02_manual_two_point_azimuth.pdf

File description: This document describes the installation and execution of the program "gui_shield_field_azimuth_v1p1_mt.app" (OS X) or .exe (PC). This program is a stand-alone executable bundle that does not require a MATLAB license to run. However, the first time the software is run, the MATLAB Runtime Environment will need to be installed (~500 MB), which is a free set of Mathworks-supplied files necessary for execution. The source code is also provided separately for those with MATLAB already installed or expert users who may which to modify the code or probe its inner workings.

Author: Brad Thomson (bjt@bu.edu), Boston University Center for Remote Sensing

Date: 2016-04-18

Publication documenting software use: Thomson, B. J. and N. P. Lang (2016) Volcanic edifice alignment detection software in MATLAB: Test data and preliminary results for shield fields on Venus, *Computers & Geosciences*.

Table of Contents

1. Installation

1.1. Installing on a Mac

To begin the installation process, navigate to the folder "04_executable_gui_mac" and double-click the icon to launch the installation wizard "MyAppInstaller_web.app." The wizard will step through a series of screens to guide the user through the installation process.

The first prompt to the user is to allow java to make changes, and the second asks the user to confirm they want to start the installation process by clicking "Next." The following prompt queries the user to, "Choose an installation folder" (**Fig. 1**).

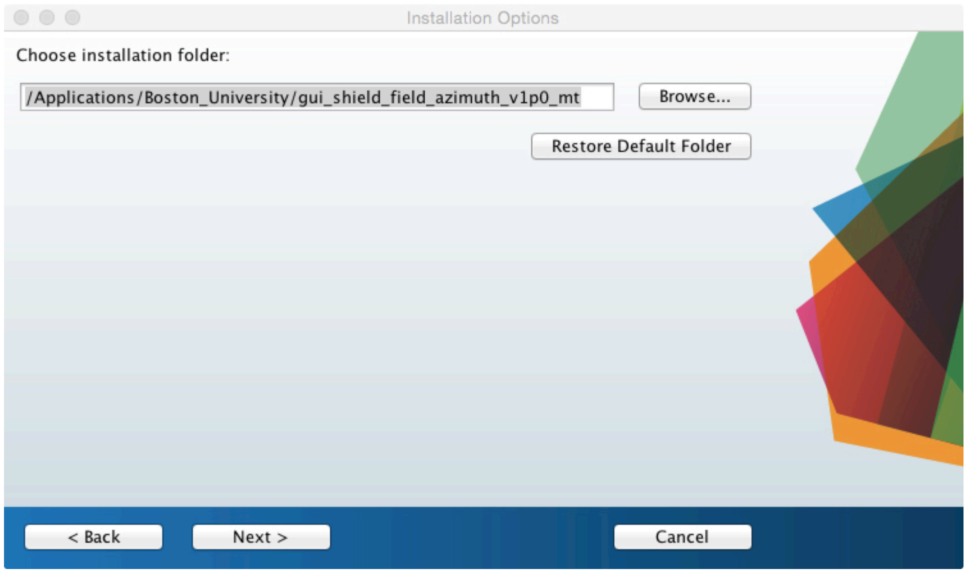
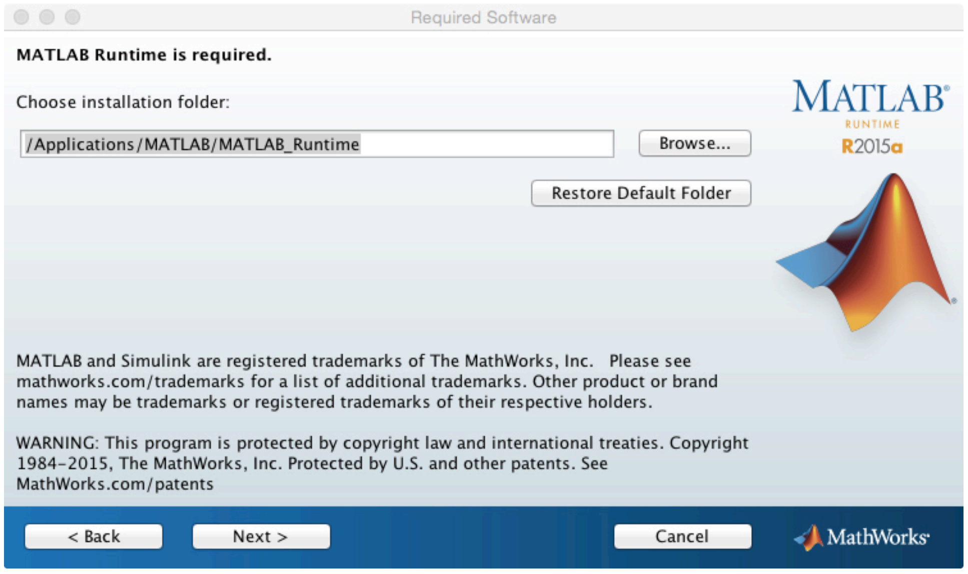

