Pymagra manual

Program for treatment of magnetic and gravity data

The program allows reading magnetic or gravity data with different formats:

- Geometrics « .stn » files for magnetic gradiometry data
- Geometrics ".dat" files (MagMap export in Surfer format): x, y, top, bottom, time, date, line
- GXF format
- BRGM aeromagnetic data files
- Simple text file with x,y,z,v(,v2), e.g. outputted by program mgwin.exe

Installation

Mouse

It is strongly recommended to use a mouse. The program needs the central mouse button (or wheel). If you are working with a touchpad, configure it such that it simulates the central mouse button. The simplest is certainly to configure a three-finger-touch as central button. For this, under Windows press the WINDOWS-Key+I. On the screen that appears, click on "Périfériques" (the English version may be "Devices"). There, on the left-side panel, search "Pavé tactile" ("Touch Pad" in English?). Scroll down until you see the image with three fingers. Below this image, under "Appuis", change to "Button central de la souris" ("Central mouse button"?).

Installation is done with

pip install pymagra

Once installed, open an Anaconda Powershell window, cd to the folder containing the data file(s) and type:

pymagra followed by ENTER

This may be done also from any other command window (cmd under Windows) if the folder containing the executable is placed in the Path. In Windows, this path is C:\User\user_name\anaconda3\Skripts.

You may also open file pymagra_start.py (in C:\User\user_name\anaconda3\Lib\ site-packages\Pymagra) in a GUI like Spyder and run it from there. This gives the user the possibility to define a working directory at the beginning of pymagra (near line 18, search: "dir0 =" and change the given path), especially useful if you are working for several days with data in the same folder. If using Spyder, it is recommended to run the program within its own console: "Run -> Configuration per file -> Run with custom configuration -> Execute in dedicated console".

Running

After starting the program, the user is asked to choose the data file to be treated. By default, the program proposes files with extension .stn (Geometrics data), .gxf, .xyz or .dat. If the file name has a different extension, click in the extension selection and choose "all" to see other files. Extensions may be in capitals or not. For the moment, only .out is accepted as additional valid extension. For the

corresponding data formats, see appendix. If your data have one of the valid data formats but not the expected extension, just change the file extension.

Once the file chosen, different dialogue windows open:

1) Data types: choose whether the file contains magnetic or gravity data
2) Geometry parameters:

   The contents of this dialogue box depend on the data type. For .stn and .dat files, if they are tagged as magnetic data, the program supposes gradiometer data and asks for the following inputs:

   - Height of sensor connected to channel 1 above ground [m]
   - Were the two sensors placed vertically or horizontally?
   - Distance between the two sensors [m]. For vertical disposition, a positive distance implies that the sensor connected to channel 2 is above the one of channel 1 and a negative distance the reverse. For horizontal disposition, a positive distance means that the sensor connected to channel 2 is placed to the right of channel 1 in the direction of movement a negative distance means the reverse.
   - The coordinates of the data stored in the file are mostly local coordinates and the lines may not have been measured in N-S direction. In this case, you may give the direction of the local Y coordinates with respect to magnetic North (measured in degrees, positive from N to E). This direction will be indicated on all maps and is used for certain data treatments (reduction to the pole) and for inversion.
   - Text used in the title of most plots to specify data set.

   If data are in gxf (.gxf) format, only one sensor per file is accepted. In this case, only the height of the sensor and the direction of the Y coordinate with respect to magnetic N (for magnetic data) or geographic N (for gravity data) is asked for.

   For BRGM flight files (.xyz) this dialogue box is not opened since all necessary information is given within the file.

   If several files are opened, the parameters of the first one are proposed for all further data files.

3) Earth's magnetic field data: If one file contains magnetic data, a further dialogue box is opened asking for the local inclination and declination of the Earth's magnetic field measured in degrees.
   This dialog box appears only for the first magnetic file read, since it is supposed that all other files have been acquired nearby and the Earth's magnetic field has the same properties.

At the first time one works with data of a folder, the program presents these dialogue windows. The answers are stored in file "file_name.config" in the following order:

- Line 1: File type ("GEOMETRICS", GXF", "BRGM" or "MGWIN"
- Line 2: Data type ("magnetic" or "gravity")
- Line 3: Title text, used in the title of most plots. Typically place of data acquisition.
- Line 4: Line direction with respect to North in degrees
- Line 5: Height of sensor 1 in meters
- Line 6: Distance between two sensors in meters

- Line 7: Disposition of sensors (0=vertical, 1=horizontal)
- Line 8: Strength of Earth's magnetic field in the area in nT
- Line 9: Inclination of Earth's magnetic field in the area in degrees
- Line 10: Declination of Earth's magnetic field in the area in degrees
- Optional: Line 11: Time to be added to instrument time in seconds

Most of these data will not be needed in case of gravity data or magnetic data acquired at one height. In this case, dummy values are stored.

In the beginning of a run, the program searches this file pymagra.config. If the file is found, only the first dialogue box is presented (choose between gravity and magnetic data). If the config file contains wrong values, edit it manually or erase it and the questions will be asked again. This may be necessary if there are different data files with different characteristics in the same folder.

The data are then read and displayed in a full-screen window (Fig. 1). In addition, the program searches for a file called "lineaments.dat" which, if it exists, contains magnetic or gravity lineaments picked in former runs of pymagra. These lineaments are by default plotted onto the data map, but the plotting may be switched off later-on (see menu Display -> Plot lineaments).

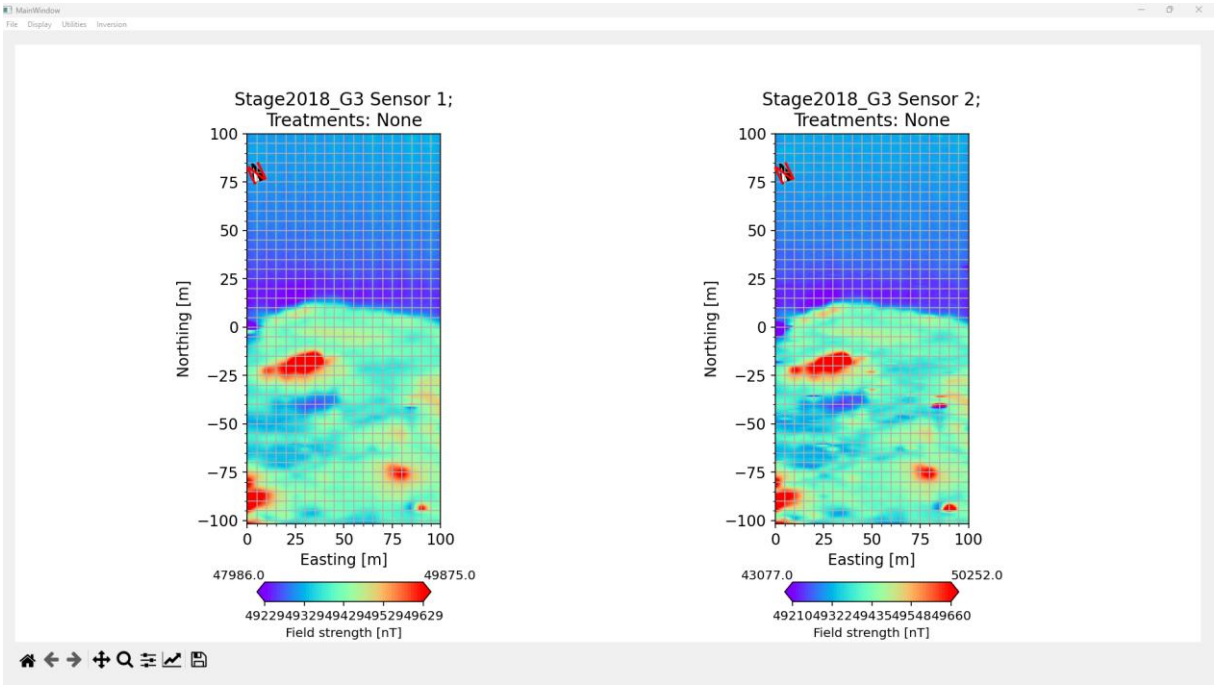

Fig. 1: Screen copy of the main screen with gradiometer data. Lines were measured in direction N30°.

The following treatments may be used:

1. Correction of diurnal variations for magnetic data (gravity data are supposed to be already corrected).
2. Data clipping and muting to eliminate measurement errors (again, mainly for magnetic data).
3. Time correction if instrument time was wrong and does not correspond to base station times
4. Reduction of directional effects in magnetic data.
5. Interpolation onto a regular grid, filling also internal gaps.
6. Extrapolation in order to create a full rectangular grid for use of FFT.
7. Pole reduction for magnetic data.
8. Vertical field prolongation.

9. Application of Spector and Grant method of spectral depth determination.
10. Tilt angle calculation of picking of lineaments.
11. Calculation and inversion of analytic signal.

In general, .stn or .out files do not contain gridded data. Also gxf, though it is a format for gridded data, may contain holes. Therefore, before applying some of the treatments (especially those from 6 to 11), data should be interpolated onto a regular grid. However, before gridding, diurnal variations and directional effects should be eliminated and, if necessary, wrong data (null measurements) should be eliminated or clipped. Gridded data are no longer associated to measurement times, which makes reduction of diurnal variations impossible and they are also no longer associated to original lines and therefore measurement directions and therefore directional effects can no longer be reduced. Data clipping is possible on gridded data; however, it is better done on the original ones in order to avoid overshooting effects on interpolation. This can be done in two ways: Or use “Utilities -> Clean data” to clip using absolute limiting values or quantiles, or – preferred by myself if there are not too many errors – using “Display -> plot single line”, scrolling through the lines and eliminating zero values and surrounding overshooting data using mouse wheel (see below).

Once data are interpolated all following treatments are done on the interpolated data. It is, however, possible to return to the original data, losing all treatments, and restart (see menu Utilities).

Menus

File

Get additional data files (keyboard shortcut: SHFT-D)

ATTENTION: this was programmed for an older version of pymagra. In the actual version, this point is not tested and will most probably make the program crash…

Allows adding additional data to the ones read in the beginning of the program. Attention, if any treatment has been done on the data read before, those data should be reset to the original values (see menu Utilities) before reading new data, since new data are added to the actual (i.e. potentially treated) data set. All files must have data measured in the same direction and with the same line spacing (not necessarily the same spacing along the lines). If the blocks are continuous within the measurement direction, the data should have been acquired on the same lines (not all lines must be occupied in both blocks, but, e.g., it is not allowed that in block 1, line coordinates are [0, 1, 2] m and in block 2 [0.5, 1.5, 2.5] m).

Save magnetic data .stn format (keyboard shortcut: S)

Save (eventually treated) magnetic data to Geometrics *.stn format.

Save .dat format (keyboard shortcut: F)

Save (eventually treated) data to simple text format, used by program mgwin.exe. This file has one header line, followed by one line per data point: X, Y, Value (optionally: second sensor value, vertical gradient). The format corresponds to the one exported by Geometrics Magmap program in Surfer format.

Get base station data (keyboard shortcut: SHFT-B)

Read base station data from file in Geometrics .stn format. The program only reads those data and plots them onto the screen in a floating window (Fig. 2), allowing interactive editing, but it does not do the base station correction (see menu Utilities). If file contains erroneous data

(spikes, noise…), one may choose segments to be cut out by clicking beginning and end times with mouse wheel or with right mouse button (Fig. 3). Left within the plot window closes the base station window.

