconsistent Periods / Components

Sometimes it may be desireable to invert a data set that has a non-uniform period set and / or component set, for example when mixing AMT and BB stations. If the number of wasted data points is low (maybe 3-4 frequencies total), it is usually simpler to just set high errors to the unavailable frequencies and let the inversion run as normal (the number of processors required is the same regardless, addition RAM required for wasted data points is low). Note, a wasted data point is a data point which is will be inverted at a subset of stations, but is unavailable at other stations (most often when mixing stations with different bandwidths, or stations with / without tipper) However, if you intend to invert many frequencies and / or components that would be unavailable at a subset of stations, there is a workaround. This workaround requires some knowledge of Python, and preferably a text editor that allows you to find, select, and remove all lines containing a particular string (e.g., Sublime Text).

First, you should set up the ModEM data file as usual (i.e., using Data Plot or j2ws3d). Make note of (or intentionally set up) any patterns you can use to distinguish your stations. In this example, broadband stations end in M, and AMT stations end in A. The final 4 periods of AMT stations, and first 3 periods of BB stations will be flagged and removed.:

```
data = ds.Data('demo.dat')  # Load your data set
for site in data.site_names:  # Loop through stations
        if site.lower().endswith('a')  # If its an AMT station
                for comp in data.components:  # Loop over available components
                        data.sites[site].used_error[comp][-4:] = data.REMOVE_FLAG  #␣
↪Flag the errors for the last 4 periods
        elif site.lower().endswith('m'):
                for comp in data.components:
                        data.sites[site].used_error[comp][:3] = data.REMOVE_FLAG  #␣
↪Flag the errors for the first 3 periods
data.write('demo_flagged')  # Write out a new 'flagged' version of the data file
```

This will create a new data file with the periods to be removed flagged with errors corresponding to the Data.REMOVE_FLAG (at the time of writing, it is 1234567)

Open this file in your text editor of choice. Find all instances of the REMOVE_FLAG, cut and paste the corresponding lines so they are all at the end of the file (this isnt required, it just makes life easier). Save this to demo_flagged.dat. Now remove the corresponding lines. Save this to a new data file (e.g., demo_removed.dat) You can now invert the demo_removed.dat. When using pyMT, you will still have to use the demo_flagged.dat version which has a uniform period band. Furthermore, when reading in the response file created by inverting demo_removed.dat, you will have to copy-paste the removed periods from demo_flagged.dat into it (this is why I moved the flagged lines to the end of the file).

I apologize for this incredibly roundabout method. Allowing for non-uniform data files in pyMT will require a large re-working of the code, and at this time it is more important to have working tools with some odd quirks than to have fully featured tools with game-breaking bugs.

## 5.2 Scripts

A number of scripts come with pyMT, although only a few of these are ready for *General Use* . Others will need to be manually edited for your particular needs. Only a few will be described here.

### 5.2.1 General Use

The scripts listed here are ready to run out of the box, and either have command line inputs to control them, or are so simple as to not require any inputs.

#### j2ws3d

j2ws3d is a command line tool for preparing data for inversion. It is included in your search path upon installtion of pyMT, so it can be run from anywhere (preferably wherever your raw data is).

Note, that by default j2ws3d.py tries to create both models and data through command line inputs, but the model creation aspect is bugged and will crash. Either run:

```
j2ws3d.py -data
```

to initial a data only mode, or just ignore the program crash (as it occurs after the data file is written out).

Other than that, just follow the on-screen prompts.

This program has is still usuable, but has been superceded by the *Data Plot - Main Window* GUI.

### to_vtk

Script for converting site locations and models to VTK files (compatible with, e.g., Paraview).

Usage:

```
to_vtk.py
```

Follow the command line prompts to select your model and/or data files. You will also have the option to project your model and data into a particular UTM zone.

Note that any station / model rotations will have to be handled manually.

### ws2modem

Simple script to convert WSINV3DMT data files into ModEM 3-D data files. Usage:

```
ws2modem.py <ws_input_data> <modem_output_data>
```

CHAPTER

SIX

# OTHER INFO

## 6.1 Help

### 6.1.1 HELP

For feature requests, bug fixes, design suggestions, or anything else contact me at eroots087@gmail.com

## 6.2 Disclaimer

### 6.2.1 DISCLAIMER

This code base has been cobbled together over a few years. Parts of it were written to fulfill a specific need at a specific time and then promptly forgotten, while others were written when I was still figuring out the difference between a class and a method. While I have tried to address bugs as I come across them, this code is not unit tested (yet), and so is best used with an understanding of the the expected outcome is. There are some *known bugs*, but any further unexpected behavior can be reported.