Installation Options
Choose installation folder:
/Applications/Boston_University/gui_shield_field_azimuth_v1p0_mt
Browse...
Restore Default Folder
< Back
Next >
Cancel

Figure 1. Snapshot of Mac installation wizard prompting user to specify the location of the installation folder.

The next user prompt is to choose a location to install MATLAB Runtime. The MATLAB Runtime is a standalone set of shared libraries that enables the execution of compiled MATLAB applications or components on computers that do not have MATLAB installed. The default installation location is given in **Fig. 2**.


Required Software

MATLAB Runtime is required.

Choose installation folder:

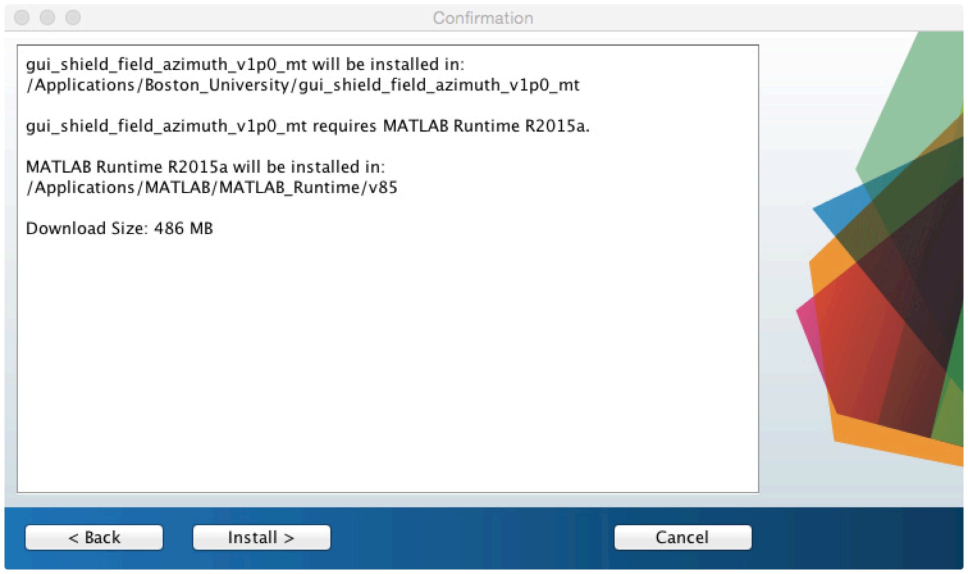
/Applications/MATLAB/MATLAB_Runtime

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Please see mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

WARNING: This program is protected by copyright law and international treaties. Copyright 1984-2015, The MathWorks, Inc. Protected by U.S. and other patents. See MathWorks.com/patents

Figure 2. Snapshot of Mac installation wizard prompting user to specify the location of the MATLAB Runtime shared libraries.

A subsequent prompt asks the user to accept or reject Mathworks license agreement. Finally, a summary page displays the user-specified installation locations and asks the user to confirm the installation details (**Fig. 3**).



gui_shield_field_azimuth_v1p0_mt will be installed in:
/Applications/Boston_University/gui_shield_field_azimuth_v1p0_mt

gui_shield_field_azimuth_v1p0_mt requires MATLAB Runtime R2015a.