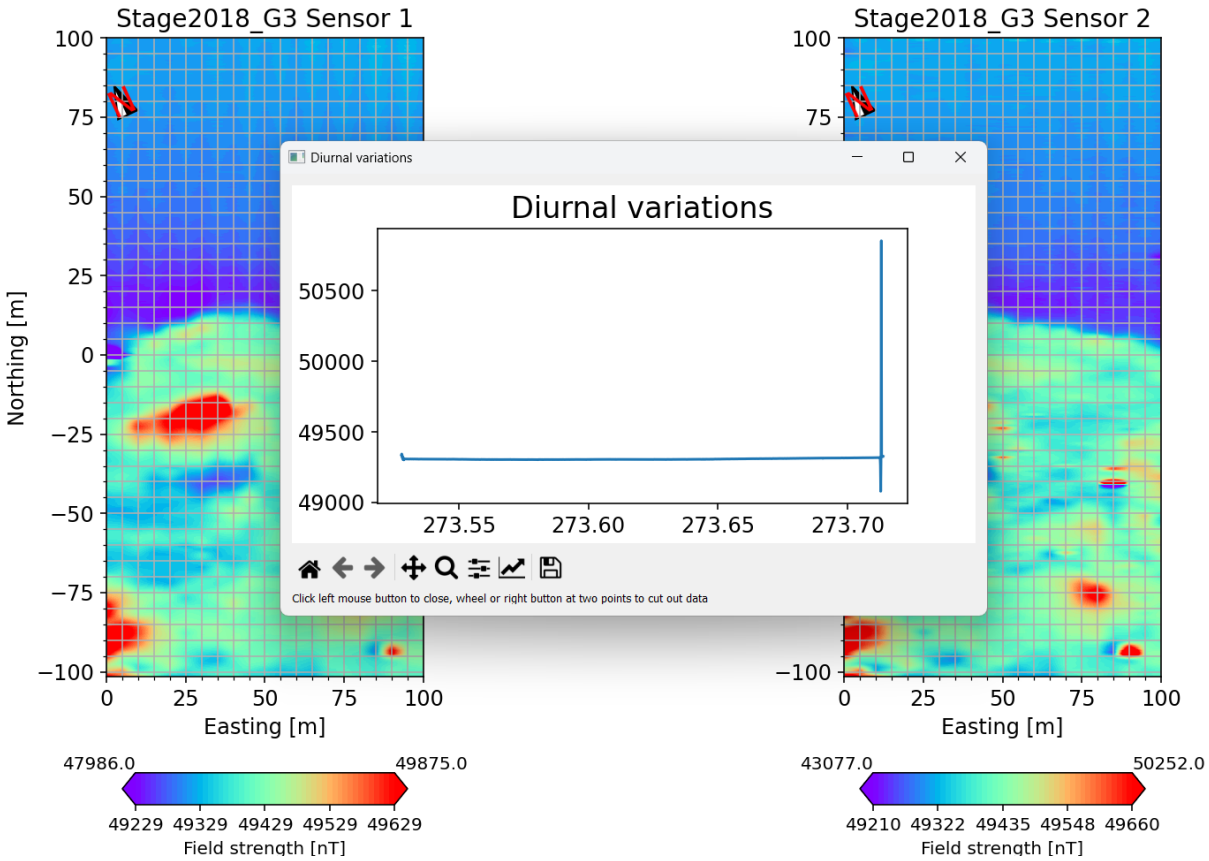

Fig. 2: Plot of read base station data with spikes.

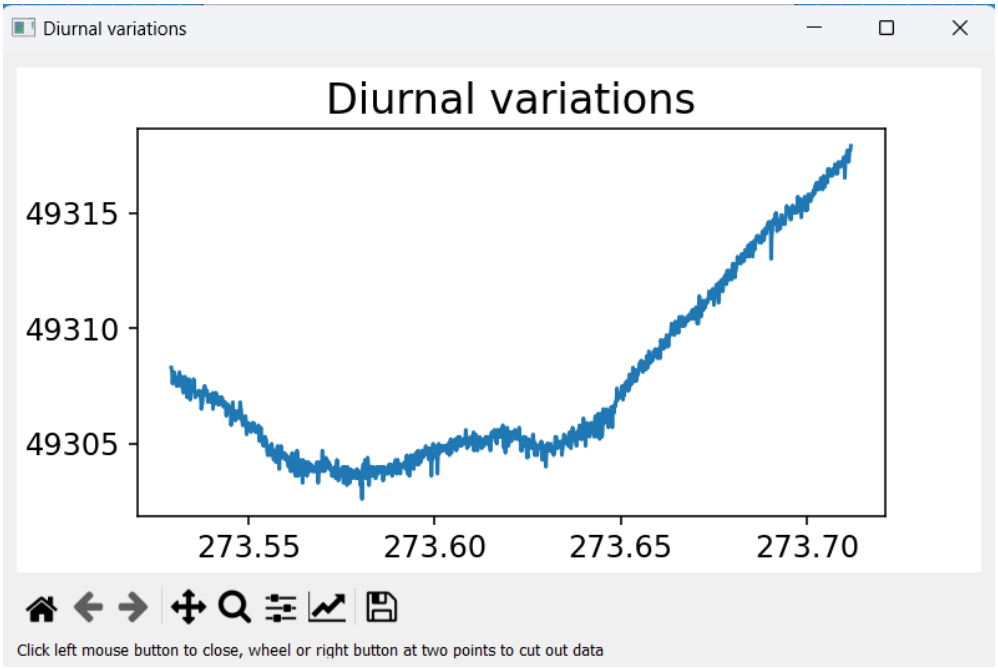

Fig. 3: Base station data having cut out the two spikes.

Save base data (keyboard shortcut: CTRL-B)

Save magnetic base station data to Geometrics ".stn" format. This option is mainly used in two situations: Or base station data were edited, or if no base station data were measured, but they were estimated from the measured field data using medians of the lines (see menu Utilities).

Save plot (keyboard shortcut: SHFT-P)

Save plot of main window into png file. The file name contains the date and time of file creation.

Get geography data (keyboard shortcut: SHFT-G)

Read geography and geology data to be plotted onto the maps (Fig. 4). The file is chosen interactively and has the following format:

Keyword (may be "#LINE", "#POINT" or "#END"; the first letter of Keyword is the hash tack)

If "POINT" one line follows with:

x y text

If "LINE", one line follows for every point defining the line:

x y

The file must be finished with a line containing #END in the first four columns

Usually, points are towns and text the name of the town. x and y must be given in the same coordinate system as the data. If lines limit a closed surface, the last point must be equal to the first one.

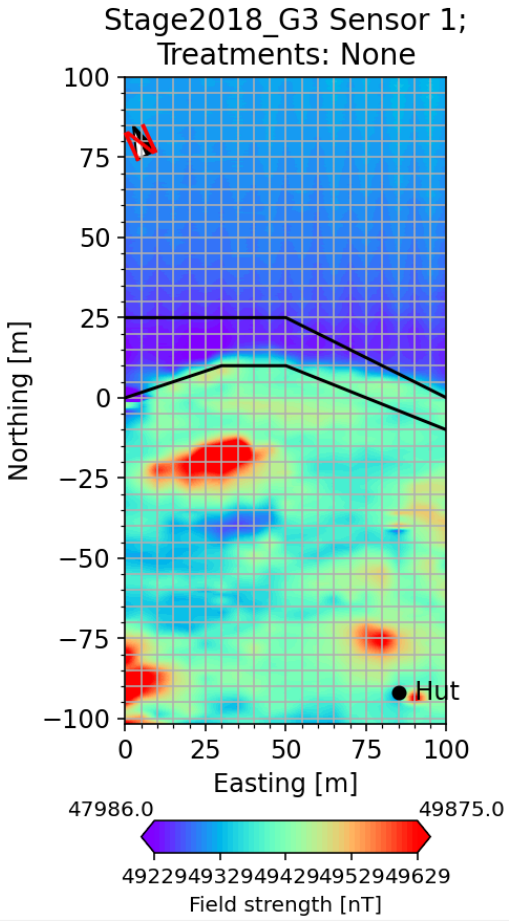

Fig. 4: Magnetic data with lineaments and geographic/geologic information

Quit (keyboard shortcut: CTRL-Q)

Finish program, the user is asked for confirmation.

Display

Plot original data (keyboard shortcut: O)

Plots the original (read) data. This does not reset the actual data arrays to original data. In the beginning of the program, a backup copy of the data is made. This backup is plotted. Further treatments continue to be done on already treated (time correction, interpolated…) data.

Plot treated data (keyboard shortcut: T)

Plots the treated data (actual data array). This option is only useful after having used option Plot original data.

Change data file (keyboard shortcut: D)

If several files have been opened (together at the beginning or using File -> Get additional files), this menu allows switching display and treatment from one file to another. The option is greyed if only one file is read.

Join data sets (keyboard shortcut: J)

If several files have been this menu allows joining all data into a single set (Fig. 5). ATTENTION: If the data have not been acquired within contiguous blocks, the program will interpolate meaningless data in the gap between blocks. The option is greyed if only one file is read.

The data used are the actual displayed ones, i.e. potentially treated data. It may be important to eliminate first directional effects for each measured block separately, may be also diurnal variations if no base station data exist, and only the join the data sets. When joining, the data are ordered by in-line coordinate. Therefore, the direction information will be lost, all data within a line are ordered from smaller to larger coordinates.