### 6.2.2 KNOWN BUGS

#### General

- Text is occasionally printed to the terminal. This text is usually meant to convey some information about something unexpected, and the codes attempts to work around it. Occasionally text will be printed that was meant for debugging purposes, and has just not been removed. Generally, any text that is printed that isnt followed by a crash is fine.

#### Data Plot GUI

- **When only a *List File* is specified for the loaded dataset, a *Data* object will be initialized by taking logarithmically spaced p**
  - In this instance, the *Data Plot - Map Viewer* is not initialized properly, and no site locations are shown.
  - The workaround is to re-sort the station locations (e.g., sort by west-east then re-sort back to default) and click the forward or back button. These operations should get the stations plotting.
- When multiple datasets are loaded, the misfit table may not be properly updated to reflect the currently selected dataset.

- **Rotating the data / stations using the Azimuth box has a few associated bugs**
  - The station locations and annotations in the *Data Plot - Map Viewer* may not be accurate
  - Non-zero azimuths also do something with the ordering of the stations in *Data Plot - Map Viewer*, and cycling through stations with the *Forward and Back Buttons* changes the order on the map. Will try to fix this.
  - **The Azimuth editor is meant to be used for 3-D data, and rotates the data accordingly: Station locations are rota**
    - A consequence of this setup is that the GUI is not suitable to rotation and projection into 2-D.
- **The Write All Plots action sometimes crashes the GUI.**
  - Known instances of this are when you attempt to overwrite an open PDF file. A permission error is thrown and not caught, resulting in a crash
  - Occasionally, if the number of subplots in the final saved plot is different from that specified within the Plot Options tab, the next action taken can result in a crash.
- If more subplots are used than there are stations available, one station will be re-plotted in order to fill the unused subplots.
- If more than one dataset is loaded into the GUI, switching between datasets may not update the misfit table.
- Blank items in the *Data Component Table* are selectable. However, this results in the first available component being plotted instead, and so should not break the GUI.
- The RMS misfits as shown in the *Misfit Table* may differ slightly from what logged by ModEM as pyMT always applies an error floor to any data read in. ModEM does not explicitly use (or store) an applied error floor, and so the hard-coded error floors of pyMT (see *Error Floors*) may differ from those used in the inversion. As a result, if you used a lower error floor than those coded into pyMT, the misfits shown in this table will be lower.

#### Model Viewer GUI

- Hovering over the plots within the 2-D will show location and resistivity information about the cursors position. This seem to generally be correct, however sometimes the resistivity shown does not match the plot itself. Likely an issue with matching the cursors location to model cells near the edges of the model.
- **The 2-D transect plot shows left-right in the order that was clicked.**
  - This means that if you select points from north to south, the figure will plot from north on the left to south on the right.
- The 2-D transect plot does not respond to changes in the model trim. The workaround currently is to trim the volume, and then re-select the desired points for the transect plot.

## 6.3 Change Log

### 6.3.1 Change Log

- 23/04/20
  - Changed behavior of transect plotting in *Model Viewer - Main Window* such that it automatically plots and refocuses the GUI into the 3D view.
- 03/04/20

  - Changed Lock Axis behavior in *Data Plot - Main Window* to lock bounds to static values, defined in the Display Options menu.

- 30/03/20
  - Removed dependency on naturalneighbor. *Data Plot - Main Window* will now offer other options for interpolation. If naturalneighbor happens to be installed, this option will appear.

- 14/03/20
  - Fixed a few of the issues related to reading multiple data sets into Data Plot
    - Use the Recalculate RMS button in the Data Selection tab to refresh the *Misfit Table* after switching the data set.
  - Inversion type is detected from available components when loading only a list file into *Data Plot - Main Window*
    - This should fix the bugs related to *Data Plot - Map Viewer* not allowing plotting of induction arrows and phase tensors.
  - Fixed bug which stopped station locations from being plotted when using only a list file.
  - Added Coordinate System in the *Data Plot - Map Viewer*. Stations can be plotted in local, UTM, or lat/long. Note that which of these is available will depend on what data is loaded (e.g., a ModEM data file alone has no information about the geographic locations of the stations)
  - Added a JPEG menu in *Data Plot - Map Viewer*. This allows loading of a geo-referenced JPEG image into the background. So far I have only tested it with UTM referenced JPEGs (and its corresponding world file), but I dont see any reason why a lat/long referenced file wouldnt work.
    - Note that when plotting these background images, the Coordinate System needs to be set appropriately.
  - Added some documentation in *Recipes* outlining my approach to working with data with non-uniform periods and / or components.