MATLAB Runtime R2015a will be installed in:
/Applications/MATLAB/MATLAB_Runtime/v85

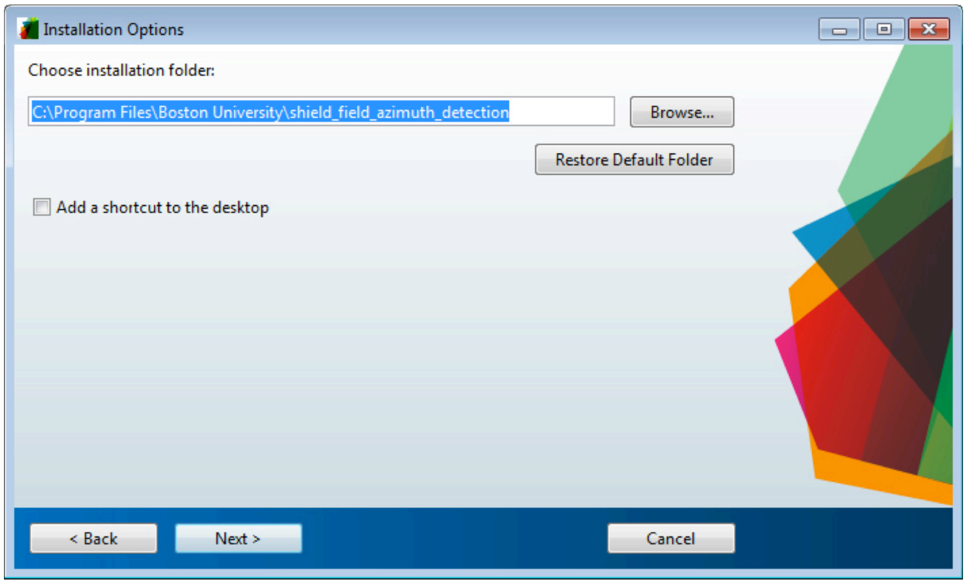
Download Size: 486 MB

Figure 3. Snapshot of final Mac screen of installation wizard prompting user to confirm the details of the installation.

Once the MATLAB Runtime libraries are downloaded and the installation is complete, the following three folders are created in the specified location (in this case /Applications/Boston_University/gui_shield_field_azimuth_v1p0_mt): "appdata," "application," and "uninstall." The appdata folder contains a series of binary files necessary for execution. The application folder contains the file "**gui_shield_field_azimuth_v1p2_mt.app**," which is the application itself. It can be launched by double-clicking in Finder just like a standard Macintosh application. The folder uninstall contains a zipped archive "installbundle.zip" that contains a copy of the first two folders in the application folder (i.e., it doesn't actually un-install the application as the name might imply).

1.2. Installing on a PC

To begin the installation process on a PC, navigate to the folder "05_executable_gui_win" and double-click the icon to launch the installation wizard "MyAppInstaller_web.exe." The wizard will step through a series of screens to guide the user through the installation process. After asking the user to confirm they want to start the installation process by clicking "Next," the wizard prompts the user to "Choose an installation folder" (**Fig. 4**).



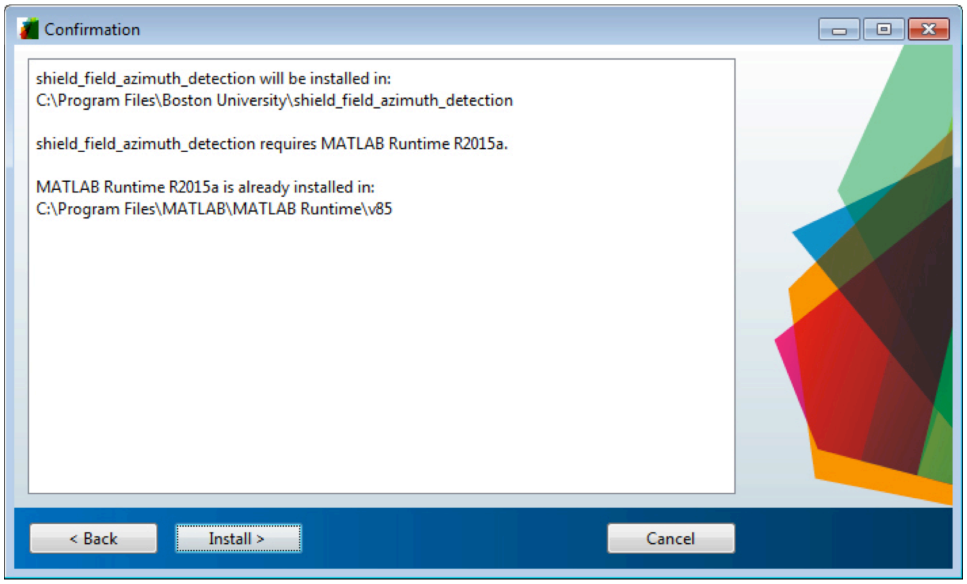
Choose installation folder:

C:\Program Files\Boston University\shield_field_azimuth_detection

Add a shortcut to the desktop

Figure 4. Snapshot of the Windows installation wizard that prompts the user to specify the location of the installation folder.

