GAMS

Release 0.1

Eric Roots

Sep 02, 2021

CONTENTS

CHAPTER

ONE

GETTING STARTED

1.1 Overview

GAMS (Geoscience Analyst Magnetics Suite) is a lightweight GUI to read, visualize, and operate on magnetic data imported from Geoscience Analyst. The GUI is designed to import magnetic data from a Geoscience Analyst (GA) workspace, perform operations and transformations on the magnetic grids, and export the results back into GA for final visualization. Grid pre-processing options are given, and fast vectorized calculations using common python libraries allow visualization and selection of optimal settings in real time.

Grid transformations include:

- Zeroth-order analytic signal amplitude (ASA_0)
- First-order analytic signal amplitude
- Spatial deriavaties of ASA_0
- Tilt to transform data to pole and vertical dip[1]
- Zeroth-order local wavenumber
- Magnetic field at the pole with vertical dip[1]
- Apparent susceptibility[1]

Visual outputs from the pre-processing stages are also available to ensure the results are as expected.

1.2 Installation

1.2.1 Installing GAMS

Installation was tested using the Anaconda Python distribution for Windows (click here to download if needed). A python compatible terminal is also needed (e.g., git-bash, Anaconda Prompt, Windows Powershell, etc.). Ensure that python - and conda, if needed - are on the path for the terminal you are using. Note that the standard Windows co

If using Anaconda, you may want to install GAMS into a separate Anaconda environment. A new environment can be created using, for example:

```
conda create -n gams pip git
```

[1] For details, see: Smith, R. 2021, Transformative geophysics: Alternatives to the reduction-to-pole transformation of magnetic data. Third Australian Exploration Geoscience Conference Extended Abstract. pp 1-4.

You can then switch to that environment using:

```
conda activate gams
```

Or, if using git-bash:

```
source activate gams
```

Note: By default, new terminals start in the 'base' environment, so you will have to enter the above command each time, or add it to your ~/.bashrc file to have it run automatically. See the the Anaconda Environemnent documentation for more details.

GAMS can then be installed with pip and git using:

```
pip install git+https://github.com/eroots/gams
```

Alternatively, experienced git / conda users may manually download the repository (https://github.com/eroots/gams) using the green 'Code' button, or by cloning the repository using a git-bash terminal. After downloading the code, the package can be installed by navigating to the GAMS folder and running (in your git + python compatible terminal):

```
python setup.py install
```

Or, if you intend to modify or extend the code, use:

```
python setup.py develop
```

1.2.2 Updating GAMS

Any updates to the code can be downloaded and installed using the pip upgrade option:

```
pip install --upgrade git+https://github.com/eroots/gams.git
```

Or if you used the second option (manually downloading + installing), you can use Git to pull the updates (while in the main gams directory):

```
git pull origin main
```

1.3 Geoscience Analyst

Geoscience Analyst (a free geoscientific data viewer from Mira Geoscience) can be downloaded from:

https://mirageoscience.com/mining-industry-software/geoscience-analyst/

Unzip the downloaded folder, run the setup executable, and follow the on-screen instructions to install the viewer. The data you will GAMS will be working with must already exist in Geoscience Analyst and be saved in a workspace (.geoh5 file).

1.3.1 Setting Up a Workspace in Geoscience Analyst

The GAMS GUI imports grids from saved Geoscience Analyst workspaces (.geoh5 files). For new users, once Geoscience Analyst is launched magnetic grids can be imported into the workspace through drag-and-drop, or by using the top menu bar (e.g., for Geosoft Grid files: 'File –> Import –> Geosoft –> Grid files'). Once the grid(s) you want to work on are loaded into the Geoscience Analyst viewer, save the Geoscience Analyst workspace ('File –> Save workspace'). This may require that you have already created / entered your Seequent ID.

For existing Geoscience Analyst users, while no issues were found during testing, GAMS does overwrite the loaded workspace and so it is recommended you save an alternate (backup) version of your workspace prior to using the GAMS GUI.

