**MTcode Manual**

[1 Introduction and Overview 1](#_Toc27654237)

[1.1 Before You Start 1](#_Toc27654238)

[1.2 Data Analysis and Inversion 1](#_Toc27654239)

[1.3 Time Series Processing 2](#_Toc27654240)

[1.4 Tensor Decomposition 2](#_Toc27654241)

[1.5 Occam 1-D Inversion 2](#_Toc27654242)

[1.6 View Mackie 2-D NLCG Inversion Results 2](#_Toc27654243)

[1.7 Miscellaneous 3](#_Toc27654244)

[2 Sample Workflow for 3-D MT Survey 3](#_Toc27654245)

[3 Code Structure 4](#_Toc27654246)

[3.1 Load Functions 4](#_Toc27654247)

[3.2 Data\_Plotting and Model\_Plotting 5](#_Toc27654248)

[3.3 MT\_Analysis 6](#_Toc27654249)

[3.4 Utils 6](#_Toc27654250)

[3.5 Write Functions 7](#_Toc27654251)

[3.6 Data\_Processing 7](#_Toc27654252)

[3.7 S2D 7](#_Toc27654253)

[3 Bug Fixes or Additions 7](#_Toc27654254)

## 1 Introduction and Overview

The software includes all MATLAB programs related to the MT workflow including time series processing, data analysis, data editing, and viewing inversion models. It does not include 2-D or 3-D inversion source code. These must be run external to MATLAB. Important scripts are highlighted. The majority of these scripts have been developed by the Magnetotelluric Research Group at the University of Alberta lead by Martyn Unsworth. Time series processing uses Fortran executables from Egbert and Booker (1986), tensor decomposition uses programs from McNeice and Jones (2001), and several scripts in Utils are taken from the MATLAB file exchange.

### 1.1 Before You Start

The software has only been thoroughly tested for Windows running MATLAB 2016a and MATLAB 2020a. The scripts also require specific MATLAB toolboxes which are not included in the standard MATLAB release. Specifically, the Statistical toolbox and the Mapping toolbox are required. To avoid problems, ensure that you have these toolboxes and are running MATLAB 2016a or 2020a.

Many of the scripts require the user\_defaults.m script. This script contains all the useful user inputs such as color bar axes, period limits, map projections, data conventions, interpolation methods, etc. This script should be on the user’s path. In practice, it is probably useful to copy the script to your project directory because different projects will have different defaults. For example, a long-period MT survey over a regional area will have different period limits and model plotting defaults than a small broadband survey. It will be necessary for each new user to update this script. **Note: the name of this script cannot be changed because it is a function which is called by other scripts.**

Type “help [function\_name]” for any MATLAB function to get information about how to use the function.

### 1.2 Data Analysis and Inversion

There are three high-level programs related to the MT workflow which rely on many of the same underlying structure. **These scripts are the ones most often used by our group in day-to-day work:**

1. MTplot: View, Edit and Analyze MT Data
   * Load data from a single EDI or interpolate all EDIs in the folder onto a common set of frequencies
   * View apparent resistivity and phase data, phase tensors, edit data, D+ curves, etc.
   * It is recommended to first edit each EDI and save the output as an edited EDI and then interpolate all the EDIs onto a common frequency set for inversion.
   * Output after Interpolation: Matfile (\*.mat) containing variable “d” with all survey information
2. M3D: Prepare data and build model for inversion
   * Load data from matfile, make model mesh, set error floors, and save data and model file for ModEM (or WSINV3D)
   * Output: ModEM or WSINV3D model and data files
3. S3D: View 3D inversion model results
   * You can load ModEM or WSINV3D inversion results. You will need the input data file, the input model file, the output inversion data response and the output inversion model. You will also need a log file (ModEM) or startup file (WSINV).
   * Once data and model are loaded, you can view cross-sections, view data response, misfit statistics, edit the inversion model, etc.

### 1.3 Time Series Processing

The MTcode folder also contains all the time series processing in MTcode\Data\_Processing. There are two types of MT instruments each with its own stand-alone processing software:

1. Phoenix uses mtu\_batch (see MTcode\Data\_Processing\Phoenix)
2. NIMS uses nims\_batch (see MTcode\Data\_Processing\NIMS)

These scripts are very different in structure from MTplot, M3D and S3D. They also rely on Fortran executables. **There is additional documentation on the Phoenix and NIMS processing in their respective folders**. The output from both of these time series processing programs is EDI files containing MT data which can be used in MTplot and elsewhere.

### 1.4 Tensor Decomposition

Tensor decomposition is often used to analyze the dimensionality and directionality of MT data. This is a stand-alone script written by McNeice and Jones (2001) called run\_strike. This is a stand-alone set of scripts saved in MTcode\MT\_Analysis\Tensor\_Decomposition. The program requires EDI files to run. **There is additional documentation about run\_strike in this folder**. To view results from tensor decomposition, use mj\_pcolor\_vxx. Note that permission is required to use this code from Alan Jones.

### 1.5 Occam 1-D Inversion

A 1-D inversion is available using OCCAM1D. This script is saved in MTcode\MT\_Analysis\Occam1D. This program uses an EDI file to run an inversion, or the user can build a model and run an inversion using synthetic data.

### 1.6 View Mackie 2-D NLCG Inversion Results

A 2-D inversion is available using an external Fortran executable that requires a license. The model must be constructed using another external program (WinGLink). These programs are not included in MTcode folder. However, once the inversion is completed, the results can be viewed in MATLAB using pmv20\_iso (or pmv20\_aniso for anisotropic models). The intention is eventually to get rid of pmv20 and replace it with S2D which uses the same structure as S3D.

### 1.7 Miscellaneous

There are also some additional miscellaneous scripts that have not yet been incorporated into other high-level scripts. These programs usually require more inline user input and are often for specific applications or purposes. This currently includes convert\_edi\_spectra\_imp, load\_model\_geozd, load\_model\_ubc, add\_discontinuity, edit\_covariance, plot\_model\_topography, dplus\_detailed\_statistics, and frequency\_analysis. Type “help” in MATLAB followed by any of these for more info.