The user is then prompted to select the installation location of MATLAB Runtime, which is a standalone set of shared libraries that enables the execution of compiled MATLAB applications or components on computers that do not have MATLAB installed. A subsequent prompt asks the user to accept or reject Mathworks license agreement. Finally, a summary page displays the user-specified installation locations and asks the user to confirm the installation details (**Fig. 5**).



shield_field_azimuth_detection will be installed in:
C:\Program Files\Boston University\shield_field_azimuth_detection

shield_field_azimuth_detection requires MATLAB Runtime R2015a.

MATLAB Runtime R2015a is already installed in:
C:\Program Files\MATLAB\MATLAB Runtime\v85

Figure 5. Snapshot of the Windows installation wizard that prompts the user to specify the location of the installation folder.

Once the MATLAB Runtime libraries are downloaded and the installation is complete, the following three folders are created in the specified location (in this case C:\Program Files\Boston_University\shield_field_azimuth_detection\): “appdata,” “application,” and “uninstall.” The appdata folder contains a series of binary files necessary for execution. The application folder contains the file “**gui_shield_field_azimuth_v1p2_mt.exe**,” which is the application itself. It can be launched by double-clicking in Windows Explorer just like a standard PC application. The folder uninstall contains a zipped archive “installbundle.zip” that contains a copy of the first two folders in the application folder (i.e., it doesn’t actually un-install the application as the name might imply).

1.3. Already have MATLAB installed?

If you already have MATLAB installed on your system, the GUI can be launched by navigating to the “03_source_code” folder and typing the name of the GUI (**gui_shield_field_azimuth_v1p2_mt**) into the MATLAB command window. If the Mapping Toolbox is not licensed on your system, one can still run the compiled executable by installing the MATLAB Runtime Environment. Installing the Runtime Enivornment will not interfere with your existing MATLAB installation.

2. How to run the software

This software is an implementation of the original two-point azimuth method [*Lutz*, 1986] and a second modified two-point azimuth algorithm to focus on smaller spatial scales [*Cebriá et al.*, 2011] in a single graphical user interface (GUI) built using MATLAB (MATrix LABoratory) software. The GUI was constructed using the GUIDE functionality in MATLAB (Graphical User Interface Development Environment).

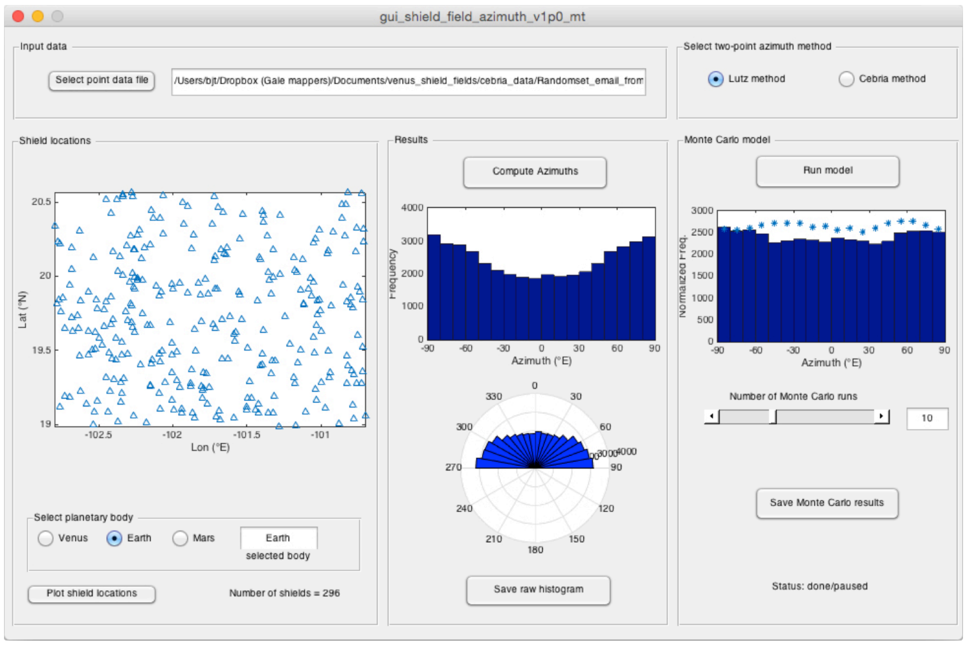
There are two ways to run the software. Existing MATLAB users can launch the GUI at the MATLAB command line by typing the GUI program name: **gui_shield_field_azimuth_v1p2_mt.** The alternative way to run the software is to use the automated wizard to download and install the MATLAB Runtime Environment. After installation, launch the application by navigating to the directory specified during the installation process in Finder (OS X) or Windows Explorer (PC). Then double-click on the file **gui_shield_field_azimuth_v1p2_mt .app** (on a Mac) or **gui_shield_field_azimuth_v1p2_mt.exe** (on a PC). Once the GUI appears, the user executes a series of steps. These are given in **Table 1** and illustrated graphically in **Fig. 7**.