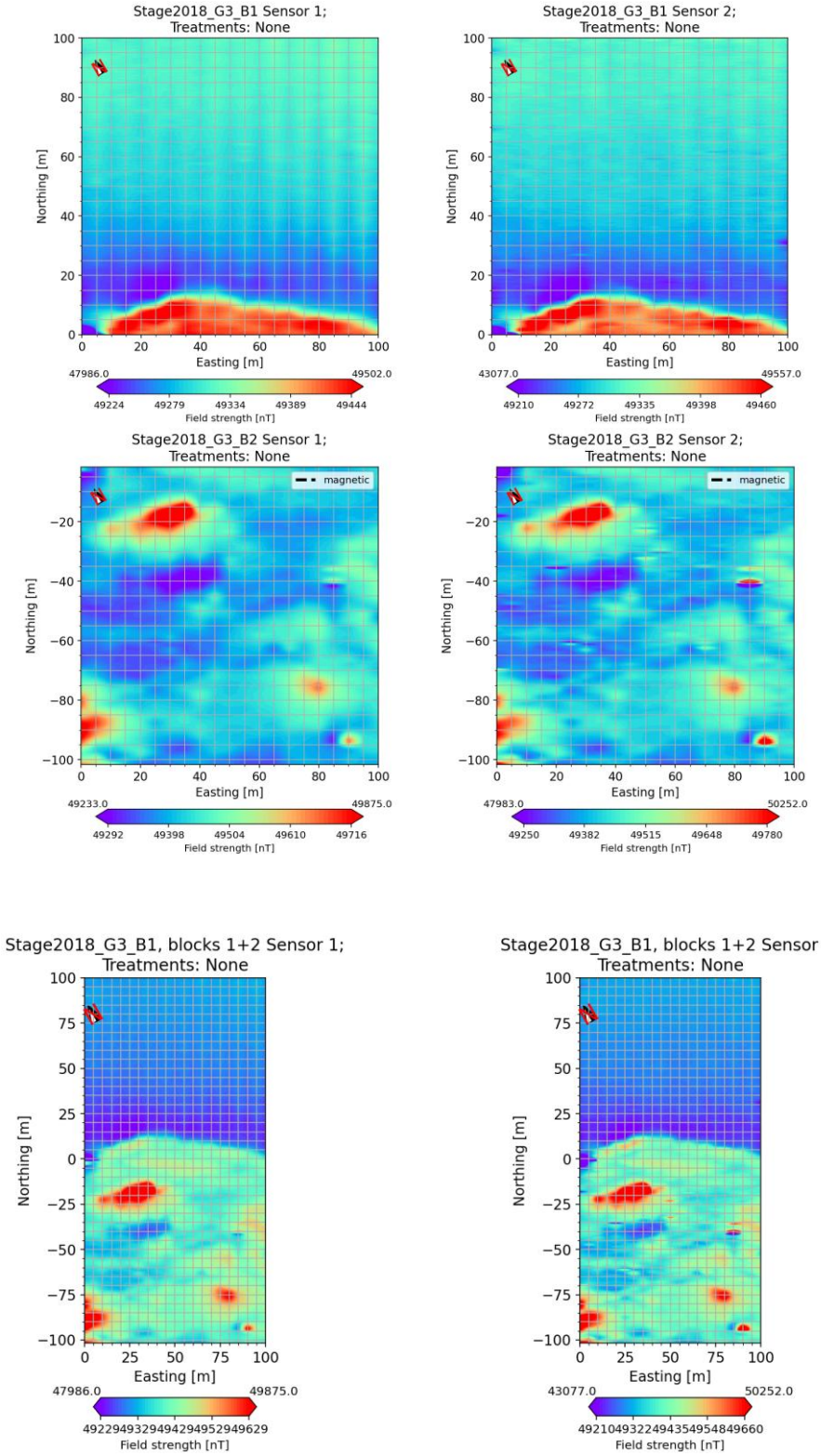

Fig. 5: Example of joining blocks. Upper row: data of block 1, middle row: those of block 2 and lower row: joint data sets. Attention: color scales of the plot are different!

Plot single line (keyboard shortcut: L)

Waits for a mouse click and plots the line nearest to the mouse click (Fig. 6). In the shown window, it is possible to mute data. For this click on the mouse wheel ones inside the axis (coordinate system) to choose the beginning of the section to be muted and a second time for the

end of the section to be muted. This procedure may be applied several times. Clicking the right mouse button inside the axis shows the next line to the right of the actual line (i.e. larger coordinates: to the East if lines were measured in N-S direction or to the North, if lines were measured in E-W direction). Clicking on the left button inside the axis goes to the next line in the other direction. Clicking any mouse button outside of the axis (outside of the coordinate system) closes the window and returns to the main window.

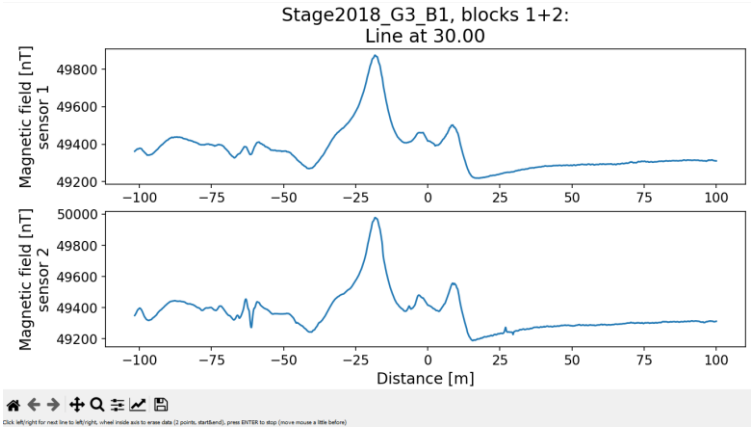

Fig. 6: Example of line plot

Plot base station data (keyboard shortcut: B)

Plots base station data (read or calculated, see menu Utilities->Apply diurnal correction). The time is given in Julian days since the beginning of the year of measurement or, if this information is not available, days from the first measurement. It is possible to mute erroneous data interactively in the new window. For this, click on the mouse wheel or the right mouse button at the beginning and the end of the section to be muted. This may be done several times. A click with the left mouse button or keyboard ENTER closes the window and returns to the main window.

Plot medians (keyboard shortcut: M)

Plots the median values of all lines (Fig. 7). The points are coloured as function of the sensor used in case of gradiometer data and as function of line direction (useful mainly for magnetic data having been measured in go and back directions). Pressing ENTER closes the window.

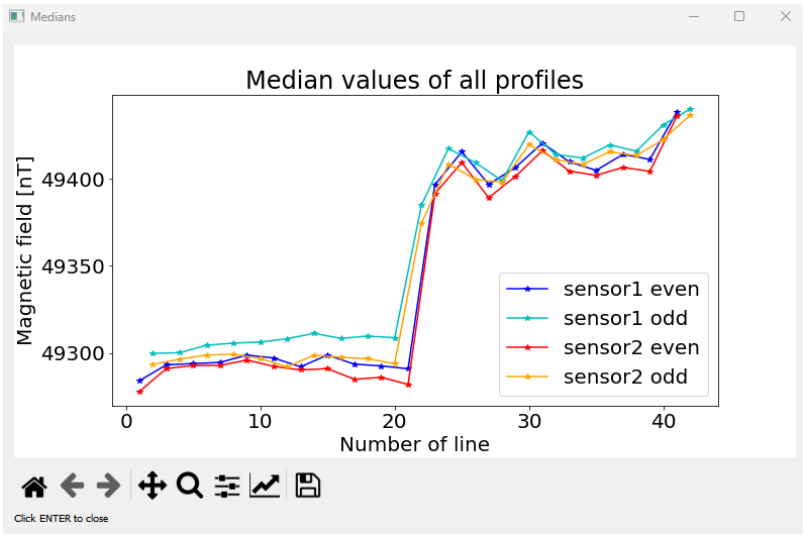

Fig. 7: Example of median values of gradiometer data measured in two directions.

Plot gradient (keyboard shortcut: G)

If gradiometer data are displayed, it is possible to show only the data of the two sensors or the two sensors and the gradient (Fig. 8). This option toggles between the two representations.

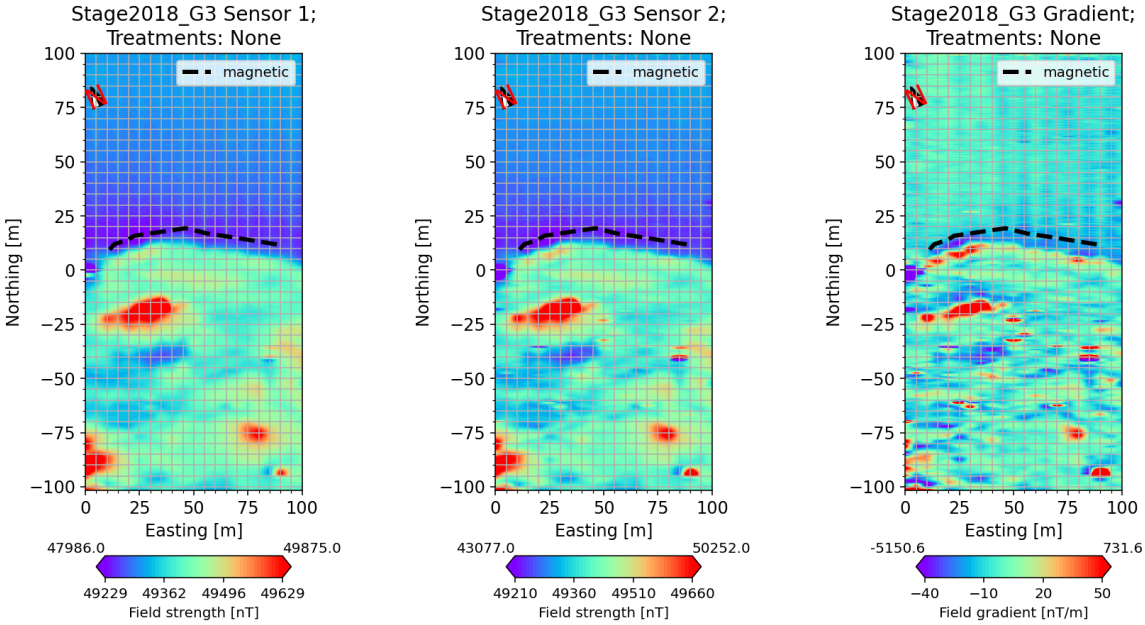

Fig. 8: Example of gradiometer data with vertical gradient.

Plot geography (keyboard shortcut: ALT-G)

If geography data were read (see menu File), this option toggles between displaying them or not.

Plot grid (keyboard shortcut: X)

Option toggles between showing or not grid lines of the minor ticks onto the maps.

Plot lineaments (keyboard shortcut: SHFT-L)

If lineaments were measured in this session or reading file lineaments.dat, this option toggles between displaying them or not.

Zoom

The different zoom options are not yet implemented. For the moment, the zoom should be done using the icon in the QT toolbox at the bottom of the screen. It is, however, not permanent and applies only to one subplot.

Change color scale (keyboard shortcut: Q)

By default, the color scale is constructed such that its lower end corresponds to the quantile 0.01 (the lowermost 1% of the data are clipped) and the uppermost to 0.99 (the upper most 1% of the data are clipped). This option allows changing this by setting a new overall quantile (valid for all sensors and the gradient) for clipping or by defining fixed upper and lower limits for each sensor and gradient separately. This allows, e.g., fixing a color scale for comparison of different maps. If fixed limits are given and the upper and lower limit are different, these limits have preference over a given quantile. If quantile is zero and fixed limits are both the same (usually both 0), the color scale limits correspond to the data limits for every sensor independently.

The color maps offered are “rainbow” (default), “viridis” and “seismic”, all linearly scaled between the minimum and maximum of the color scale. In addition, a special color scale called “special mag” is offered, which is blue for negative values, white at zero and red for positive values (mag,

since it is mainly thought for magnetic teaching purposes to show more clearly positive and negative anomalies for a simple body). This color map is linear within both domains (negative and positive), but non-linear overall. It does only give good results if positive and negative values exist, e.g. after diurnal corrections of magnetic data or having eliminated median values of magnetic or gravity data. If not, colors are purely red or blue shadings

Every second line (keyboard shortcut: 2)

Plot only one line out of two. This is useful if a strong directional effect is visible in magnetic data. The user is asked whether odd lines or even lines should be plotted (natural counting, i.e. first measured line is considered odd number). Clicking again on this option allows for plotting the other line direction or all lines again.

Utilities

Restart with original data (keyboard shortcut: CTRL-R)

In contrast to the option Plot original data (menu Display), this option undoes the treatments applied to the actual file type and replaces the actual data arrays by the backup arrays: If data have been gridded, the program goes back to the un-gridded data. You lose all treatments having been done on the gridded data, but keep those done on the un-gridded data (e.g. diurnal variations, elimination of directional effects…). A second click on this menu erases also those treatments and goes back to the original data set. All preparative treatments (e.g. diurnal variations of interpolation) will have to be done again. The program does not allow partial undo of operations on a given type of data (gridded or ungridded).

Clean data (keyboard shortcut: CTRL-K)

Possibility to mute data by defining different quantiles or different fixed values for the upper and lower data limits. In contrast to the option Change color scale (menu Display), this option set the values outside the defined limits to nan. The limits may also be defined on a histogram. The same limits are used for both sensors.