## 2 Sample Workflow for 3-D MT Survey

1. Use nims\_batch to process long-period MT survey collected using NIMS instruments
   * Edit nims\_settings.m, nimsbatch.txt, and the survey.clk file to suite your survey
   * Final output is one EDI file for each MT site
2. Use MTplot to view and edit single EDI files
   * Go to folder containing EDI files and load single EDI
   * Click “edit data” and remove bad points and save edited EDI files
     + Note: Good idea to check for bad outliers that go beyond plot axes by using D+.
   * (Optional: Use run\_strike to do tensor decomposition on edited EDIs)
   * (Optional: Use OCCAM1D to look at 1-D inversions of edited EDIs)
   * (Optional: Use frequency\_analysis.m to examine frequencies of dataset)
3. Use MTplot to interpolate all EDI files onto a common set of frequencies
   * Go to folder containing edited EDI files
   * Click “interpolate all EDIs in current folder”
     + Generally preferable to “interpolate periods from file” using a txt file containing a list of frequencies
     + Interpolating will automatically save a \*.mat file containing a “d” structure with all data.
   * Assess data by looking at curves, phase tensors, polar plots, tipper, dimensionless parameters etc.
4. Use M3D to build 3-D inversion mesh
   * “Load the \*.mat data file” which was saved automatically after interpolation
     + (Optional and rare: Rotate stations and data)
   * “Generate Model” with x and y cell width, padding cells, etc.
     + (Optional: Add topography)
     + (Optional: View and edit model)
     + (Optional: Fix ocean and set ocean resistivity)
   * Specify the periods you want to use for the inversion, which data type you want, error floors etc. and then “View Data”
   * Save files to desired format (usually ModEM data, model and covariance)
   * Save M3D parameter file so that you can re-load it later to recreate all the parameters you used
   * There are many guidelines for generating a mesh, data set, etc. In general, it is a good idea to start simple. Start with no topography, no data rotations, no tipper, and no ocean.

## 3 Code Structure

MTplot, M3D and S3D (and their dependencies) are organized around basic modular “structures” in a standardized format. MATLAB uses structures as a convenient way to organize variables in the same way that folders organize files.

In general, any set of MT data and models require the same basic elements. There are certain elements which are part of the “data” such as impedance, period list, site name, apparent resistivity, phase, tipper, data errors, site location, etc. Similarly there are certain elements which are part of the “model” such as cell thickness, cell center locations, resistivity matrix, model origin, etc. When loading a set of data (e.g. a single EDI file, a bunch of EDI files, or a ModEM data file) it is helpful to put all the data into one structure (“d”) which contains all the necessary information for that dataset. Once this standardized structure is loaded into the MATLAB environment, then the same functions can be used to manipulate or plot that data set regardless of whether the data came from an EDI or an inversion file. The same philosophy applies to loading models.

The scripts are organized into different folders. These are outlined below.

### 3.1 Load Functions

The “Load\_Functions” folder contains all the functions to load data or models. The inputs for these functions are filenames and the outputs are “d” or “m” data structures. For example:

m = load\_model\_modem(‘halfspace.model’);

This command will load a ModEM model with the name “halfspace.model” which is in the current directory (or on the MATLAB path). The output is “m” which is a model structure containing all the necessary variables related to that model. This “m” structure is in a standardized format. The axis is such that “x” is related to the north-south direction and “y” is related to the east-west direction. The origin of the model is referenced to the *top, southwest edge* of the model. Different variables within the “m” structure can be accessed by typing “m.variable\_name”. Examples:

m.A; %The resistivity matrix is called “A”

m.cx; %The mesh cell centers in the x-direction (north-south)

m.y; %The mesh edges in the y-direction (east-west)

A full list of variables in the “m” structure can be found by typing:

m

in the MATLAB command window.

Similarly, a data structure can be created by using similar “load” functions where the input is a file name. For example:

d = load\_data\_edi(‘MT\_site.edi’);

This command will load a single EDI file with the name “MT\_site.edi” which is located in the current directory (or on the MATLAB path). The output is “d” which is a data structure containing all the information related to the data in that EDI. The data structure format is very specific. For example, impedances (d.Z) must be in the following format: frequencies in rows, tensor component in columns and sites in the third dimension. In other words d.Z has dimensions (nf x 4 x ns) where nf is number of frequencies and ns is number of sites. If data does not exist at a particular frequency or component or site, the entry is filled with NaN. The vector of periods (d.T) must be ascending. The latitude and longitude coordinates are organized into an (ns x 3) matrix with the first column containing latitudes, the second containing longitudes and the third containing elevations in meters above sea level. Everything is very specific and if data are loaded into the data structure incorrectly, it will cause errors later on. In general, the “d” structure is in e^{-iwt} format such that the yx phases are in the -90 to -180 quadrant. Impedances are in SI units.

If there is an error in loading data or model, then the bug fix can be found and corrected in these load functions and other functions are left untouched. It is also nice because if there is ever a new inversion package or new data format or new model format, all we need to do is write a function to load this new file into the standardized data or model structure and then all the other codes are automatically compatible.

The Load\_Functions folder also contains one other important function: *link\_model\_data.m.* In general, data are in latitude and longitude space whereas models are in x-y UTM space. It is necessary to link them up so that data can be plotted in model space and models can be geo-referenced in latitude and longitude. This code takes care of that. If there are bugs in coordinate conversions they should be found in this function.

### 3.2 Data\_Plotting and Model\_Plotting

These two folders contain all the plotting functions. Generally, the data plotting functions take “d” as an input and output a plot (or series of plots). The model plotting functions take “m” as an input and output a plot (or series of plots). Sometimes the function also requires a few additional arguments (like the index of the site to plot). The model plotting functions often take both “m” and “d” as input arguments but the “d” input is usually optional and is only required for plotting the site locations on the model. Note: If you include the “d” input argument in these model functions, it is usually necessary to run link\_model\_data first to reference the model and data to the same space.