Table 1: Sequence of processing steps.

Step No.	Description
(1)	Click “select point data file” button to bring up file-selection dialog.
(2)	Navigate to chosen folder, select comma-separate value file (*.csv) that is a two-column text listing of center longitude, latitude points.
(3)	Select radio button indicating choice of two-point azimuth method: Lutz method (default) or Cebriá et al. method.
(4)	In left panel, select radio button for planetary body of interest (currently Venus, Earth, or Mars).
(5)	Click “Plot shield locations” button to create x-y plot of shield locations in left panel. Note this also served to verify that the point data file was ingested correctly.

Step No.	Description
(6)	In the middle panel under “Results,” select the “Compute Azimuths” button. This computes a “raw” two-point azimuth method or Cebriá et al. method, and displays the results in a histogram and rose diagram.
(7)	[Optional] Data from the raw histogram can be saved in comma-separated value (csv) format by clicking on the “Save raw histogram” button at the bottom of the middle panel.
(8)	In the right-most panel within the “Monte Carlo model” button group, the user specifies the number of Monte Carlo runs desired using either the bar slider or by entering an integer value into the text field to the right of the slider. The default value is 10, although the recommended minimum number of runs is 100.
(9)	The user then clicks the “Run Model” button on the right panel to execute the specific number of Monte Carlo runs.
(10)	[Optional] Similar to step #7 data from the normalized histogram can be saved in csv format by clicking on the “Save Monte Carlo results” button at the bottom of the right panel.

In the GUI, the user ingests a pre-prepared text file that is a 2-column listing of the center latitude and longitude of each volcanic construct (Table 1, steps 1-2). Inputting the point data as decimal degrees rather than Cartesian *x, y* distances implicitly avoids introducing distortion due to planetary curvature. The software has been configured so that the user can designate the planetary body of interest (currently Earth, Venus, or Mars). The main body of the GUI consists of three panels (**Fig. 6**), and the sequence of processing steps in given in the flowchart in **Fig. 7**. In the left-most panel in **Fig. 6**, the distribution of point features (e.g., shields) can be visually confirmed in a lat-lon scatter plot. The middle panel displays a raw, uncorrected histogram of orientation measurements. These measurements utilize the MATLAB function “azimuth,” part of the Mapping Toolbox, which determines the azimuth between two points on a given ellipsoid. In the right-most panel, the user specifies the number of Monte Carlo runs to randomly place an equivalent number of shields within the boundaries defined by the edge edifices.



/Users/bjt/Dropbox (Gale mappers)/Documents/venus_shield_fields/cebria_data/Randomset_email_from

[✓ Lutz method] Cebria method

Venus [✓ Earth] Mars

Earth selected body

Number of shields = 296

Number of Monte Carlo runs 10

Status: done/paused

Figure 6. Snapshot of GUI (graphical user interface) depicting software layout.

In the right-most panel, the user specifies the number of Monte Carlo models to run. Each model randomly places an equivalent number of shields within a region identical in shape to the original, and the results from these empirical distributions are used to correct for the effect of field shape and also determine if the observed distribution is consistent with a random distribution. There are five basic steps in the Monte Carlo model. First, the bounding region is defined by the edge edifices, which delineate the convex hull or convex envelope. Second, this convex hull is divided into a set of constituent triangles using a Delaunay triangulation [e.g., *Lee and Schachter*, 1980]. Next, we compute the area of each triangle, and assign it a weighting factor that is proportional to its area. Finally, the position of each random point x in a given triangle is determined by generating two random numbers a_1 and a_2 to calculate $x = a_1\ (v_1 - v_0) + a_2\ (v_2 - v_0)$, where v_0, v_1, and v_2 are vertices of the triangle [*Weisstein*, 2016].