Correct time (keyboard shortcut: SHFT-C)

If instrument time was incorrect, it is possible to correct this time (may be necessary for base station correction). The program askes for the correct GPS time at one moment and for the time stored by the instrument at the same moment.

Apply diurnal correction (keyboard shortcut: CTRL-D)

If diurnal corrections were read from a file, subtract measured base station data from field data. If no base station data exist, a polynomial is fitted to the medians of the first sensor.

Reduce direction effect by median (keyboard shortcut: CTRL-M)

The directional effect of magnetic data is reduced by setting the median of every second line to the average value of the medians of the two adjacent lines (for lines at the edges, the same median value as the neighbouring line is used). The user may decide whether to change the medians of the odd lines or those of the even lines.

Reduce direction effect by Gauss (keyboard shortcut: CTRL-G)

See Masoudi et al., J. Geophys. Eng., 2023. The adjustment may be done on the odd-numbered lines or on the even-numbered ones (natural counting, first measured line is odd). In addition, the histogram adjustment may be done globally, i.e. all odd vs all even lines, or line by line, i.e. the histogram of a line is adjusted to the combined histogram of neighbouring lines. In the actual

version, data must be interpolated on a regular grid. Do this interpolation such that in the cross-line direction, the sampling step is equal to distance between the lines.

Interpolate (keyboard shortcut: CTRL-I)

This option interpolates data onto a regular 2D grid parallel to the measurement lines. The user is asked for the interpolation step dx and dy parallel to the local coordinate system. For each line, the positions of the first and last measurement are searched and interpolation is limited to the area between these measurement points, rounded to the next multiple of the corresponding sampling step. In a second pass, interpolation is done in the sense perpendicular to the measurement direction with data spacing equal to line spacing. In this way, if there are missing values at the beginning or the end of certain lines inside the measurement area, they will be interpolated. However, missing points at the edges of the area (which do not have neighbouring data on both sides in the measurement direction and in the opposite direction), are not extrapolated. I.e., a convex zone of data is created, presenting possibly zones without data in the corners. Also, if too many data are missing within a line or on lines between them, they are not interpolated. The limit is set arbitrarily to 10*dy within a line and to 3 missing lines in the cross direction. If you want to change this, search for the occurrences of “10.*dp” (inline) and “pos_ind[:-1] > 3” (cross-line) in file ./utilities/utilities.py.

Fill rectangle (keyboard shortcut: CTRL-F)

Extrapolate data in the corners in order to fill the full rectangle with values. Nearest neighbour algorithm is used with squared distance weighting.

The following figures 9 to 14 show a typical sequence of data treatment for data without base station and a strong directional effect. These figures are from another data set to better see the effects.

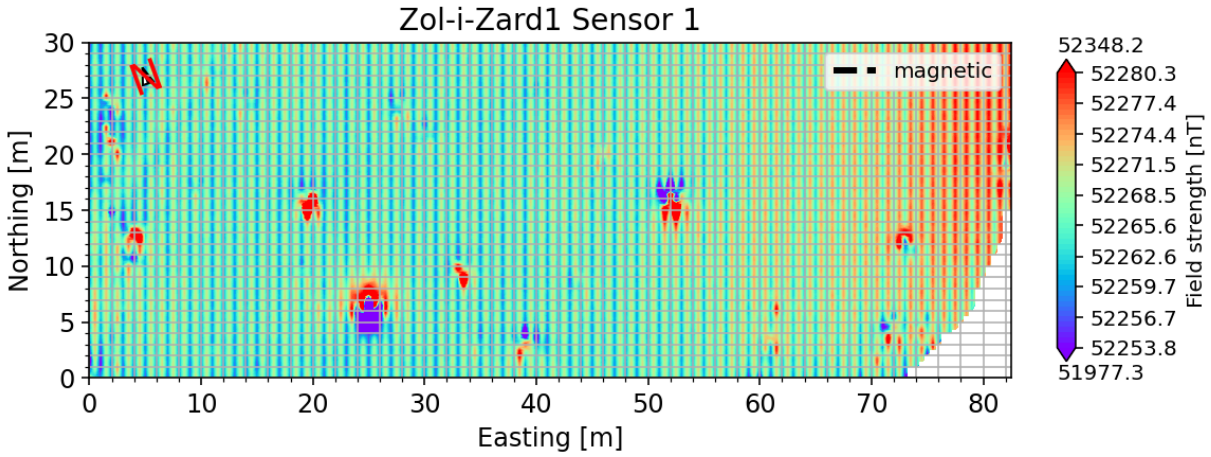

Fig. 9: Original data

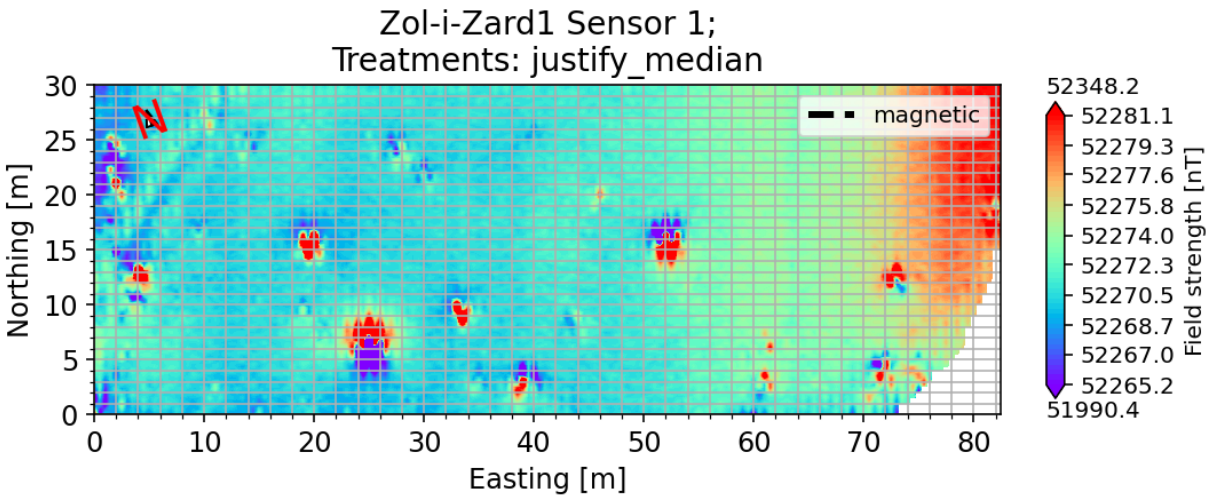

Fig. 10: Data after elimination of directional effects by median adjustment.

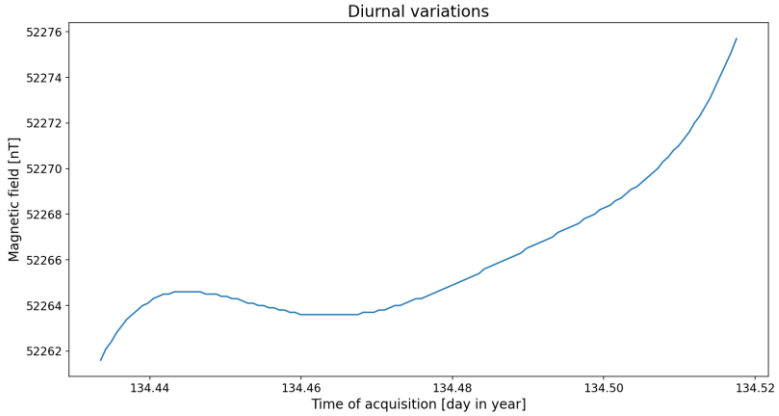

Fig. 11: Estimation of diurnal variations based on fitting a polynomial of degree 5 to median values of lines

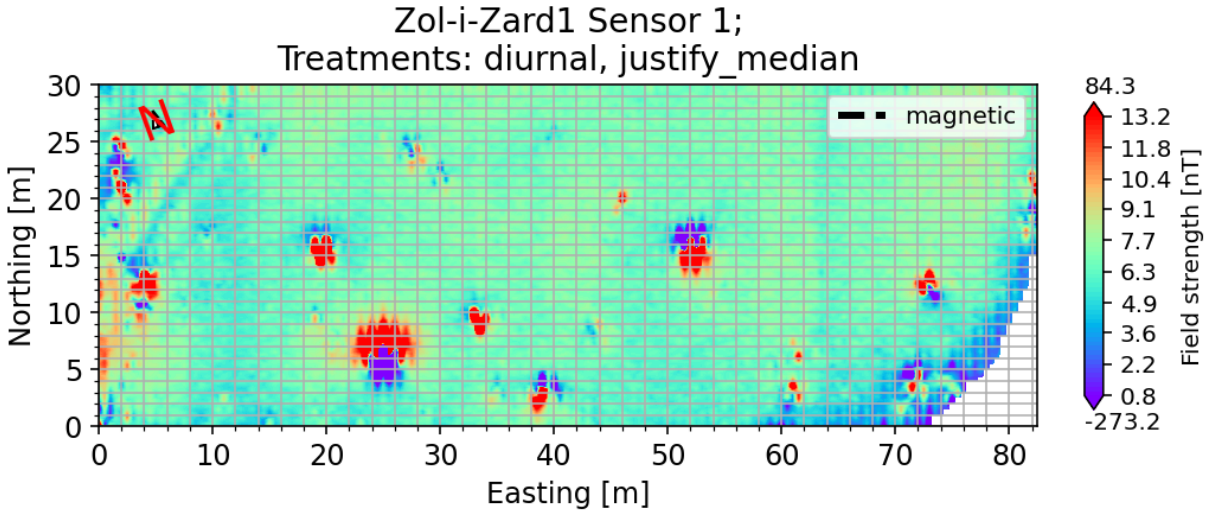

Fig. 12: Data after elimination of directional effects and elimination of diurnal variations

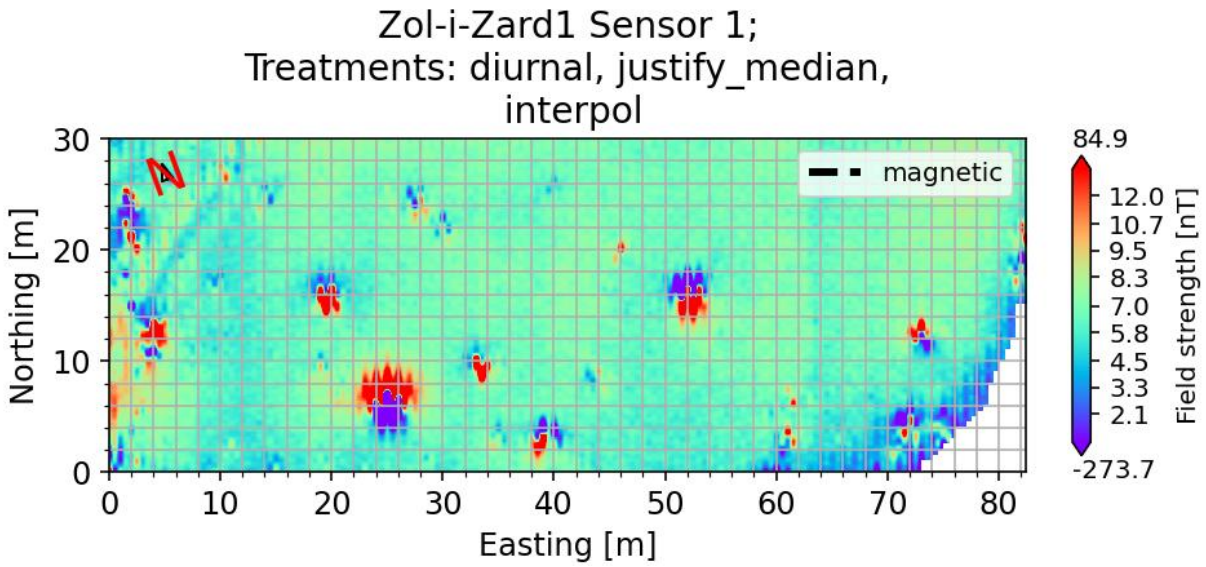

Fig. 13: Data after former treatments interpolated on regular grid (hardly any visible difference with Fig. 12)