All these functions are wrapped into MTplot and S3D but, because the code is more modular, it is possible to just write a short script to view things without using S3D as well. For example, suppose you want to view a diagonal slice without going through S3D. This can be done in 4 lines:

m = load\_model\_modem(‘inversion.model’); %loads model

dobs = load\_data\_modem(‘observed.data’); %loads observed data

[m,dobs] = link\_model\_data(m,dobs); %geo-references model

plot\_diagonal\_section(m,dobs); %plot diagonal slice

### 3.3 MT\_Analysis

This folder contains all the real calculations necessary to do a lot of the plotting such as calculating phase tensors, rotations, D+, data interpolation etc. Generally, these codes are not as modular and require more complicated inputs (such as Z, rotation angles, period vectors, etc.). This folder also includes Alan Jones’ strike decomposition program, Enci’s Bostick transforms, and Ersan’s OCCAM1D inversion. These sets of scripts have not been incorporated into the “structures” of other scripts and exist as stand-alone functions.

Some of the more important functions here are:

interpolate\_edi: This code finds all the EDI files in your current directory and interpolates them onto a common frequency set. The output is a “d” data structure which is also saved in a matfile. This is the matfile which is necessary to load into M3D.

calc\_Z: This code takes apparent resistivity and phase and calculates impedance. Note that I am a little unsure of the error propagation. The error propagation is based on a PDF handout from Greg from several years ago.

calc\_rho\_pha: This script takes impedance and calculates apparent resistivity and phase. Same concerns about error propagation as above.

rotate\_d: This script takes a “d” structure and rotates all the data in it by the same rotation angle. Positive angles are clockwise. This function also calls rotate\_Z and rotate\_tip which are more specific functions which rotate individual data points at a specific site and specific frequency.

### 3.4 Utils

This is a catch-all folder of scripts which contains a mish-mash of necessary functions. There are a couple important ones to highlight:

mv: a very generic script which takes any number of matrices of any shape and converts it all to a single vector. For example, if A is N x M and B is P x Q, then the result of mv(A,B) is a vector of length NMPQ. I looked to see if such a script existed in MATLAB but, surprisingly, couldn’t find one.

set\_figure\_size: a function used all over the place to set a figure up with certain dimensions relative to the current screen size.

print\_figure: a useful function which can be used any time a figure is open to save the figure into a specific folder with a filename

statistics: a useful function which takes any observed data vector, predicted data vector, and error vector and spits out a structure which contains the rms, L2 norm, chi-square etc. An optional input is an alpha parameters (between 0 and 1) which can then be used to produce critical chi-square and rms values. This function is useful because it takes care of NaN values cleanly.

### 3.5 Write Functions

These functions generally take a “d” or “m” structure and output some file format such as an EDI or a ModEM data or model file.

### 3.6 Data\_Processing

This folder contains all the time series processing scripts for NIMS and Phoenix. These are stand-alone scripts and have their own stand-alone manuals which can be found in their respective folders.

### 3.7 S2D

This folder currently contains the Mackie\_NLCG scripts to plot 2-D inversion results. This is currently a stand-alone folder. Eventually, it would be good to incorporate the 2-D input/output files into the standardized structure formats.

## 3 Bug Fixes or Additions

I’ve started a running list of bug fixes and additions which I will update here with each version I send. New scripts added to the library are highlighted in blue. Any changes to user\_defaults.m is highlighted in yellow.

**2018-11-20**

* calc\_phase\_tensor: **bug fix** in calculating the phase tensor. There was a typo when calculating the xx component of the phase tensor. I was doing X(i,4)\*Y(i,4) when it should read X(i,4)\*Y(i,1).
* edit\_data: additional changes made to make the editing more stable. I included the ability for the code to output a matfile containing the data structure of the last edit so if the program crashes or something, you can restart from your last edit rather than starting from the beginning. I also included a whole host of additional if statements to avoid crashes and instead output warnings. For example, if all of your data are out of quadrant for a given mode, D+ no longer crashes but instead gives a warning.
* edit\_data: added the ability to choose stations from the map rather than cycling through all the stations.
* plot\_ks\_test\_map: new function which allows user to compute the Kolmogov-Smirnov test (KS test) for a data set and plot on a map. This function has not yet been incorporated into S3.
* dplus and user\_defaults: added the option to enter “0” as the u.dplus\_percent and this will then take the real Z errors as the errors in D+ rather than applying any error floor based on a percentage of |Z|.
* statistics: added an additional structure variable stats.r which includes the raw data residual

**2018-12-04**

* ViewMT is now renamed MTplot at Martyn's request
* plot\_rho\_pha\_pseudo: Phase pseudo-sections have corrected colorbar (0 = blue, 90 = red)
* user\_defaults: There is now a flag in user\_defaults to choose what type of file to output (eps, png, or jpg)
* MTplot: now includes menu option to plot rose histograms of phase tensor strike angle
* user\_defaults: now includes an option to set elevation colorbar limits when plotting SRTM topography (see plot\_stations\_map\_srtm).
* user\_defaults: now includes a "nskip" option to skip certain frequencies when plotting maps of frequency slices (e.g. if nskip = 3, then [1:nskip:nf] will go from the first frequency to the last frequency skipping every 3rd frequency)
* user\_defaults: now includes some rose diagram options (axis limits, plot inset on phase tensor map, etc.)
* frequency\_analysis: New function called "frequency\_analysis" lets you look at the frequency content of the EDIs in your survey. It is in the MT\_analysis folder. This is a stand-alone function and requires no inputs, you can just run it in MATLAB.

**2018-12-20**

* load\_data\_modem: Cedar spotted an error in load\_data\_modem when reading tipper data. Long story short, it came down to the fact that a loop was indexed wrong. This is an important fix if you are using tipper.
* load\_geoboundary\_file\_list: I've made it a bit more versatile so that if you have extra lines or characters at the bottom of your file, it won't crash anymore.

**2019-01-07**

* load\_data\_edi: A bit more general now. Sometimes EDI files have a space at the beginning of each line which would cause problems. Now, it doesn't matter whether there is a space or not.
* load\_data\_edi: A bug where the output variables from calc\_rho\_pha were in the wrong order. Fixed.
* dplus\_detailed\_statistics: New code added to MT\_Analysis folder called "dplus\_detailed\_statistics". See code for details.
* plot\_rho\_pha\_pseudo and plot\_rho\_pha\_map: Fixed colorbar limits
* load\_model\_modem: Edit o that x and y model coordinates are referenced to model origin rather than referenced to center of the mesh. Usually our codes are referenced to the center of the mesh so it doesn't matter either way, but this is more general. I was doing some synthetics where the center of the mesh was at (25000, 25000) and it was causing problems. This should also fix the +80000 problem that lurks around sometimes in a much more easier way. One line versus 60 lines...
* M3D\_v1: Can now load matfiles containing "d" data structures. This type of matfile is output from "interpolate\_edi". So the workflow would be to first run interpolate\_edi and then load that matfile into M3D\_v1. I think this is the last spot where the old "EEE"- and "ZZZ"-style matfile was used so we no longer use that. But M3D and MTplot are still compatible with both (there is an option to load a "legacy" matfile).