CHAPTER

TWO

GUI OVERVIEW

2.1 Launching the GUI

Once installed, the GUI can be launched from the terminal used during the installation process (also ensure you are in the same conda environment as gams was installed in) as:

```
gams
```

A file-open dialog box will immediately open where you can load the Geoscience Analyst workspace (.geoh5 file) you wish to work from. A second dialog box will open where you can select the specific grid you want to work on. Currently, any of the grids from the selected workspace will be displayed in this list, and there is no check that the grid you select is magnetic data. An error may occur if the grid you select is not compatible with the GAMS processing workflow. You may later select a different grid within this workspace, or open a different workspace altogether.

Note that the padding and tapering calculations used work best if the IGRF has been removed from the data - the apparent susceptibility calculations also assume IGRF has been removed.

2.2 Main Window

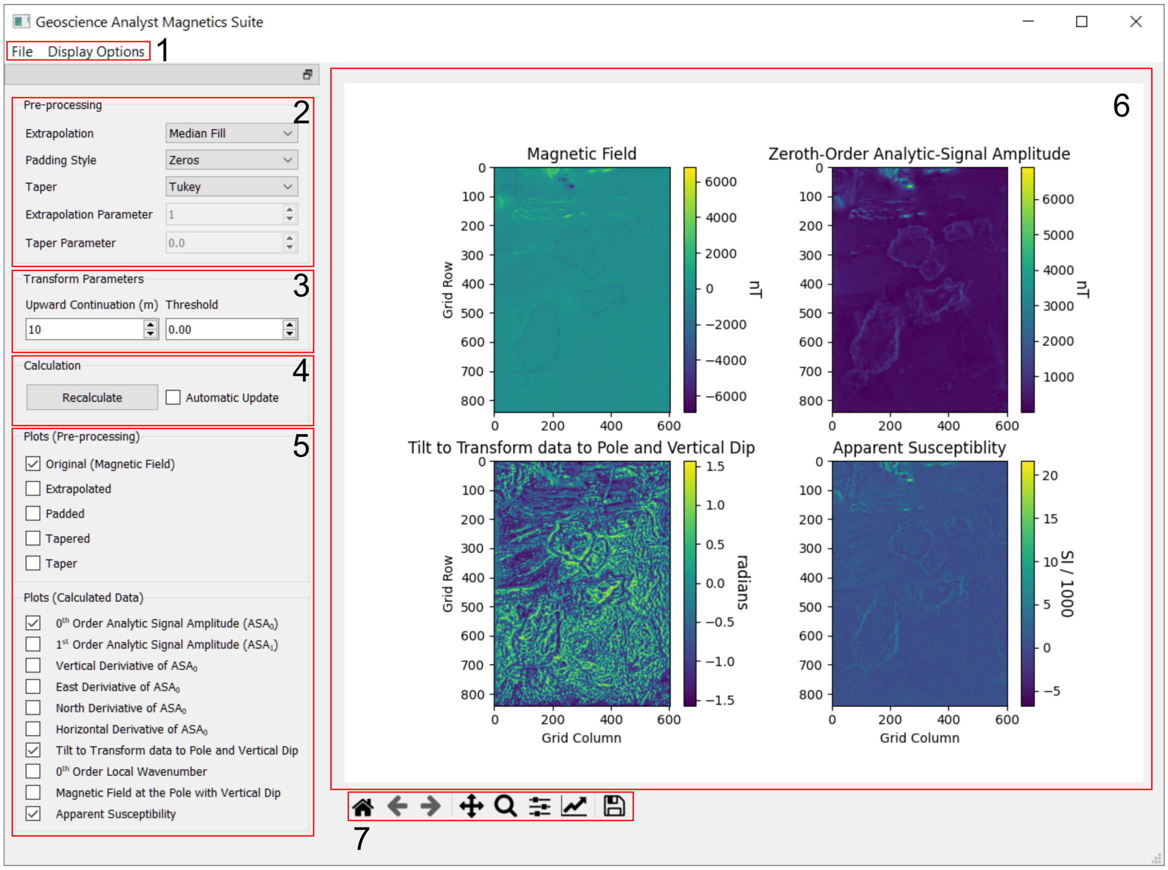
The selected workspace and grid are imported and the data transformations are calculated using default pre-processing settings. In general, this should only take a few seconds, however for large data sets or those with irregular geometries may take a bit longer.

- Note, the default pre-processing options have been chosen to minimize the initial load time, and may not be well suited to your particular data set. Some experimentation with the different options may be required. These options are discussed below.