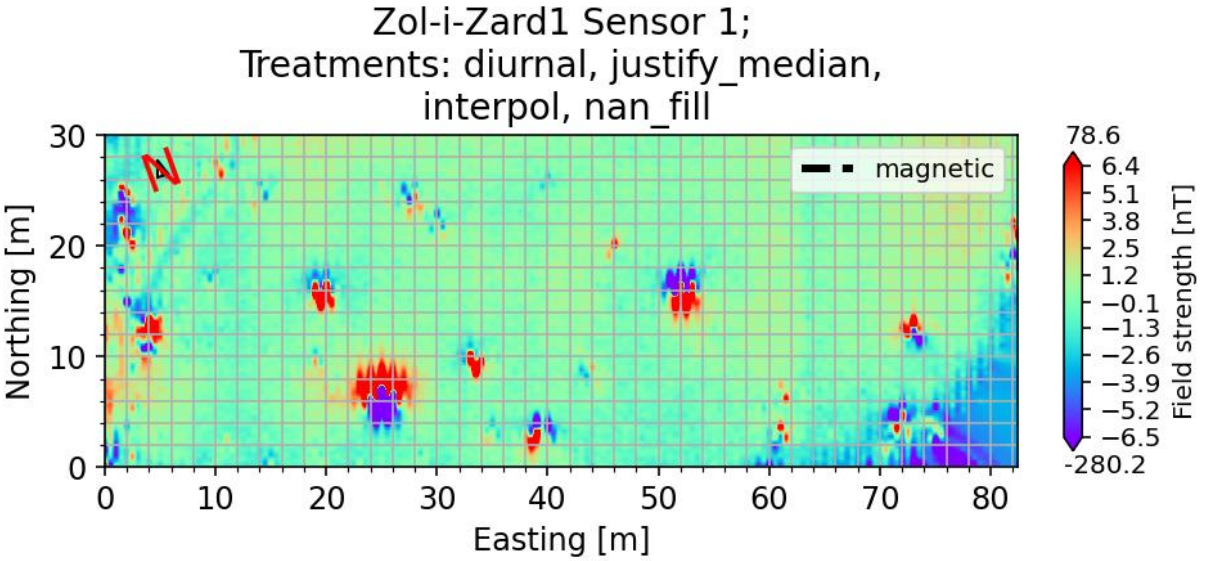

Fig. 14: Data grid filled with extrapolated values in the SE corner

Pole reduction (keyboard shortcut: CTRL-P)

Data are reduced to pole (Fig. 15). Algorithm considers only direction of Earth's field, i.e., magnetization is considered to be parallel to Earth's field. Earth's field direction had been given when starting the program. After pole reduction, the values that were nan before application of "Fill rectangle" are set again to nan.

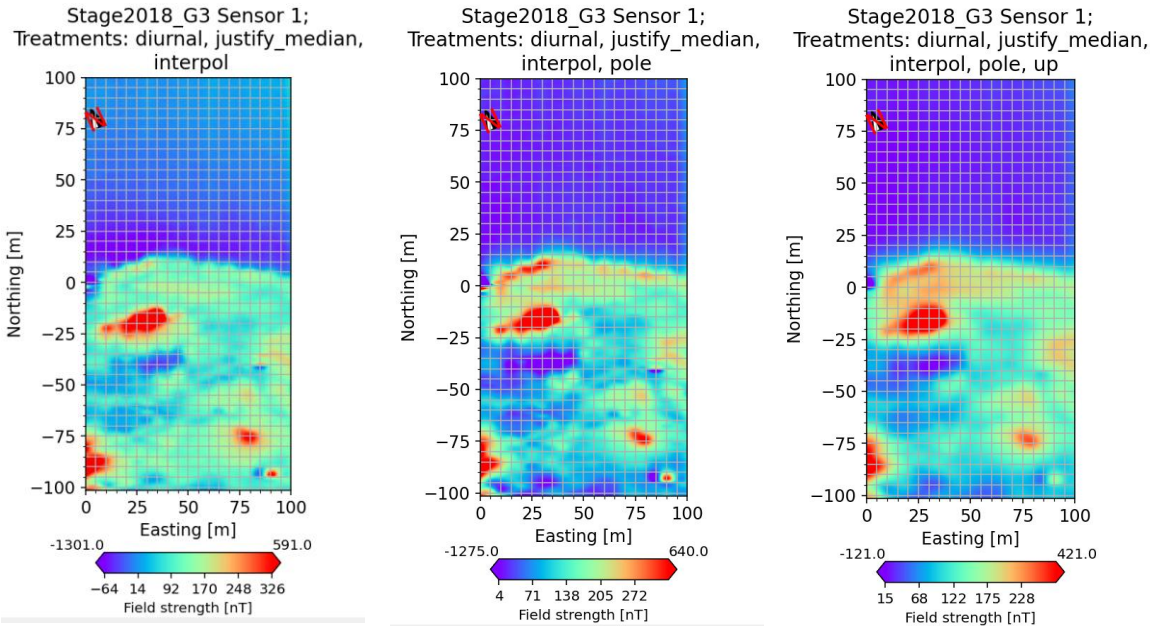

Fig. 15: Left: Data after former treatments. Center: Data of left panel reduced to pole. Right: Data of central panel upward continued by 2m.

Upward continuation (keyboard shortcut: CTRL-U)

Data are upward continued by a user given distance (Fig. 15). Downward continuation is also possible by giving a negative distance, but it becomes rapidly unstable. If the value (given in meters) is positive, data are continued upward, if it is negative, continuation is downward.

Tilt angle (keyboard shortcut: CTRL-T)

Calculate tilt angle (Fig. 16) and search maxima of its gradient (Fig. 17) (Miller & Singh, JAG, 1994). In addition, program plots first and second vertical as well as horizontal gradients.

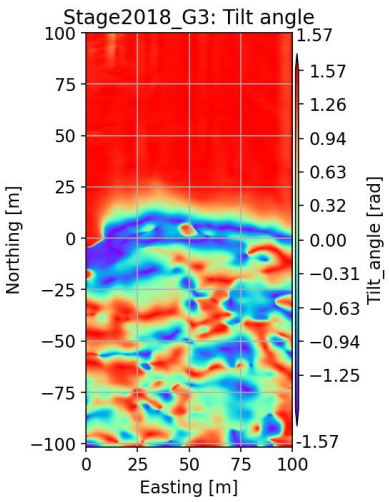

Fig. 16: Tilt angle of data shown in Fig. 15

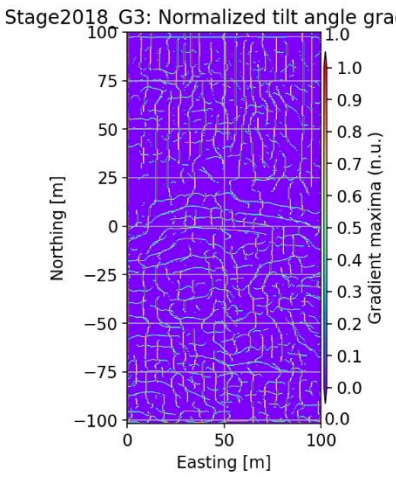

Fig. 17: Maxima of tilt angle

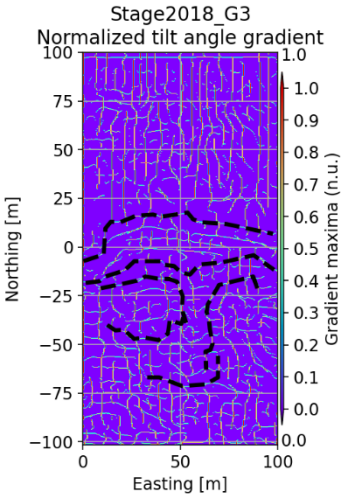

Fig. 17a: Some lineaments Traced interactively

Once the tilt angles are calculated, its values are smoothened with a 2D Gaussian filter with a size of 5 cells, the absolute horizontal gradient is calculated and local maxima are determined with a

half-width of 3 cells (i.e. a maximum is found if the value at [i,k] is larger than all values of the next 3 cells in x and y directions. These maxima are plotted on a third figure, which appears on top of the other two, hiding them (Fig. 16):

On the resulting map, lineaments may be visible and may be traced. For this, left click the points defining a lineament and finish line with a right click (position of right click is not considered to belong to the lineament). You may continue choosing further lineaments. A right click not preceded by a left click finishes picking and returns to the main window. A click with the mouse wheel searches the nearest traced point and erases the corresponding lineament.

The measured lineaments are written into file "lineaments.dat" and are displayed on all maps, if this option is not disabled in menu Display. Depending on the data type used for tracing a lineament, different names are associated to it, stored with the coordinates in file lineaments.dat and are used for legend on the maps.

Analytic signal (keyboard shortcut: CTRL-H)

Calculate analytic signal along measurement lines (Nabighian, Geophysics, 1972) and plot a map with these analytic signals. The results are given with respect to the Earth's surface (it is supposed that the "height of lower sensor" asked for when starting the program is the height above ground and all subsequent treatments (upward continuation, return to original data) are added to this height). On the map, a left click triggers interpretation of analytic signals along N-S lines, a right click along E-W lines and a wheel click finishes the module. If solutions were found, they will then be stored in file "Analytic-signal-solutions_*dir*-lines.dat" where *dir* is "N-S" or "E-W" depending on the chosen direction of interpreted lines. The line nearest to the click will be shown later in an extra window.

In a next step, the lower part of the cumulative sum of all analytic signal data is shown and the user should click at the value below which the signal is no longer interesting (threshold). Typically, there is a large number of very small values and only in the order of 1% or so data are useful. Then, the chosen line is shown with the solution. A click into the window closes it and all solutions are shown on the map of the analytic signal. If the user estimates that not enough or too many solutions were found, it is possible to restart with a click on the map of the analytic signal.

Next, the user has the possibility to change manually the threshold (e.g. to maintain the same one if the calculations are done in both directions) and to indicate the maximum and minimum accepted depths. These depths are given with respect to the Earth's surface. It is thus possible to eliminate all results above ground and the maximum height may mainly limit the color scale for plotting the results (a few of the results are often quite large and don't make geological sense).