**2019-01-16**

* M3D\_v1: Modified the "view\_data" callback function. Cleaned up the plots to make them look nicer. Added a "% Error" plot in the bottom left corner to check that the error floor is working as expected (e.g. there should be no points with an error % less than the error floor). In the plot, this means that no points should be below the dashed line.
* load\_data\_edi: Added a check to the script to see if the EDI impedances are in field units or SI units and do the correction early on here
* write\_data\_edi: Edited the script so that the EDI file is output in field units instead of SI units. This means that the EDIs will match WinGLink EDIs
* plot\_diagonal\_section: Bug fix when finding sites to project onto a diagonal section. If there are no sites to project then it was automatically plotting the first two sites regardless of where they were in the array. This is also fixed in plot\_diagonal\_section\_smooth.
* plot\_isosurface, plot\_slice\_multi, and plot\_model\_3D: Make elevations in the z direction consistent (positive is elevation below sea level in all plots now). Make vertical exaggeration consistent (ve>1 stretches the plot in all scripts now).
* load\_geoboundary\_file\_list: Added extra entry to geofile to specify line width.
* plot\_geoboundaries: Made elevations in the z direction consistent (positive is elevation below sea level)
* load\_data\_modem: Now loads in the d.responses strings in the right order rather than sorting them alphabetically
* M3D\_v1 (save\_modem\_data\_Callback): When saving ModEM data files, write out tipper and impedances as NaNs based on user input (e.g. full impedance, tipper only, off-diagonals only, etc.)
* plot\_geoboundaries: Added MarkerFaceColor to be defaulted to the symbol
* plot\_rho\_pha\_map: Added set\_map\_projection to this function
* plot\_geoboundaries: Edited so that the L structure is taken as an input so that you only have to load the geofile list once to save time. The L = load\_geoboundary\_file\_list is now called externally throughout the scripts
* plot\_diagonal\_section\_3d, plot\_diagonal\_section, plot\_diagonal\_section\_smooth: Edited so that the stations are plotted on the cross section at the correct (projected) elevation rather than the true station elevation.

**2019-02-05**

* load\_model\_wsinv: Corrected the x-vector (m.x and m.cx) so that they run from negative to positive and flipped the x dimension of m.A. See note in the code.
* plot\_slice: Minor change to axis to incorporate u.xylims
* rose\_geog: Fixed so that the histograms are colored in correctly
* M3D\_v1 (save\_wsinv\_data\_Callback): Pass proper variables to this function to make it the same as save\_modem\_data\_Callback
* plot\_diagonal\_section and plot\_diagonal\_section\_smooth: Initialized d\_elev variable so that if your diagonal section doesn’t cut through any sites (within tolerance), the code doesn’t crash
* interpolate\_edi: Added d.origin to variable output so that the origin of the EDIs is the same as the origin of M3D data files.
* fwd\_MT1D: new code to forward model 1-D MT data.

**2019-02-26**

* grdread2.m: new code to read \*.grd files saved in Utils folder
* M3D\_v1 (add\_topo\_Callback): Modified so that you can load a \*.grd file using grdread2. Script is also modified to be able to incorporate bathymetry if the grd file contains bathymetry.
  + Note: It is assumed that any elevation below zero is ocean. Ocean resistivity can be set in “Ocean Model” box. Default is 0.3 Ωm.
  + Note 2: The coastline of the model may not precisely match the true coastline because of the interpolation and large cell sizes. There may be some ocean cells “stranded” inland due to the interpolation.
  + Note 3: If you are saving a covariance file, make sure to check “Fix Ocean” if you want to fix it
  + Bathymetry Grid files can be downloaded from <https://www.ngdc.noaa.gov/mgg/global/global.html>
  + <https://maps.ngdc.noaa.gov/viewers/wcs-client/>
    - Select area from map
    - Export as GMT NetCDF export format
* plot\_diagonal\_section: Modified so that stations are plotted at the correct distance along the profile to correspond with the topography at that model cell. This ensures that stations are projected onto the slice correctly so that they are plotted on the projected elevation surface even if they are projected from very far away.
* M3D\_v1 (add\_topo\_Callback): Modified the x-y utm conversion of topography coordinates to match link\_model\_data so that the topography matches the mesh. This was only an issue when using very large gridfiles that cover large areas.
* load\_edi\_data: Added some additional “or” statements for the tipper error (TXVAR) to be more general so that it is compatible with TXRVAR and TXIVAR error formats.
* plot\_phase\_tensor\_ellipse\_pseudo: Corrected the scaling of the ellipses so that circles always plot as circles no matter how the figure is stretched.
* interpolate\_edi: If the user supplies a frequency text file with frequencies beyond the minimum and maximum frequencies of the EDI dataset, then those frequencies are excluded rather than interpolating NaNs.
* write\_data\_modem: Fixed so that tipper header is written correctly when including tipper + impedance or tipper only data files.
* M3D\_v1 (save\_modem\_data\_Callback): Fixed so that tipper header is written correctly when including tipper only or tipper+impedance ModEM data files.
* compare\_models: Corrected to update dx, dy, and dz for the interpolated mesh structures.
* M3D\_v1 (generate\_model\_Callback): Added the option to enter a vector of thicknesses into the “first thickness” box to manually specify the thicknesses. If you enter a vector of thicknesses, then it takes that as the vector and does not use “Number of Layers” or “Increase Thickness By” entries.
* write\_data\_edi: Renamed this function from “write\_edi” so that it is more consistent with the other function names.