Upon execution, a "normalized" histogram is produced from the Monte Carlo results whereby each histogram cell is set equal to the expected value times the observed value divided by the mean value in the Monte Carlo runs (equation 1, after *Lutz* [1986]).

$$\hat{z}_i = \left[\frac{\frac{N(N-1)}{2k}}{z_{MC-mean,i}}\right] z_{obs,i} \text{ for } i = 1 \text{ to } k \qquad (1)$$

Here, $\hat{z}_i$ is the normalized value of the i^{th} bin; the quantity $N\,(N - 1) / 2k$ is the expected value per bin; k is the number of bins (18 in this instance); $z_{MC-mean,i}$ is the mean value of the i^{th} bin averaged from all of the Monte Carlo runs, and $z_{obs,i}$ is the observed histogram value of the i^{th} bin. See *Thomson and Lang* [2016] for additional details.

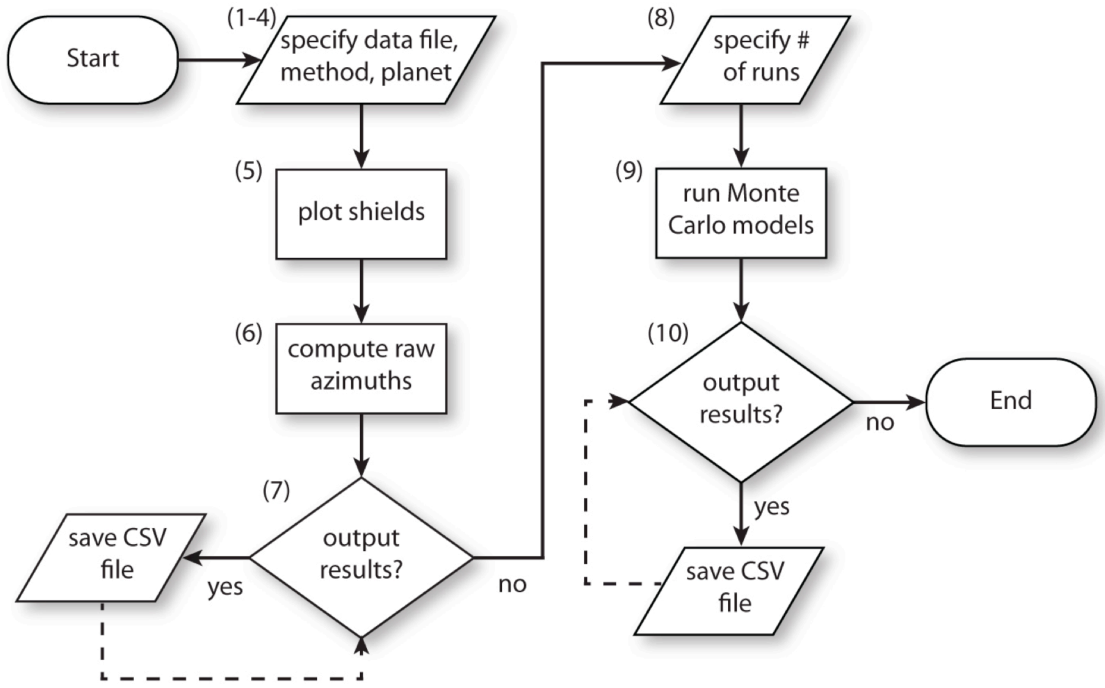


Figure 7. This flowchart details the sequence of processing steps in the GUI to run the *Lutz* [1986] or *Cebria et al.* [2011] two-point azimuth methods. Rounded squares give the starting and end points of the sequence. A parallelogram indicates an input or output operation, a rectangle indicates a data manipulation step, and a rhombus indicates a decision point. Numbers enclosed in parentheses correspond to the processing steps given in Table 1.

3. Source data files

The folder "03_source_code" contains all of the source code files developed for this project. Not included are the files *referenceSphere.m* and *azimuth.m*, which are two components from the Mapping Toolbox that are used in some functions. The text file "aaa_readme_source_code_file_list.txt" gives a list of all functions and sub-functions.