The main window has 7 main areas:

1. *Menu Bar*
2. *Pre-processing*
3. *Transform parameters*
4. *Recalculate Buttons*
5. *Plots Options*
6. *Plot Window*
7. *Toolbar*

Items 2-5 are contained within a docking window that can be detached from the main window if needed by dragging and dropping the bar at the top with the detached window icon, above the "2" label in the figure below.



2.2.1 Menu Bar

The top menu bar has two buttons: 'File' and 'Display Options'.

Under the 'File' button, you can load a new workspace ('File –> Open –> GA Workspace ') or load a new grid from the current workspace ('File –> Open –> Grid').

Using 'File –> Write', the currently plotted grids can be written back into the loaded GA workspace. Note that the GA viewer may not automatically update with the new grids. To access the new grids, re-open the workspace in GA. The calculated grids are imported into GA using the same names as are listed in the checkboxes and plot titles

- Note: The 'padded' and 'tapered' grids are cannot be exported back into GA as they are different sizes than the original, and will be ignored when writing the grids.

Under the Display Options menu button, you can select and invert the colour map for the plots ('Display Options –> Color Map –> ?'), as well as toggle the display of colour bars on and off in the plot window ('Display Options –> Colour Bars'). The five available colour bars are standard from Matplotlib, and include perceptually uniform options suitable for red-green colour blind users (viridis and grey).

If 'Link Axes' is checked, panning / zooming any plot will apply the same operation to all plots, allowing closer examination of each grid at a specific location. These operations are undertaken using the *Toolbar* options described below.

- Note: The Link Axes option does not work for the 'Tapered' and 'Padded' grids, as they have a different size than the rest.

By default, the colour cut-off values for the plotted grids is set to +/- two standard deviations from the median value. The number of standard deviations can be modified by using the '# Standard Deviations' button. Unchecking 'Standard Deviation Cutoff' will reset the cut-off limits to the actual minimum and maximum values of the grid.

2.2.2 Pre-processing

The Fourier transformation applied when calculating the final grids assumes that the discretely sampled data repeats periodically - violation of this assumption can result in discontinuities or ringing in the calculated (transformed) data. Similarly, discontinuities in the original data (as might occur when an irregular shaped grid is infilled with zeros or nulls) can also result in artefacts (e.g., Gibb's phenomenon). Careful application of the pre-processing steps is required to reduce or remove such artefacts.

The 'Pre-processing' section contains options for the main steps in pre-processing (prior to the Fourier transform) the magnetic data prior to calculation of the relevant transforms. These are:

- *Extrapolation* - Options on how to infill holes and extend edges to generate a rectangular grid
- *Padding* - Options on how to generate the grid padding
- *Tapering* - Options on what kind of taper to use prior to calculating the FFTs

In addition to the main drop-down menus, additional parameter options become available for particular selections:

- *Extrapolation Parameter* - Controls the infill distance when using 'Mirror Image' extrapolation
- *Taper Parameter* - Controls the roll-off distance for the 'Kaiser' and 'Blur' taper options

2.2.3 Transform Parameters

The Transform Parameters contains spin boxes to control two parameters used during the calculation of the transformed grids.

- 'Upward Continuation' is the distance that the data are upward continued for calculation of the vertical derivatives and subsequent grids that rely on those. This parameter is specified in whatever units the original grid is in (likely meters).
- 'Threshold' value is an estimate of the noise level applied to the vertical and horizontal derivatives when calculating the tilt transformed data. The units of this value will be the units of the magnetic field divided by the units of the cell spacing (typically nT/m). In most cases, we find that it is not necessary to apply a threshold, so the default value is zero.

2.2.4 Recalculate Buttons

In order to minimize unnecessary computations of all the calculated data, the grids are recalculated and plots redrawn when the 'Recalculate' button is pushed.

This can be bypassed by checking the 'Automatic Update' box, which forces recalculation and redrawing when any of the pre-processing parameters are changed.

2.2.5 Plots Options

This section contains options for which plots to generate.