- 01/02/20
  - Models can now be read into pystart files in *Data Plot - Main Window*.
    - If loaded, plan view slices can be plotted in Map View.

- 26/01/20
  - Can now update the RMS table with a button after changing the plotted dataset.
  - Plotting of imaginary tipper arrows.
  - Added a legend for induction arrows showing colours and reference lengths

- 23/12/19
  - Added option to set equal or auto aspect ratio in the *Map Window* of *Data Plot - Main Window*
  - Also added freezing of axis limits, so you can zoom / pan and keep the same view after changing what is plotted.

- 14/12/19
  - Fixed induction arrow plotting in data_plot so that un-normalized arrows are actually useable. Not thoroughly tested however.
  - Added option to specify a cutoff length for induction arrows. Arrows with magnitudes greater than this will not be plotted.

  - Added secondary phase tensor plotting as inner bars within the phase tensor ellipses (as in Hering et al., 2019)
  - Fixed bath2model script to properly specify ocean and air cells within the covariance file.
  - In the process of fixing and testing how covariance files need to be written.

- 28/11/19
  - Fixed a bug where ModEM data files would include elevations if data was read directly from EDI files (which would put the receivers underground)
  - Added a static value to the Data class Data.REMOVE_FLAG, which is meant to be assigned to data points you want removed from the inversion data file
    * Currently not functional with the GUIs, but can be used to assign recognizable error values to data points to be removed, which can then be removed manually
    * Will (eventually) add these things into the GUIs
      · For now, see *Recipes* for an example on how to assign the errors and remove the data points from a ModEM data file.
  - Added an option to write model to CSV file (accessible via the API only right now)
  - Added functionality to read / write 2-D ModEM models and data
    * Still buggy and less than ideal. Is youre data really that 2-D anyways?

- 10/11/19
  - ModEM data file read function now checks for sign convention and units
    * Will automatically convert to exp(-it) and ohms

- 01/10/19
  - Added a script to add oceans and topography
    * Still experimental - model seems to be built correctly, but covariance file needs corrections
  - To fit above, changed default behavior of data file writing:
    * By default, elevations will not be used (i.e., Z = 0 for all stations). Add use_elevation=True as a named parameter in your write to include elevations
    * Note that due to a bug, previous versions may have included elevations in the written ModEM data files which could lead to spurious results.

- 09/09/19
  - Need to add these to the docs changelog when I get that fixed
  - Added turbo and turbo_r to colour maps
    * Should automaticaly be working in Model Viewer, not yet in Data Plot
  - Changed the way ModEM data files are read in to allow for arbitrary ordering of the data lines
    * This seems to be working without complaint, but may have some unintended side effects!

- 08/08/19
  - Some changes to IO to start to allow different periods for different sites (not fully implemented yet)
  - Some bug fixes related to reading data files

- 30/07/19

  - *Model* class can now read and write model covariance files
  - *Mesh Designer - Main Window* will automatically prompt for covariance file output when writing a model.
  - Added documentation for some of the more usuable scripts.
- 23/07/19
  - Re-release of pyMT onto GitHub
  - Now with (some) documentation!
    * See the pyMT/docs folder for a PDF version, or pyMT/docs/build/html/index.html to load up a browser version (complete with navigation bar and search tool)

## 6.4 License

### 6.4.1 LICENSE

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers and authors protection, the GPL clearly explains that there is no warranty for this free software. For both users and authors sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

This License refers to version 3 of the GNU General Public License.

Copyright also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

The Program refers to any copyrightable work licensed under this License. Each licensee is addressed as you. Licensees and recipients may be individuals or organizations.

To modify a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a modified version of the earlier work or a work based on the earlier work.

A covered work means either the unmodified Program or a work based on the Program.

To propagate a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To convey a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays Appropriate Legal Notices to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The source code for a work means the preferred form of the work for making modifications to it. Object code means any non-source form of a work.

A Standard Interface means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The System Libraries of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A Major Component, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The Corresponding Source for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the works System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the works users, your or third parties legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Programs source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to keep intact all notices.

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a

storage or distribution medium, is called an aggregate if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilations users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A User Product is either (1) a consumer product, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, normally used refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

Installation Information for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product

in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

Additional permissions are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered further restrictions within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An entity transaction is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the partys predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A contributor is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributors contributor version.

A contributors essential patent claims are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, control includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributors essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a patent license is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To grant such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. Knowingly relying means you have actual knowledge that, but for the patent license, your conveying the covered work

in a country, or your recipients use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is discriminatory if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License or any later version applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxys public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR

IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

CHAPTER

SEVEN

# INDICES AND TABLES

- genindex
- modindex
- search