Main program, i.e., GUI for two-point azimuth method:

gui_shield_field_azimuth_v1p2_mt.m

%

% Notes: The "_mt" appended to the file name indicates that the MATLAB mapping toolbox was used by some functions. So if the user decides to modify the code and re-run it, the Mapping Toolbox will have to be enabled.

Functions called:

(1) *plot_xy_point_data.m*

% Notes: called upon execution of "Plot shield locations" button, plots point data in left most panel of GUI.

(2a) *azimuth_method_v3_mt.m*

Notes: Called if radio button "Lutz method" is selected upon selection of the "Compute Azimuth" button in middle panel. This function uses the Lutz two-point azimuth method and outputs a histogram and rose diagram in the middle panel of the GUI.

Sub-functions called:

(2a.1) *read_xy_point_data* : read in designated input file and return arrays of X and Y values.

(2a.2) *two_pt_azimuth_calcs_v3_mt* : calculate 2-point azimuth values.

Sub-Sub-functions called:

(2a.2.1) *referenceEllipsoid* : From Mapping Toolbox, used to define a planetary ellipsoid with specific radii.

(2a.2.2) *azimuth* : From Mapping Toolbox, used to find angle and distance between two points on a given reference ellipsoid.

(2b) *azimuth_method_cebria_mt.m*

Notes: Called if radio button "Cebria method" is selected upon selection of the "Compute Azimuth" button in middle panel. This function uses the Cebria two-point azimuth method and outputs a histogram and rose diagram in the middle panel of the GUI.

Sub-functions called:

(2b.1) *read_xy_point_data* : read in designated input file and return arrays of X and Y values.

(2b.2) *two_pt_azimuth_cebria_v3_mt* : calculate 2-point azimuth values

Sub-sub-functions called:

(2b.2.1) *referenceEllipsoid* : From Mapping Toolbox, used to define a planetary ellipsoid with specific radii.

(2b.2.2) *azimuth* : From Mapping Toolbox, used to find angle and distance between two points on a given reference ellipsoid.

(2b.3) *plot_xy_point_data.m* : replots x-y data in left hand panel, but adds line segments between points that meet Cebria's criteria.

(3a) *monte_carlo_model_v4_mt.m*

Notes: Called if radio button "Lutz method" is selected upon selection of the "Run Model" button in right-hand panel. This function runs a user-specific number of Monte Carlo models, computes the Lutz method on each, and then averages the results. The output is a normalized histogram in the right-hand panel of the GUI.

Sub-functions called:

(3a.1) *read_xy_point_data* : read in designated input file and return arrays of X and Y values.

(3a.2) *two_pt_azimuth_calcs_v3_mt* : calculate 2-point azimuth values.

Sub-Sub-functions called:

(3a.2.1) *referenceEllipsoid* : From Mapping Toolbox, used to define a planetary ellipsoid with specific radii.

(3a.2.2) *azimuth* : From Mapping Toolbox, used to find angle and distance between two points on a given reference ellipsoid.

(3a.3) *triangle_area_calc* : calculate area enclosed by 3 x,y points.

(3b) *monte_carlo_model_v3_cebria_mt.m*

Notes: Called if radio button “Cebria method” is selected upon selection of the “Run Model” button in right-hand panel. This function runs a user-specific number of Monte Carlo models, computes the Cebria method on each, and then averages the results. The output is a normalized histogram in the right-hand panel of the GUI.

Sub-functions called:

(3b.1) *read_xy_point_data.m* : read in desigated input file and return arrays of X and Y values.

(3b.2) two_pt_azimuth_cebria_MC_only_mt.m : calculate 2-point azimuth values using the Cebria et al. (2011) method. Similar to two_pt_azimuth_calcs_v3_mt, but here the number of azimuth values in each empirical Monte Carlo model matches that in the observed distribution.

Sub-sub-functions called:

(3b.2.1) *referenceEllipsoid* : From Mapping Toolbox, used to define a planetary ellipsoid with specific radii.

(3b.2.2) *azimuth* : From Mapping Toolbox, used to find angle and distance between two points on a given reference ellipsoid.