The inversion of the analytic signal for depths is done as follows: the local maxima along a line for all values above the threshold are searched from every maximum, the program searches to both sides or the first value below the threshold or the next local minimum. The so found data are used for inversion. A regression line is calculated for the equation y =a*x+b where y = 1/analytic_signal, x=(x-x0)**2, a = 1/alpha**2 (see Nabighian) and b=(depth/alpha)**2.

x0 is the location of a body limit. This position is located at the position of the maximum of the analytic signal. Since the maximum does not necessarily coincide with the maximum of the calculated signal, x0 is varied between the position of the measured maximum plus/minus dx/2 (dx being the sampling step in the direction of the line). In addition, the data are weighted by their squared value so that the highest values should be better fitted than small values. The values x0, alpha and depth giving the best fit to the data around every maximum are stored.

Spector 1D (keyboard shortcut: SHFT-F)

Calculate spectrum of all measured lines and determine automatically depth of sources (Spector & Grant, Geophysics, 1970). The user is asked for the direction of line extraction (by default the one of the measurement direction) and the half with for the determination of the local maxima along the spectra. This value determines the distance over which a value must be larger than neighbouring ones (1 means it is larger than the immediate neighbours on each side; 2 means it is larger than the two next samples on each side…). The slopes of the logarithmic spectrum, which are proportional to the depths of source rocks are calculated using these local maxima and give the average depth for the whole line. Two lines are fitted to these maxima, allowing for a break in slope. The program shows the results of all lines in a common plot (Fig. 18).

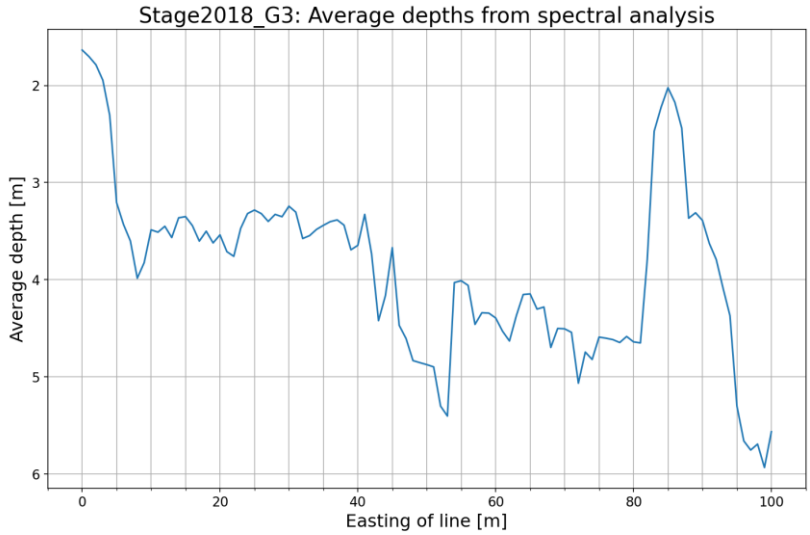

Fig. 18: Example of depth inversion results using line spectra of all lines in N-S direction.

The automatic fits are not always satisfying, especially if the data would need at least three different slopes. Therefore, the user may now click with the left mouse button on one point of the results. The spectrum of the line nearest to the clicked position will show up (Fig. 19) and the first slope may be modified manually by clicking at the beginning of the line with the right mouse, pulling the line and releasing the mouse button at the end of the desired slope. The depth is immediately corrected in the plot of the results. This procedure may be applied to as many lines as desired. A right click within the results plot finishes the procedure and returns to the main window. The resulting depths are then stored into file “spector.dat”.

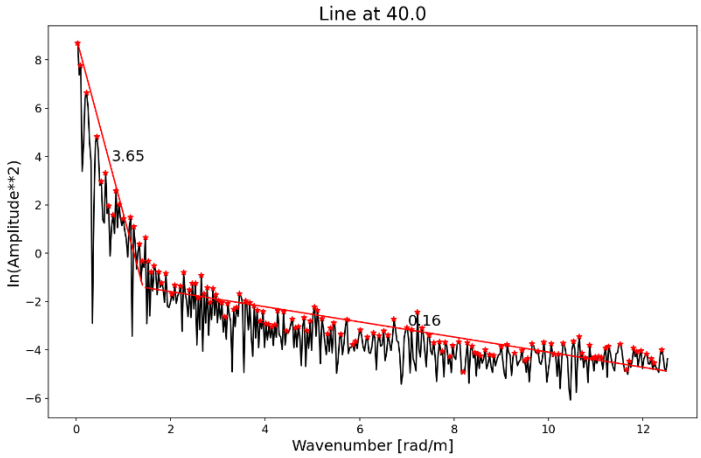

Fig. 19: Example of line spectrum in black with automatically fitted slopes in red. The red asterisks are used for line fitting. The best breaking point is in this case located at wavenumber 0.6 rad/m. The numbers written at each line are the calculated depths in meters.

Spector 2D (keyboard shortcut: ALT-F)

Calculate spectrum moving windows along all measured lines and determine automatically depth of sources (Spector & Grant, Geophysics, 1970). The user is asked for the window length (default is ¼ of the shortest dimension), the step size of calculation points (default is calculated such that the number of points to be calculated is about 500 to limit the calculation time) and the half with for the determination of the local maxima along the spectra. This value determines the distance over which a value must be larger than neighbouring ones (1 means it is larger than the immediate neighbours on each side; 2 means it is larger than the two next samples on each side…). The slopes of the logarithmic spectrum, which are proportional to the depths of source rocks are calculated using these local maxima and give the average depth for the whole line. Two lines are fitted to these maxima, allowing for a break in slope. The calculation is done twice at every point: once for data in N-S direction (half of the window width to the south, half to the north of the calculation point) and once in E-W direction. The retained depth is the average of the two results. The results are plotted as map with pixels centred on the data points (Fig. 20).

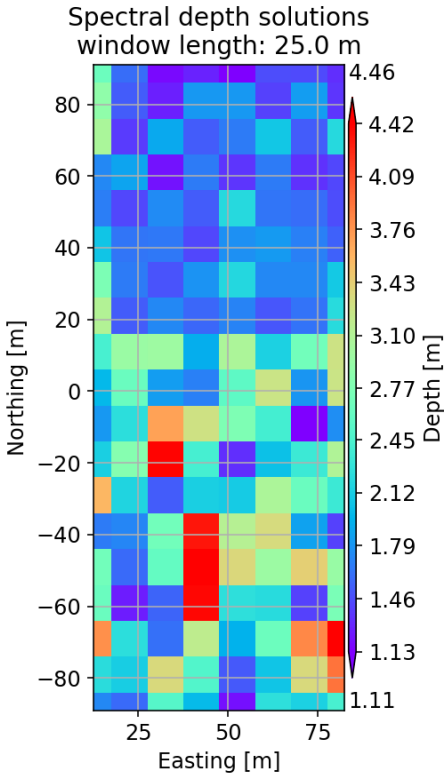

Fig. 20: Example of windowed depth calculations.

The user may then click with the right mouse to finish the module and return to the main screen or click with the left mouse on any point. In the latter case, the two spectra (NS and EW directions) are plotted with the fitted lines (Fig. 21). The user is then asked to trace a new line to manually define a new depth or to make a simple click to accept, and this for both directions (i.e., to accept the automatic results and return to the map of solutions, two clicks are necessary).

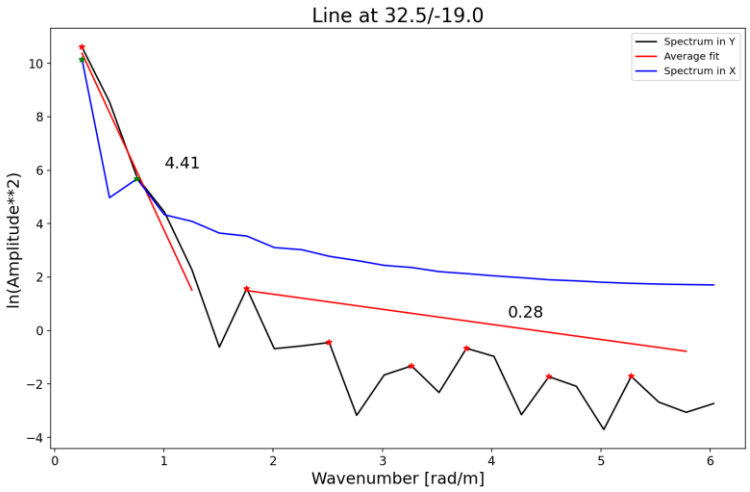

Fig. 21: Spectra of one point in NS and EW direction shown after clicking on a point of Fig. 20

Inversion

2.5D inversion: (keyboard shortcut: I)

Do inversion in 2.5D of one line of magnetic or gravity data using a self-adaptive model of rectangular prisms. It is possible to use original, not interpolated data or data interpolated onto a regular grid. For the inversion procedure see 3D inversion.

The user must first click onto the data map to choose the line. If data are not interpolated, the nearest line is chosen. If data are interpolated, a left click chooses the nearest line in Y direction, a right click chooses a line in X-direction.

A first and second dialogue box contain the same information as for 3D inversion, see there for explanations.

Then, a third dialogue box opens, asking for the data to be interpolated. The data may be reduced in two ways: by defining the part of the line to be inverted and by reducing the density of points (choosing one point out of n points). In addition, the program asks for the width of the prisms perpendicular to the line. All prisms have the same width. The width may be defined by observing the width of anomalies on the map. In addition, one may use topography (if those data exist; in the actual version of the program only for BRGM or TXT data) for the upper limit of the model and for the height of measurement points by activating the corresponding box at the bottom of the dialogue box.

ATTENTION: If after iteration 0 the program stops, it is most likely that there is a message box open, hidden behind other windows, suggesting to increase smoothing or regularization parameters.

At the end of the inversion, the program shows in the top frame the measured data ("+") and the calculated data (continuous line). In addition, if two sensors have been used, sensor-one data are plotted in blue/cyan, whereas data from second sensor are plotted in red/orange. In a second panel the vertical cross-section of the model is represented with colouring as function of the prism properties (Fig. 22). Initially, the cross section is not scaled. As long as the window is open, pressing the keyboard key "e" toggles between **e**qual horizontal and vertical scales and filling the window such that the horizontal and vertical scales are different (Fig. 23). Keyboard key "c"

toggles between logarithmic and linear **c**olor scale of body properties. Keyboard key “r” allows to continue (**r**etake) inversion. The program proposes automatically 10 iteration steps more. Close the window by pressing ENTER. Also the misfit evolution is plotted (Fig. 24)

The plot is saved in a png file containing the line position in its name. In addition, the following files are recorded, some only for testing purposes, all in a new subfolder called 2Dmodel_date_time (date is the date of inversion run, time the time of the end of the run):

- model2D.dat contains the prism properties of the final model
- parameters.dat contains all inversion control parameters used for this run
- synthetic_model.dat contains the prism positions and properties for calculation of the model as synthetic model (see later, p.22).