The first group of checkboxes, 'Plots (Pre-processing)', is for each of the pre-processing steps, including the original data. Checking or unchecking any of the boxes will automatically update the plots included in the plot window to reflect the change. These plots are likely not useful from an interpretion standpoint, but can be helpful in determining whether or not the GUI is performing the pre-processing steps as you expect, and in determining what is the best combination of pre-processing parameters for your data.

The second group of checkboxes, 'Plots (Calculated Data)', contains checkboxes for some of the calculated (or transformed) grids, including the vertical and horizontal derivatives, tilt transform, and apparent susceptibility.

2.2.6 Plot Window

The largest section of the GUI is reserved for the plot window, which as the name suggests, is where the plots are drawn. This window is updated whenever any of the plot checkboxes are checked or unchecked, whenever the 'Recalculate' button is pushed, or if any pre-processing options are changed while the 'Automatic Update' checkbox is checked.

2.2.7 Toolbar

The toolbar at the bottom of the *Plot Window* operates as a normal matplotlib toolbar. The default buttons from left to right are:

- Home - restores the default view of all plots
- Back - Undo the last modification performed by a toolbar button
- Forward - Redo the last undone action
- Pan - Click to activate, then click and drag on any plot window to pan
- Zoom - Click to activate, then click and drag a box to zoom into any plot window
- Configure Subplots - Click to access specifications of the plotting grid (e.g., axes layout)
- **Axis and Image parameters - Click to access specifications for a particular axis (e.g., title, colour map min/max values).**
  - Note that more sophisticated plotting options (colour options, specifically) are available in Geoscience Analyst.
- Save - Click to save the current plot window to an image file

If the 'Link Axes' checkbox is checked, then any panning / zooming will be applied to all axes simaltaneously.

CHAPTER

THREE

PROCESSING OPTIONS

The options described below are implemented as of the current version of GAMS. As this code has been made open-source, other pre-processing methods which might be more appropriate for magnetic data could be added to the software.

3.1 Extrapolation

Some of the options are wrappers for the scipy.interpolate.griddata function. Grids are padded using the 'Reflect' method prior to extrapolation, so that missing corners are interpolated from between mirrored images of the original grid.

These are listed in order of increasing computation time.

- Median Fill
  - Infills all zero (or NaN) values with the median value of the grid. The zero / NaN values are kept as a mask, which is reapplied after all the grids have been calculated. This is default value, as it is fast to apply, but may not be appropriate for all grids.
- Zero Fill
  - Keeps all zero values, but infills NaNs with zeros. Not useful on its own, but necessary for the *Blur* taper.
- Mirror Image
  - Attempts to infill missing data using a mirrored version of the original grid (mirrored about the edge of the missing data). Uses the 'Pad Parameter' spin box to determine how far to infill / mirror. Optimal 'Pad Parameter' value will depend on the size of
- Nearest
  - Wraps the scipy.interpolate.griddata function with method='nearest'. Replaces zeros or NaN values with the value of the data point nearest to the point of interpolation.
- Linear
  - Wraps the scipy.interpolate.griddata function with method='linear'. Tessellates the input point set to N-D simplices, and interpolates linearly on each simplex.[1]
- Cubic
  - Wraps the scipy.interpolate.griddata function with method='cubic'.[1]

[1] See the https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.griddata.html for further details.

3.2 Padding

Many of the options are wrappers for the numpy.pad function, with padding sizes calculated to make the grid into a square.

All padding methods extend the grid such that it is square with dimensions 2*max(nx,ny), where nx and ny are the dimensions of the original grid, by mirroring the original grid in each direction.

- Zeros
  - Wraps numpy.pad with mode='constant'.
  - Likely never the optimal choice of padding, but is quick to calculate and provides a useful starting point for comparison. This is the default value.
- Wrap
  - Wraps numpy.pad with mode='wrap'.
  - Perfectly satisfies the FFT assumption of periodicity, but it also will create discontinuities for most data at the wrapped edges (unless the data is also already periodic)
- Reflect
  - Wraps numpy.pad with mode='reflect'. Mirrors the vectors about the first and last values.
  - Not perfectly periodic (pre-taper), but is less likely to create discontinuities at the grid edges. May not be the case if spatial derivates are high near the edges.
- Reflect (Inverse)
  - Applies numpy.pad with mode='reflect', and then inverts the mirror images about the edge of each vector.
  - Operates similarly to the 'Reflect' option, but inverting the mirror images about the edge values can reduce the likelihood of creating discontinuities when the spatial derivatives are large near the edges.
  - Note that with this method, the corner pads may be discontinuous where they join.
- Derivative Mirror
  - Mirrors the grid similarly to the 'Reflect (Inverse)' option. Calculates the mirror images by applying the negative of the derivative at the mirrored position in the original grid.
  - This gives the same result as 'Reflect (Inverse)' for top, bottom, left, and right pads; however, the corner pads should smoothly connect to the other padding sections.
  - The downside is that some striping can occur in the padded regions if the spatial derivative in the direction parallel to a particular edge varies rapidly (e.g., the north-south derivative along the western edge).