**2019-04-08**

* M3D\_v1 (add\_topo\_Callback): Updated so that the longitude and latitude vectors for the topography are updated when adding more SRTM tiles. Previously, only the x/y UTM vectors were being updated.
* calc\_determinant: new function added to MT\_Analysis to calculate the determinant of the impedance (a rotational invariant).
* edit\_model: Updated edit\_model to output the edited indices (ix,iy,iz,rho) for all options
* edit\_covariance: new function added which allows user to load indices to edit covariance file

**2019-05-07**

* edit\_data\_tipper: Added a new function to edit tipper data. Identical in structure to edit\_data.m
  + Implemented in MTplot
* edit\_data and edit\_data\_tipper: Modified both of these so that if you remove a point from impedance, it *does not* remove that same point from the tipper and vice versa.
* plot\_misfit\_ks\_test\_map: Edited so that the code is more general. It can now handle datasets with the different lengths of vectors (but will not make cross plots if they are different lengths)
* plot\_misfit\_cross\_plot: New code which makes a cross plot of the observed and perturbed residuals. This is used in plot\_misfit\_ks\_test\_map.
* plot\_misfit\_convergence: Added an additional “L-curve” plot to this script which shows which inversion iteration is nearest to an rms of 1 and a model norm of zero.
* plot\_induction\_vector\_map: Added a negative sign to the induction vectors to make them match the convention of Ersan’s original MTplot. I am not sure why the negative is required.
* Deg2dms, deg2utm, and dms2deg: Added all three functions to the Mapping utilities folder for completeness. These functions are only called in mj\_pcolor as far as I can tell.
* calc\_fwd\_1d: New script to compute 1D impedance data from a synthetic model.
* dplus: corrected so that the error floor is being applied correctly rather than re-writing the errors
* dplus\_detailed\_statistics: corrected so that the output is the predicted D+ data rather than the original data.

**2019-05-16**

* Major changes to the database. I copied over all the “other” scripts that we use such as the Tensor Decomposition scripts (run\_strike, mj\_pcolor, etc.), Data Processing scripts for NIMS and Phoenix, Time Series viewers (tsplot and ts\_simple\_view), Bostick Mapping, and 2D Inversion (pmv)
* I also copied some miscellaneous scripts such as readEDI\_spectra and convert\_edi\_spectra\_imp (both saved in Load\_Functions), shear\_twist\_anis (saved in MT\_Analysis), and gen\_jfile\_x (saved in Tensor\_Decomposition).
* Plot\_slice\_multis: Corrected bugs to axes when plotting multiple vertical slices.
* Frequency\_analysis: Edited to include the ability to load any d structure
* Plot\_phase\_tensor\_beta\_strike and plot\_phase\_tensor\_map: Edited so that figures are saved sequentially with proper filenames
* Plot\_model\_topography: New code to plot the model topography
* Calc\_phase\_tensor: When calculating alpha and beta, it now uses atan2 for both so that the tangent appears in the correct quadrant. This ensures that the phase tensor ellipse will always align with the strike (rather than 90 degree ambiguity).
* Calc\_phase\_tensor: The calculation of phimin and phimax was incorrect (Eq A5 and A7 from Caldwell). The phase tensor was constructed with swapped off-diagonals.
* Calc\_phase\_tensor: The calculation of the phase ellipse was incorrect. We were plotting the transpose of the phase tensor ellipse. A very tricky bug because most phase tensors are approximately symmetric so it is not noticeable.

**2019-08-12**

* Link\_model\_data: Moved around the if statements so that if there is no x and y utm coordinates in the data file yet, then the usual checks are not performed
* Load\_model\_geozd: Removed the x and y utm coordinates from the dummy data structure. If you want to include station location information, you will need to supply it separately.
* Load\_model\_geozd: Added “name” to model structure to match other model structures
* Compare\_models: Modified so that origin, x, y, z, and Z model structure variables are also updated
* Plot\_diagonal\_section: Updated so that clicked points are output to the Command Window in lat long rather than model coordinates.
* Link\_model\_data: There are some weird things that can happen in this code when you have very small cells (e.g. <50 m) because the ModEM output data files have the origin in lat/long to 3 decimal places. I’ve added some comments to the code and changed the display errors.
* Plot\_phase\_tensor\_ellipse\_pseudo: Bug fix to ensure that the pseudo-section ellipses exactly match the map view ellipses.
* Edit\_data and edit\_data\_tipper: Changes to the way whole periods and whole sites are removed from the dataset. Now a list GUI is used to select periods and sites instead of entering period indices or station names one-by-one
* Plot\_rho\_pha\_map\_determinant: New script to plot the interpolated determinant apparent resistivity and phase in map view. Similar to plot\_rho\_pha\_map.
* MTplot: Added plot\_rho\_pha\_map\_determinant as an option in MTplot

**2019-10-28**

* M3D: Bug fix when loading models with topography
* Load\_data\_modem : Bug fix when writing out datasets which have different numbers of frequencies and/or stations
* User\_defaults: Very important bug fix in user\_defaults. When checking geoboundary file, it was doing “fclose all” which interferes with other files. Now it does fclose(fid) to specifically close the geoboundary file only.
* Changes to nims.m, nims\_rotate\_in\_readnims, and nims\_rotate\_in\_zss2edi: Get rid of any reference to plot\_edi and replace with d structure.
* Plot\_tipper: Minor addition of an optional flag to set subplots correctly
* Print\_figure: Minor addition of an optional flag to override user defaults and automatically plot PNG
* Added Ben’s stand-alone Phoenix processing to the code library in MTCode/Data\_Processing/Phoenix
* Nlcg\_mdl\_iso: Commented out some lines that are specific to my project and not yet general
* Plot\_tipper: Added u.tiplim to axis bounds instead of hard-coded [-1, 1]
* Nlcg\_mdl\_iso: Commented out axis bound that I had previously used and made more general with ymin, ymax, xmin, xmax from scales.m
* Kml2geo: New script to convert GoogleEarth kml file to a geofile.txt for use with geoboundaries.txt in user\_defaults. Only works for placemarks currently (not paths)
* MTplot: Added a menu option to plot SRTM map so that it doesn’t plot everytime (this was especially annoying when loading a new survey and suddenly it starts downloading tons of SRTM tiles even if you are only wanting to quickly look at some data).
* Plot\_misfit\_rms\_by\_period\_map: Added u.rmsscale to have user input for the size of the circles
* Plot\_cross\_section\_3D: updated so that the vertical sections plot the xylims from user\_defaults as well
* M3D: Added ability to input data from a ModEM data file. Streamlined some of the code when importing models, and implemented load\_model\_wsinv. And now you must import data before importing a model.
* Write\_model\_wsinv: There was an inconsistency between x and y coordinates (NS and EW). This has been corrected.