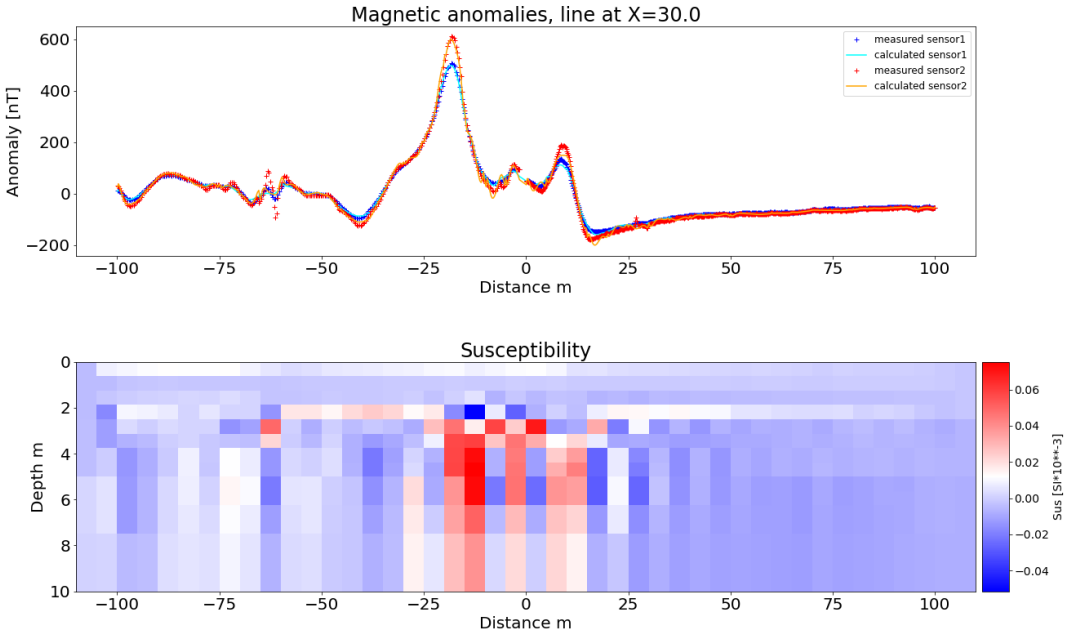

Fig. 22: Example of 2.5D inversion result.

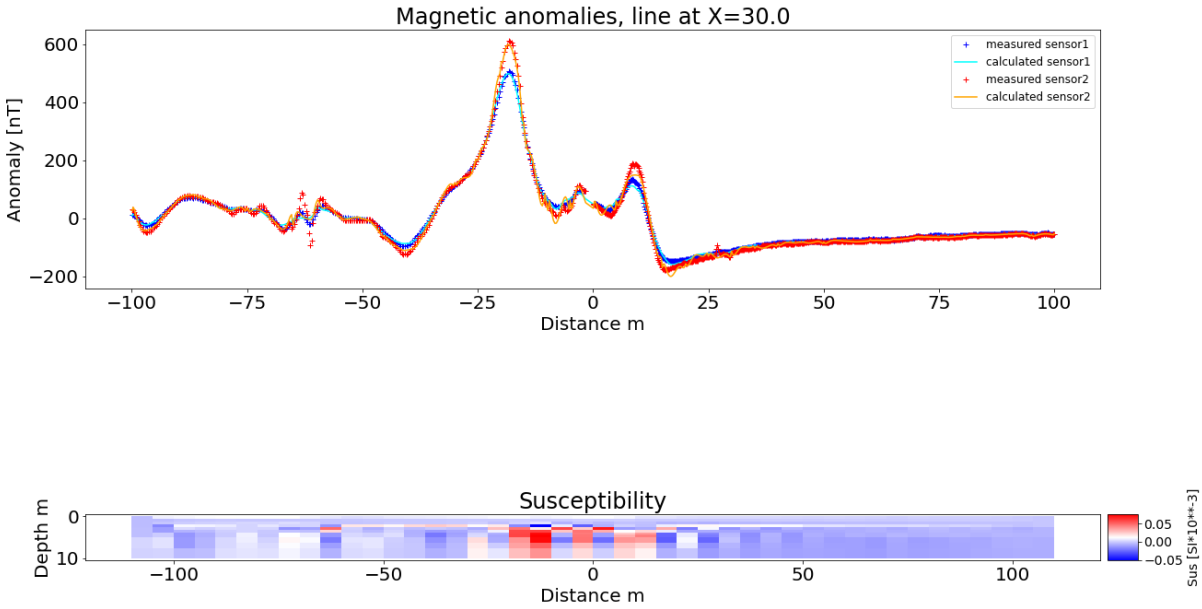

Fig. 23: The inversion result with vertical scale = horizontal scale.

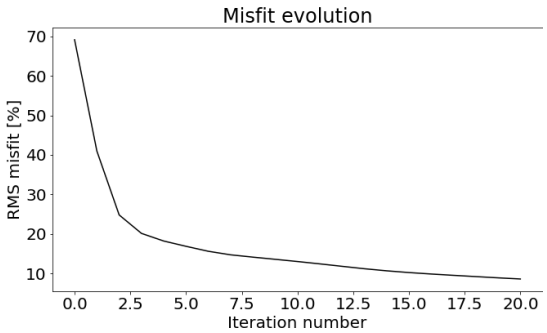

Fig. 24: Misfit evolution of the presented inversion.

3D inversion: (keyboard shortcut: 3)

3D inversion of magnetic or gravity data using a self-adaptive model of rectangular prisms. To use this tool, the data must first be interpolated onto a regular grid (Utilities -> Interpolation).

In a first panel, the user gives information concerning the conditions to stop iterations as well as data smoothing and matrix regularization. The iterations may be stopped in four ways:

- User-defined maximum number of iterations is reached
- The number of prisms becomes larger than the number of data
- The rms-misfit becomes smaller than a user-defined value
- The variation of rms-misfit from one iteration to the next becomes smaller than a user-defined value

The user gives in addition a factor lambda for matrix regularization (adding lambda to the diagonal of the Frechet matrix) and a factor gamma for smoothing (minimizing the parameter variation between neighbouring prisms). Both these factors may diminish from one iteration to the next. In the beginning, while the model is still far from optimum, regularization and smoothing may need to be important, whereas, when approaching the optimum model, they may become less important. Also the minimum values for these parameters is defined by the user.

In addition, the user may activate a positivity constraint, i.e. all parameters (susceptibility, remanence or density) of all prisms must be positive. See appendix 2 for more information.

ATTENTION: If after iteration 0 the program stops, it is most likely that there is a message box open, hidden behind other windows, suggesting to increase smoothing or regularization parameters. The smoothing or regularization should not be too small in order to have an effect, but also not too large not to dominate the inversion process. Have also a look at the consol from where the program has been launched: there, the maximum values of the different matrices are indicated for information.

In a second panel, the user is asked for several other inversion parameters:

- The uncertainty of data values (constant value for all data)
- The parameter variability (kind of uncertainty of parameters)
- A reference depth z_ref for weighting parameter influence (for magnetic data, the sensitivity is supposed to decrease proportional to (z/z_ref)**3, whereas for gravity data, it is (z/z_ref)**2. Without this regularization, usually deep prisms will get very strong parameter variations.
- The number of data points around a given point i for which the value at i must be larger than all other neighbouring values to be considered as local maximum

- The amplitude of absolute misfit with respect to the strongest detected misfit for which prism will be subdivided.

In a last panel, the user is asked for the area to be treated. It is thus possible to concentrate on a partial zone and in addition, it is possible to reduce the number of data points to be used by taking only one point out of several ones (same reduction in both directions). The area is given by defining the extend in 3D and size of prisms in the initial model. The data are then chosen such that the prisms extend by the maximum model depth further in all directions (e.g. if initial prisms are defined in an area from 100 to 500m in X and Y and up to a maximum depth of 50m, the data are extracted from 150 to 450m in X and Y).

The prism properties (susceptibility, remanence (for the moment only susceptibility or remanence parallel to the Earth's field or density are supported) are set to 0.001 SI for susceptibilities, 0.1 A/m for remanence or 10 kg/m^3 for density. During the inversion process, the program calculates the effect of the actual model and searches the maxima of absolute differences with measured data. For the maxima having a value of up to 10% (or another user defined value) of the extreme maximum, the program searches the prism having the strongest effect on this data point. The corresponding prisms are subdivided into 2x2x2 prisms (i.e. their size is halved in all directions) and a new optimum model is calculated. The prisms are subdivided until a minimum size is reached, defined by the user.

At the end of the inversion, the program plots horizontal maps of parameter distribution and prism limits, the calculated data and their difference with the measured data (calculated minus measured) and the misfit evolution. The parameter maps are plotted in the vertical center of the smallest programmed prisms (e.g., smallest size in Z direction is given as 10m, then the plots are placed at 5m, 15m, 25m depth etc.). Pressing ENTER having one of the result windows in the front closes all result windows. All plots are saved in png files. In addition, the following files are recorded, some only for testing purposes:

- model3D.dat contains the prism properties of the final model
- par_hist.dat contains the prism properties after every iteration (prism number, value and center coordinates
- prism_control.dat contains the history of all prisms being subdivided
- parameters.dat contains all inversion control parameters used for this run

The inversion parameters are quite tricky to be used. It will take a number of tests to find the best values for your data set.

Synthetic model: (keyboard shortcut: SHFT-S)

Calculation of a synthetic model. The model may be given in a file or entered interactively. It is composed of rectangular prisms with faces perpendicular to the axes. The calculation is done in agreement with the data. I.e., if data are gravity, gravity anomalies are calculated. If data are magnetic, magnetic anomalies are calculated using the same line orientation and regional magnetic field components (strength, inclination and declination) as the presented data.

The program asks first for a file to be opened. This file must have the following format:
One line per prism containing
x1, x2, y1, y2, z1, z2, sus, rem, rem_i, rem_d, rho where
x1 and x2 are coordinates of the faces perpendicular to the x-axis in meters (left and right)

y1 and y2 are coordinates of the faces perpendicular to the y-axis in meters (lower and upper)
z1 and z2 are coordinates of the faces perpendicular to the z-axis in meters (positive downwards, smaller depth followed by larger depth)
sus is susceptibility [SI units]
rem is strength of remanent magnetization [A/m]
rem_i is inclination of remanent magnetization [degrees]
rem_d is declination of remanent magnetization [degrees]
rho is density of declination.

If no file is chosen or if there is an error reading the file (e.g., not enough columns), a dialogue box is opened asking for the positions and properties of the prisms. For every prism, a new dialogue box is opened. If Cancel is pressed, the prism definition finishes without taking into account the values shown in the last box.

In any case, after entering the prisms, a last dialogue box opens asking for the points where the synthetic anomalies should be calculated. By default, the area of existing data is proposed with a grid step allowing approximately 50 points in each direction.

After finishing the calculation, the resulting map is shown on the screen and saved into a png file prefix_synthetic.png where prefix is “Mag” or “Gravi”. Clicking any mouse button within the axis closes the figure and saves the calculated data into file "synthetic_data.gxf".

History:

v.24.8.1: first beta version

Aug. 2024: An earlier version from 2023 was completely rewritten to be publishable on pypi.

Sept. 2024: Integration of inversion

Since then, versioning exists.