3.3 Tapering

This section gives details of each of the tapering options available in the GUI.

Most of the options are wrappers for the signal.windows.get_window function, with parameters and additional logic set to generate tapers at the correct size and, when necessary, roll-off parameters.

The choice of the best taper will depend largely on the shape of the original grid, and the quality of data along its edges. For example, while the Tukey taper preserves the values of the data along the grid edges, it may not be optimal for irregular grid shapes, nor does it have an optimal frequency response. On the other hand, the Blur taper is form-fitted to grids of any shape, but overwrites values near the grid edge.

Taper roll-off starts at the center of the grid for all tapers except the Tukey, PSD, and Blur.

- Tukey
  - Wraps the signal.windows.get_window function with the window set to 'Tukey'. The taper is flat (equal to one) until the edges of the extrapolated grid, after which point there is a cosine taper.[1]
- Kaiser
  - Wraps the signal.windows.get_window function with the window set to 'Kaiser'. Uses the 'Taper Parameter' to determine the roll-off. A parameter value of 0 is a boxcar; increasing values give larger roll-off windows.[2]
- Hamming
  - Wraps the signal.windows.get_window function with the window set to 'Hamming'.[3]
- Hanning
  - Wraps the signal.windows.get_window function with the window set to 'Hanning'.[4]
- Blackman
  - Wraps the signal.windows.get_window function with the window set to 'Blackman'.[5]
- Blur
  - The 'Blur' taper sets the *padding* setting to 'Zeros', and the *extrapolation* setting to 'Zero Fill'. It then applies a gaussian blur along the edge of the original grid such that the grid values roll off into the median value.
  - The blur is applied using scipy.ndimage.gaussian_filter, with sigma set to the Taper Parameter. Larger values of the Taper Parameter give larger blur windows.
  - Likely not as good as a Tukey taper for perfectly rectangular grids, but is able to better fit irregularly shaped grids.
- PSD
  - The Periodic-Smooth Decomposition (PSD; Moisan, 2011) method is not actually a tapering method, but rather a decomposition method which isolates the periodic and smooth components of an image. Replacing the original data with only its periodic component means that there is no need for actual padding (beyond that required to make the grid square) nor tapering to fulfill the assumptions of the FFTs.
  - When this option is selected, the original grid is only padded as much as is needed to make the image square.
  - Note that the isolated periodic component may not be an accurate representation of the original data, particularly close to the edges.
  - An example of the decomposition and the basis for the PSD code used in GAMS can be found here.

[1] See https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.signal.tukey.html for more details.
[2] See https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.signal.kaiser.html for more details.
[3] See https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.signal.hamming.html for more details.
[4] See https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.signal.hanning.html for more details.
[5] See https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.signal.blackman.html for more details.

CHAPTER

FOUR

OTHER INFO

4.1 Acknowledgements

All test data used in the making of GAMS were kindly provided by the Ontario Geological Survey, courtesy of Desmond Rainsford.

4.2 References

Smith, R. 2021, Transformative geophysics: Alternatives to the reduction-to-pole transformation of magnetic data. Third Australian Exploration Geoscience Conference Extended Abstract. pp 1-4.

Moisan, L.J., *Periodic Plus Smooth Image Decomposition*, Journal of Mathematical Imaging and Vision 39, 161-179 (2011). https://doi.org/10.1007/s10851-010-0227-1

4.3 Help

For feature requests, bug fixes, design suggestions, or anything else, an issue or pull request can be made through the github page.

4.4 Change Log

4.5 License