**2019-12-19**

* Edit\_model: Added some additional options to output the edited indices without saving the model
* OCCAM1D: Added the 1D Occam script to the MTcode library. Code was originally written by Ersan Turkoglu
* Apply\_errorfloor.m: Added this simple function to MT\_Analysis folder. This function is currently only implemented in OCCAM1D but could (and should) be implemented in other scripts like M3
* Add\_discontinuity: A new script which allows user to add discontinuity to ModEM model. This is not user-friendly in that each block must be run individually. Also, the discontinuity surface needs to be coded up manually by the user because each discontinuity dataset is project specific and from a variety of data sources (e.g. xyzd, ArcGIS, GRD files, etc.)
* M3D: Ben made major changes to the script to make the functions external to the GUI code. All these new functions are in the M3D folder. Also some changes to buttons (to make the workflow more logical). Added option to delete layers and smooth topography. Normalized button positions so you can resize the window.
  + M3\_view\_data: Bug fix so that the plot still works when using NaN errors
  + Option to rotate data and mesh to set a new x-direction in the inversion. First you rotate the data and then you rotate the mesh. Due to projection issues, this may not work for large data sets like the AB/BC grid.
  + Added yx mode to error floor plot in view data and added info about number of data points where error floor is applied
* Kml2geo: Edited so that it can take both points and lines as input.
* User\_defaults: Very important little bug fix in user\_defaults so that if you do not have a geobound file, the check on it does not crash
* S3D: Added compatibility to load anisotropic inversion model files
* Plot\_induction\_vector\_map: Removed unnecessary for loop

**2020-02-07**

* Plot\_diagonal\_section: added an option to do an interpolation over the diagonal section to make it smoother. This is arguably better than both the current options (line or smooth) but does not plot the “true” model but rather an interpolated version of it
* User\_defaults: Changed to add additional option for plot\_diagonal\_section. “interp” is now the default.
* plot\_rho\_pha\_pseudo: Added 4 lines on Line 119 to make the “rho\_pha” folder prior to outputting the TE and TM data.
* Edit\_model: Cleaned up some of the if statements so that if user hits “cancel” or closes a window it always returns to the previous menu rather than crashing
* Edit\_model: Added option to replace values in a polygon directly or only values less than some resistivity
* Plot\_isosurface: Corrected z indices so that if you plot a new isosurface with different z limits, the z axis remains unchanged
* M3\_view\_data: The errors were being modified in the wrong way. The code takes the sqrt(EEE) but EEE is complex. It should be taking sqrt(real(EEE)) and sqrt(imag(EEE)) separately. The effect is relatively small as to be negligible but it is corrected anyway. This ensures that errors (and error floors) are being calculated correctly relative to Z.
* Add\_noise: Based on discussions with Ben, I think noise for a given component should not be added relative to sqrt(Zxy\*Zyx). It mixes the modes and means that if there are large differences between the two off-diagonals, the relative noise (and relative error) can be really large for one off-diagonal component compared to another. It seems better to add noise to each component relative to that component. However, this can also make the diagonals have very small amounts of noise which may not be realistic since real data generally has very noisy diagonals. One solution is to define the error for each component relative to the off-diagonal component with matching E-fields. So the error for Zxy and Zxx is defined relative to Zxy and the error for Zyx and Zyy is defined relative to Zyx.
* M3\_save\_modem\_data: Removed option to add noise when saving in M3. This option is basically never used (nor should it be) unless doing synthetics. Probably better to develop a different script for building synthetics.
* Add\_rho\_colorbar: Added function for colorbars. And also added functionality in user\_defaults to specify label format for colorbars as (1, 10, 100, 1000) or (0,1,2,3). This function is now incorporated into a variety of model plotting (and rho pha data plotting) scripts.
* User\_defaults: Change to user\_defaults to incorporate colorbar labels.
* Plot\_rectangular\_fault\_plane: Script to plot a rectangular fault plane in 3-D space given a strike, dip, width, length and location. Not incorporated into any existing scripts. Stand-alone feature in Utils/Figures
* Profile\_topo: Added a script to plot a 2-D topography profile from some specified start point to end point using a supplied grid file. Not incorporated into any existing scripts.
* MTplot: Changed file naming structure for rho\_map figures so that each map is labelled sequentially

**2020-03-21**

* MTplot: Enabled D+ when viewing individual stations
* Plot\_rho\_pha and plot\_impedance: Fixed title to include underscores in station names
* M3D\_smooth\_air: Ben made some changes to how the first non-air cell topography gets smoothed to be less glitch
* Load\_model\_ubc: Corrected d.origin to properly read in the mesh\_corner variable instead of having some hard-coded lat-long.
* Plot\_isosurface: Incorporated xylims into isosurface
* Make\_synthetic\_site\_locations and make\_ones\_data: New scripts to generate site locations and “dummy” data when doing synthetic inversions. Currently no ability to add synthetic topography.
* Edit\_data: Added ability to add noise from this set of menus
* Dplus\_plot: Added function to plot full D+ results including spectral function and D+ model
* Load\_data\_2DNLCG: Made function able to load data with different frequency sets on tipper and impedance
* Edit\_data: Added option to save edited data as text file
* Load\_covariance: New file to load a covariance file into a standard model structure with the model variable (A) containing the integer masks for each model cell. Zeros are set to NaN.
* Plot\_polar: Added figure saving output
* Plot\_polar\_map: Added figure saving output
* Plot\_misfit\_rms\_map: Added a code block to create a manual colormap for the RMS circles on the map. This code block is currently commented out because I’m not sure how to incorporate it cleanly into the script.
* Plot\_geoboundaries\_geoshow: New script to plot geoboundaries using geoshow. The original plot\_geoboundaries was only compatible with m\_map and x-y UTM plotting.
* Adapted edit\_data, edit\_data\_tipper, MTplot, get\_pseudo\_section\_indices, plot\_data, and plot\_stations\_map\_srtm so that they plot geoboundaries properly using geoshow
* M3\_generate\_model: Added check to make sure data are not all on cell edges (only happens for synthetic data)
* M3\_load\_mat\_new: Changed so that it can load x-y data directly (if x-y UTM data already exist) which avoids an unnecessary geo2utm conversion