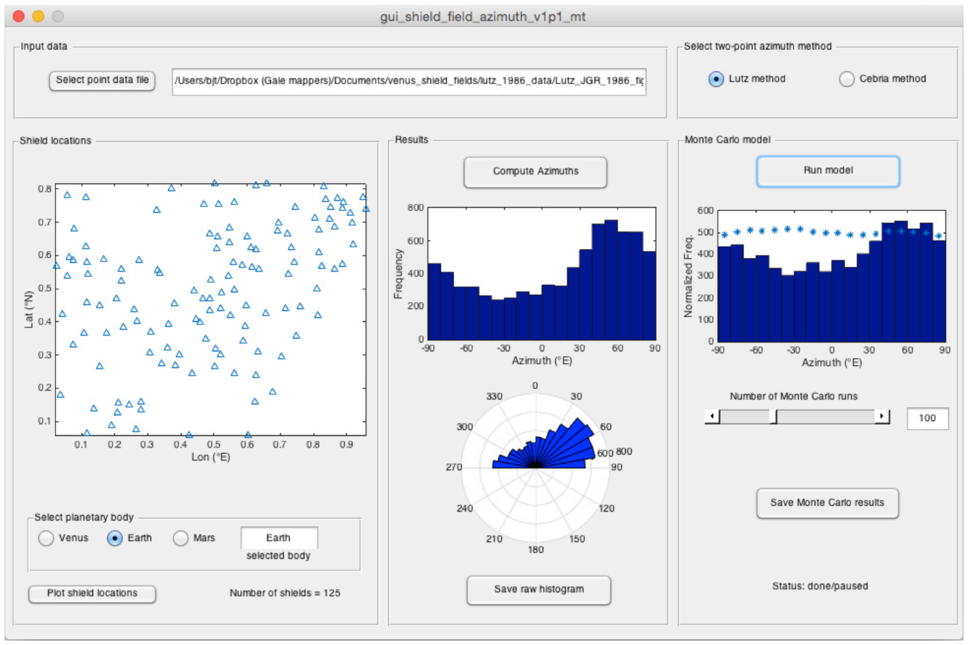
(3b.3) *triangle_area_calc.m* : calculate area enclosed by 3 x,y points.

4. Test data

We have included several sets of data to use as test cases to verify that the software is working as expected. The folder “06_test_data” contains 11 files: 4 point datasets (.csv), a readme text file (.txt), and two snapshots of the MATLAB program after processing the data using the Lutz method (in jpeg and pdf format, see **Fig. 8**). Also included are text-based outputs saved from the GUI for verification, i.e., data from the raw histogram and normalized histogram, respectively. Each output is in .csv format with a detached, text file header (.txt).

List of files in folder 06_test_data:

figs06-07_Cebria_random_points.csv
figs08-09_chernava_shields.csv
figs10-11_sf_31n310e_shields.csv
Lutz_JGR_1986_fig13_data.csv
Lutz_JGR_1986_fig13_data_readme.txt
Lutz_JGR_1986_fig13_data_gui_snapshot.jpg
Lutz_JGR_1986_fig13_data_gui_snapshot.pdf
Lutz_JGR_1986_fig13_output_hist_data_header.txt
Lutz_JGR_1986_fig13_output_hist_data.csv
Lutz_JGR_1986_fig13_output_MC_data_header.txt
Lutz_JGR_1986_fig13_output_MC_data.csv



/Users/bjt/Dropbox (Gale mappers)/Documents/venus_shield_fields/lutz_1986_data/Lutz_JGR_1986_fi

[✓ Lutz method] Cebria method

Venus [✓ Earth] Mars — Earth selected body

Number of shields = 125

Number of Monte Carlo runs 100

Status: done/paused

Figure 8. Snapshot of GUI (graphical user interface) after running 100 Monte Carlo runs on data from *Lutz* [1986].

5. References.

Cebriá, J. M., C. Martín-Escorza, J. López-Ruiz, D. J. Morán-Zenteno, and B. M. Martiny (2011), Numerical recognition of alignments in monogenetic volcanic areas: Examples from the Michoacán-Guanajuato Volcanic Field in Mexico and Calatrava in Spain, *Journal of Volcanology and Geothermal Research*, *201*, 73-82.

Lee, D.-T., and B. J. Schachter (1980), Two algorithms for constructing a Delaunay triangulation, *International Journal of Computer & Information Sciences*, *9*(3), 219-242.

Lutz, T. M. (1986), An analysis of the orientations of large scale crustal structures: A statistical ap- proach based on areal distributions of pointlike features, *J. Geophys. Res.*, *91*(B1), 421-434.

Weisstein, E. W. (2016), Triangle Point Picking, from MathWorld – A Wolfram Web Resource, http://mathworld.wolfram.com/TrianglePointPicking.html.