**2020-04-17**

* Add\_noise: Function now adds noise to the tipper as well
* Plot\_polar\_distorted: Finally added this function to MTplot
* User\_defaults: New option called u.tplot specifically for plot\_polar\_distorted
* Plot\_polar: Made some changes to this so that the iteration through stations happens external to the function in MTplot
* MTplot: Added plot\_polar\_distorted. Also made the loop over stations in plot\_polar external and is now within MTplot rather than within the function
* Plot\_induction\_vector\_map: rose diagram for induction vectors is now enabled. See user\_defaults for settings
* Set\_inset\_position is now a separate function so that the same function is called for phase tensor and induction vector rose diagrams
* User\_defaults: Added option to have the rose diagram inset locations be specified manually as NW, SE, SW or NE, or an xy coordinate pair
* User\_defaults: Added option to skip every nth station when plotting station-related functions
* User\_defaults: Added plotting limits for 2D models (u.lim\_2D) only called with S2D
* Run\_strike and strike\_settings: Variable name change from “bounds” to “gb\_bounds” to avoid duplication with function “bounds” in MATLAB 2019.
* Fix\_pcolor: function to show all columns in pcolor pseudo-sections and to center stations above their respective columns. Called by functions that plot pseudo-sections
* Plot\_rho\_pha\_map: Added check for d structures that only contain one stations to return rather than crash
* MTplot: Added some checks to make sure the code doesn’t crash when only one station is loaded
* Plot\_phase\_tensor\_ellipse\_pseudo and plot\_phase\_tensor\_map: Ability to plot phi min on colored phase tensors instead of beta
* User\_defaults: Options for ‘beta’ or ‘phimin’ as fill for phase tensor ellipses
* Load\_data\_2DNLCG: error floors are now applied to the apparent resistivity and phase data. These are converted to a relative error that is meant to reproduce the misfit values from the inversion. When used with S2D, the misfit for each site is usually within 2 decimal places of the misfit in the .rsp file from the inversion.
* S2D: added more functionality to plot the model, data curves, and misfit statistics. Still a lot of work to do on this code though.
* Detailed\_statistics and detailed statistics\_2D: these functions now output one statistics structure called “s”. This is in order to differentiate between statistics calculated from impedance data (i.e. ModEM and WSINV3D inversions) and rho+phase data (i.e. Mackie 2D inversion). If using S3D or S2D to view inversion results, the appropriate function is automatically called.
* Load\_data\_wsinv: error floors are now read from the startup file and applied to the data
* Load\_startup\_wsinv is now a separate function called by S3D
* Interpolate\_edi and load\_data\_edi: changed how these functions check the impedance units for EDI files. Impedances should be in field units in the EDI files, but it is good to check. interpolate\_edi now checks all loaded EDI files to see if apparent resistivity values computed from impedances are consistent with field units or SI units:

If >= 50% of your stations are suspected to have impedance in field units (using the arbitrary rho > 10^6 check), interpolate\_edi automatically gives a flag to load\_edi to convert all impedances to SI.

If < 50% of your stations are suspected to be in SI units, a menu will ask what you want to do (I don't think this will come up very often). The options are:

   (1) convert only the suspected stations in field units to SI (assumes the 10^6 check was correct...!)

   (2) convert ALL stations to SI units (this implies all stations are actually field units)

   (3) continue interpolate\_edi without performing conversions (implies that all stations are already SI)

   (4) exit without doing anything

* Update to plotting geoboundaries. The new geoboundary file format requires users to specify if each file contains lines, markers, or fills. Load\_geoboundary\_file\_list, Plot\_geoboundaries, Plot\_geoboundaries\_geoshow, and Plot\_geoboundaries\_diagonal\_section have been updated to accommodate the new syntax.
* Work\_mtu\_aug29\_2019 is now the version of the Phoenix time series package in MTcode. This version contains updated documentation, the new IGRF geomagnetic coefficients, and other small fixes.

**2020-05-05**

* Bug fix in plot\_rho\_pha\_pseudo line 21
* Added function compare\_data to plot comparison of observed data and two sets of predicted data
* Added ‘hold on’ after calls to m\_patch and patch in plot\_geoboundaries and plot\_geoboundaries\_diagonal\_section
* Fixed a plotting issue causing edit\_data.m to crash if viewing a dataset with only off-diagonal impedances. 2 sub-functions are added to the bottom of the edit\_data.m file to identify axis handles for plotting.
* Added button to reload user\_defaults in main menus of MTplot and S3D. Now you can edit and save changes to user\_defaults.m in the MATLAB editor, and then reload user\_defaults without having to restart these scripts.

**2020-06-04**

* plot\_geoboundaries\_geoshow: updated to correctly plot fills. The function converts the coordinates of fills to clockwise order so that they can be correctly plotted.
* M3\_add\_topo: now .grd files can be used for topography with rotated meshes. Fixed bathymetry so that a layer at z = 0 contains seawater.
* M3\_view\_data: updated to show error floors applied to data variance. Previously, this function showed the error floors applied to standard error, which was inconsistent because ModEM assumes error floors are actually applied to the data variance. As a result, this function underestimated the percent of data errors being overwritten by the error floor.
* dplus: Noticed that it was taking abs(Zerr) to compute error floor but since Zerr is stored as complex-valued matrix but is, in reality, real-valued, we want real(Zerr)
* MTplot: Made changes to plotting stations one-by-one to turn D+ errorfloor on/off when toggling D+ on/off
* Manual\_legend: Added an explicit call to turn “AutoUpdate” legends off. After MATLAB 2017a release, the legends auto-update which is annoying. Most legend calls in MTcode use “manual\_legend”, so I think this will fix most of the issues. But if it appears elsewhere then you can turn it off using legend(label1,label2,…,’AutoUpdate’,’off’)
* Edit\_data: Made changes so that D+ error bars are plotted when D+ is toggled on. Also added option to change D+ error floor percentage. Added option to plot D+ residuals.
* load\_data\_edi : fixed so that the code correctly checks units with/without units flag
* interpolate\_edi: now ignores frequency gaps in data
* plot\_geoboundaries\_diagonal-section: added check for data structure before plotting. Bug fix for plotting no geoboundaries
* compare\_data: simplified code
* fix\_pcolor: changed so that pseudo-sections column edges are halfway between stations. Previously, stations were moved to the middle of the column but stations are not moved now.
* plot\_rho\_pha: now uses u.phalimdiag for diagonal phase limits
* rotate\_d: now adds rotation angle to d.zrot and z.trot
* get\_pseudo\_section\_indices: changed the way that profiles are defined for pseudo-sections. First, define a profile with midpoint and azimuth, then choose stations associated with the profile. See new settings in user\_defaults.
* Plot\_rho\_pha\_pseudo: edited to incorporate new way of defining profile. Added xx and yy data
* Plot\_tipper\_pseudo: edited to incorporate new way of defining profile
* Plot\_phase\_tensor\_ellipse\_pseudo: edited to incorporate new way of defining profile
* Plot\_phase\_tensor\_pseudo: edited to incorporate new way of defining profile
* Plot\_skew\_pseudo: edited to incorporate new way of defining profile
* Plot\_misfit\_residual\_pseudo: new function to plot pseudo-sections of impedance residuals and tipper residuals (if tipper data were inverted)
* S3D\_v1: updated menu to include options for tipper pseudo-section and residuals pseudo-section. Added option to plot horizontal slices in model coordinates (x,y)
* User\_defaults: new parameters for pseudo-sections, diagonal phase plot limits, and residual plot limits
* Calc\_rms: minor change to suppress warnings
* Statistics: minor change to suppress warnings

**2020-06-10**

* S3D\_v1:
  + Added menu button for compare\_data
  + Added menu button for plot\_misfit\_ks\_test\_map. This function performs the two-sample K-S test between (1) residuals from the inversion response and (2) residuals from a second response, e.g. predicted data for a perturbed resistivity model.
    - Added u.significance\_level parameter to user\_defaults
  + Added menu button for plot\_misfit\_cross\_plot. This function shows data residuals for the inversion response and a second response, e.g. predicted data for a perturbed resistivity model.
* edit\_data and edit\_data\_tipper:
  + added histogram of number of sites per period
  + changed default file name for saved files
  + removed “R” button to restart… not necessary. Just close and re-load the data file.
  + Added option in user\_defaults to edit all data at a particular period, or individual components
  + Fixed TE/TM labels
* Get\_pseudosection\_indices: fixed bug for Matlab version 2016a (and presumably earlier versions)
* Plot\_tipper\_pseudo: fixed bug - not printing text files
* Plot\_rho\_pha\_pseudo: fixed bug - not printing TE/TM data files
* link\_model\_data: updated to allow plotting rotated models in geographic coordinates with plot\_slice\_map.
* Load\_geoboundary\_file\_list: updated comments in code
* Added colormap parameter in user\_defaults called u.cmap\_name. You may enter the name of any MATLAB-embedded colormap, or use the custom colormaps: ‘mtcode\_default’ (similar to jet) or ‘mtcode\_red\_blue’. The red-to-blue colormap option is new.
* New function get\_projection called by plot\_geoboundaries to check that the map projection is initialized.
* Print\_figure: updated to allow user to print more than one type of figure (e.g. .png and .eps files) without having to run a plotting function twice. Use a cell array for u.output\_figure in user\_defaults to enable this feature.

**2020-06-19**

* New function plot\_topo which plots SRTM topography data using either geoshow, m\_map or surf (specified using flags 1, 2 or 3). This function has been incorporated into many other data and model plotting functions.
  + User\_defaults has three new options: plot\_topo (true of false), topo\_file (to specify a topo file), and topo\_transparency (0<x<1).
* Also updated some of the help menus and added (C) copyright thing to some of the scripts to identify who wrote all this stuff and provide attribution where necessary.
* Print\_figure: Added check to see if user has “none” entered in u.output\_figure.
* Load\_model\_modem: Changed m.X and m.Y to be edges rather than cell centers
* Plot\_slice: Changed to plot cell edges rather than cell centers
* Plot\_mesh: new function to plot mesh independently from other functions
* Make\_synthetic\_site\_locations: Changed names of sites so that they are loaded sequentially (Site\_001 instead of Site\_1)
* M3\_generate\_model: Changed “unique” to “uniquetol” when checked for synthetic sites on model vertices
* S3D: Changed several titles of plots to match depth to top of layer rather than depth to middle of layer
* Edit\_model: Updated so that clicking and edits are referenced to cell centers rather than cell edges but depths are referenced to top/bottom of cell edges
* Plot\_misfit\_ks\_test\_map: added check for missing tipper data
* Link\_model\_data: edited ‘if’ statements for improved stability
* Plot\_diagonal\_section and plot\_diagonal\_section\_smooth: added menu option to load a text file with profile endpoints. Text file has same format as the one used to load pseudo-section profile endpoints (column 1: longitude, column 2: latitude).

**2020-06-30**

* Load\_data\_edi: Changed to convert variances to standard errors. The Zerr and tiperr variables in the d structure now are consistently storing only standard errors throughout.
* Load\_data\_edi: Cleaned up units so that the default is to always convert impedances from field units (assumed) to SI units. If someone has non-field unit EDIs they can use a new function convert\_edi\_units.m to correct it.
* Write\_data\_edi: Changed to write out variances (convert standard errors to variances).
* Interpolate\_edi: Cleaned up to avoid all the unnecessary checks on unit conversions.
* Convert\_edi\_units: New function to convert the rare EDI in SI units to an EDI file in field units. Saved in MTcode/Load\_Functions
* M3\_save\_modem\_data: Cleaned up to avoid unnecessary unit conversions
* M3\_view\_data: Cleaned up to avoid unnecessary unit conversions
* Edit\_data\_tipper: now only tipper data are masked by all editing options. This function does not mask any impedance data.