APPENDIX 1: Data structure

In main program every data block is stored in self.dat[i] (list of objects of class DataContainer)

The actually used data set is located in self.data

class DataContainer is structured as follows:

ATTRIBUTES

Principal attribute: **self.data** which is a dictionary with the following items:

One item for every measurement line.
Key: number of line (int) starting with 0, value: dict
Items in this dictionary:
"s1": numpy 1D float vector with data of sensor1
"s2": numpy 1D float vector with data of sensor2 (if only 1 sensor used, all values are 0.)
"x": numpy 1D float vector with X-coordinates of data
"y": numpy 1D float vector with Y-coordinates of data
"z": numpy 1D float vector with Z-coordinates of data (usually only zeros)
"time": numpy 1D float vector with measurement times in seconds from beginning of month
If data not Geometrics and no times are given, they are constructed from the time of DataContainer creation onward, every line starting at a full minute and time step being 0.1 seconds.
"mask": bool; if True: data will be plotted, else not. (E.g., to choose every second line)
"direction": Direction of the line with respect to the Y axes. May be 0 (N), 180 (S), 90 (E) or -90 (W). Absolute direction is then direction+line_declination (see below)
"sensor": int; if 0: 2 sensors, in vertical disposition.
"median1": float; Median value of sensor1 measurements of line
"median2": float; Median value of sensor2 measurements of line

Other items of dictionary data:
"grad_data": bool; True if two sensor exist in vertical disposition
"year": int; Year of acquisition
"dispo": int; 0=vertical disposition of sensors if there are 2, 1: horizontal disposition
"block": int; Number of block (read file or joint data sets)
"block_name": str; name the block, especially for joint data sets
"height": float; Height of sensor 1 above ground.
"height2": float; Height of sensor 2 above ground (=height1 if only one sensor).
"d_sensor": float; Distance between sensors (height minus height2)
"line_declination": float; angle between local Y axis and North (positive from N to E)
"title": str; Title shown on most plots
"type": str; Data type, may be "magnetic" or "gravity"

n_data : int; total number of data; all subsequent arrays have length n_data
n_lines : int; number of lines
h1_sensor : float; height of sensor1 above ground
h2_sensor : float; distance between sensors (h_sensor-h2sensor)
d_sensor : float; height of sensor1 above ground

dispo : int; 0: sensors in vertical disposition, 1: sensors in horizontal disposition
line_declination : angle between local Y axis and North (positive from N to E)
title : str; Title shown on most plots
sensor1 : numpy 1D float array; contains all data of sensor1, flattened such that data of line 1 are followed by those of line 2 etc.
sensor2 : numpy 1D float array; as sensor1, but data of sensor2 (zeros if no second sensor)
x, **y**, **z** : numpy 1D float arrays; coordinates of every data point
grad : numpy 1D float array; Vertical gradient ((sensor1 – sensor2)/d_sensor) If only one sensor was used, grad = np.array([0.])
grad_data : bool; True if gradient data exist
xmin, xmax, ymin, ymax : floats; extreme coordinates of all data
dx, dy : step width in x and y direction. For Geometrics data, these are the step width of the first segment of the first line
topo : numpy 1D float array; actually not used, contains only zeros. Will be topography at every data point. Sensor height is thus topo+height
sensor1_ori : numpy 1D float array; deep copy of data, allowing to restart treatment with original data
sensor2_ori : numpy 1D float array; like sensor1_ori for data of sensor2
grad_ori : numpy 1D float array; like sensor1_ori for vertical gradient data
time : numpy 1D float array; time of measurement in seconds from beginning of month
gdata : class **Geometrics** object (see below for details)
inter_flag : bool; True if data have been interpolated onto a regular grid
config_file : str; Name of configuration file
file_name : str; Name of data file
file_type : str; file type, may be "GEOMETRICS", "MGWIN", "BRGM" or "GXF"
data_type : str; data type, may be "magnetic" or "gravity"
unit : str; if data_type is “magnetic: “nT” else: “mGal”

Other attributes defined when interpolating data onto a regular grid:
nx : int; number of columns
ny : int; number of rows
x_inter : numpy 1D float array of length nx; coordinates of the columns
y_inter : numpy 1D float array of length ny; coordinates of the rows
z_inter : numpy 2D array of shape (ny, nx); Interpolated z coordinates
t_inter : numpy 2D array of shape (ny, nx); Interpolated sample times
sensor1_inter : numpy 2D array of shape (ny, nx); gridded data of sensor1. Missing data: np.nan
sensor2_inter : numpy 2D array of shape (ny, nx); gridded data of sensor2. Missing data: np.nan
grad_inter : numpy 2D array of shape (ny, nx); gridded vertical gradient data. Missing data: np.nan
mask1 : numpy 2D array of shape (ny, nx); indicator where nans are located in the interpolated data of sensor1
mask2 : numpy 2D array of shape (ny, nx); indicator where nans are located in the interpolated data of sensor2
sensor1_fill : numpy 2D array of shape (ny, nx); array where data of sensor1 have been extrapolated to fill the entire grid (needed for certain transforms like pole-reduction)

sensor2_fill : numpy 2D array of shape (ny, nx); array where data of sensor2 have been extrapolated to fill the entire grid (needed for certain transforms like pole-reduction)
grad_fill : numpy 2D array of shape (ny, nx); array where data of vertical gradient have been extrapolated to fill the entire grid (needed for certain transforms like pole-reduction)

METHODS:

__init__
set_values : Sets certain attributes of DataContainer from outside
read_geometrics : Read Geometrics .stn or .dat file (G-858 instrument)
correctTime : Correct time of recording in case there was a timing problem with the instrument
prepare_gdata : Copies actual (usually modified or non-Geometrics) data into class Geometrics for creation of a *.stn or *.dat geometrics file
write_geometrics : Wrapper to write data in Geometrics MagMap2000 .stn format
write_dat : Wrapper to write data in Geometrics MagMap2000 .dat format
read_txt : Reads a non-Geometrics format magnetic data file
read_gxf : Read a GXF file (BRGM magnetic and gravity gridded files)
store_gxf : Store gridded data in GXF format
read_BRGM_flight : Reads magnetic data from flight lines out of a BRGM data file
get_line : Get data of one single line from dictionary self.segments
lines : Put all data into simplified dictionary self.data, one entry per line
read_base : Wrapper to read base station data
write_base : Wrapper to write base station data
Interpol : Interpolate data onto a regular grid
nanFill : Fill complete grid by extrapolation of data or return to grid with nans in corners

class Geometrics

All data sets, independent of data type and file format are copied into class Geometrics as object gdata. This is to allow exporting treated data into files with format Geometris *.stn or Geometrics *.dat (ASCII surfer format).

The class contains the following attributes:
sensor1 : numpy 1D float array; all data of sensor1 stored sequentially
sensor2 : numpy 1D float array; all data of sensor2 stored sequentially
x, y : numpy 1D float arrays; coordinates of all data points (no z coordinates)
day, month, year : numpy 1D int arrays; date of all data. If data file does not contain date information, the date of class creation is used
hour, minute : numpy 1D int arrays; hour and minute of all measurements
second : numpy 1D float array; second of measurement. If data file does not contain time information, the time of class creation is used. In this case, every line starts at a full minute and time step used is 0.1 s.
time : numpy 1D float array; time of measurement in seconds from beginning of month
line : numpy 1D int array; line number of every sample
mark : numpy 1D int array; segment number within line (useful only for geometrics data; for all other, array contains only zeros)

The following attributes exist only for Geometrics data:

infile : str; file name of data set

n_blocks : number of read blocks (files)

direction : int; line directions; 0: parallel to local Y axis, 1: parallel to local X direction

d_lines : float; average distance between lines

line_pos : numpy 1D float array; median value of line coordinates for each line (median Y coordinate for lines in X direction, median X coordinate for lines in Y direction)

mark samples : list of lists of int; position of marks in every line in number of sample taken just after the mark

segments : dict; This dictionary is actually only used for ease of data input. It is, however, maintained for future purposes (possibility to eventually correct mark positions). It has the following structure:

key: int; line number starting at 0; value: dictionary with the following items:

"direction" : float; direction of measurements with respect to local Y direction (may be 0, 180, 90 or -90)

"mark_samples" : list of int; contains for every mark (starting and end of line included) the number of the sample in self.sensor1 or 2 recorded just after the mark

"mask" : bool; if True, line is active (plotted), False if not. Initially, all are True

"median1" : float; median value of sensor 1 for the line

"median2" : float; median value of sensor 2 for the line; if sensor 2 does not exist or sensors are in horizontal disposition, "mdeian2"=0., in horizontal disposition, sensor2 is treated as sensor1 (i.e. its median value is stored in "median1").

"sensor" : int; may be 0 if both sensors are used in vertical configuration, 1 or 2 if line has been measured by sensor 1 or 2 respectively in horizontal configuration.

"dx" : float; for every segment, the average sampling step in X direction

"dy" : float; for every segment, the average sampling step in Y direction

"d" : float; for every segment the sampling step length

dx, dy and d contain one value less than mark_samples

"x" : float; median value of x-coordinates

"y" : float; median value of y-coordinates

"pos" : float; position of the line; depending on the line direction, "x" or "y" are supposed to be the position of the line

"block" : int; number of block (or file) if several files are read

"dir" : str; may be "odd" or "even". First measured line is "odd"; (odd or even refers to natural counting)

METHODS

__init__

fill_dict : Fill in values of dictionary self.segments for a full line

append_data : Appends date, time, coordinate information of one segment to arrays

read_stn : Reads a 2-sensor Geometix .stn file

read_dat : Reads a 1 or 2-sensor Geometix .dat file ("Surfer" format in MagMap)

write_stn : Writes Geometrics magnetic gradiometer file in stn format

write_dat : Writes Geometrics magnetic gradiometer file in *.dat (Surfer) format

clean_data : Set data to np.nan under certain conditions (check: probably not used)

line_days : Create a dictionary containing for every day of data acquisition the numbers of the lines having been measured

get_line : Get data of one single line

geometrics_lines : Put all data into a simplified dictionary, one entry per line

get_segment : Get data of one segment (data between two marks) of a line

read_base : Read data from a base station. It is supposed the base station is a Geometrics PPM station. The function recognizes automatically whether it is a G-856 (old version) or a G-857 (newer version) instrument

write_base : Write base station data in G-856 format

APPENDIX 2: Inversion process

The inversion aims at iteratively minimizing a cost function which is defined as:

$$\sum_{Nd}\left(\frac{\sum_{Np}\left(\frac{\partial d_i}{\partial p_j}p_j\right)-d_i}{\sigma_i}\right)^2+\mathrm{lam}\sum_{Np}\left(\frac{p_j-p_{j0}}{\sigma_j}\right)^2+\gamma(Sp)^TSp \qquad \text{eq.1}$$

where Nd is the number of data points, Np the number of model parameters, d_i the data values, p_j the parameter values p_{j0} the parameter values of the former iteration, σ_i the uncertainty of every data point, σ_j the variability of every model parameter, S the smoothing matrix, p the vector of model parameters, lam a factor to weight regularization with respect to data fit and γ a factor to weight smoothing with respect to data fit. The first sum corresponds thus to data fit, the second one to a limitation of parameter variance from one iteration to the next (its effect is also to keep the overall parameter variance small) and the third term corresponds to smoothing, i.e. the difference between the properties of neighbouring blocks should be minimized. The matrix S has size (Np x Np). On its diagonal is the number of neigbouring prisms for every prism and -1 at the position of every neighbouring prism in the line and column corresponding the prism.

The data will be assembled in vector d, the parameters in vector p and the derivatives of data with respect to parameters in matrix G which has a size of (Nd x Np) and G_{ij} corresponds to the derivative of data point i with respect to parameter j. Every column of matrix G is stored as attribute of the corresponding prism class object.

Now we have to distinguish between the case of imposing positivity constraints or not.

1) **No positivity constraint is imposed**
   Since in every iteration, the position of the prisms is imposed, the forward problem (G∗p) is linear and minimizing equation 1 results in the following equation:

   $$p = \left(G^TC_d^{-1}G + lam\cdot I_p * \gamma S\right)G^TC_d^{-1}d \qquad \text{eq. 2}$$

2) **Positivity constraint is imposed**
   In this case, the values to be inverted for are the logarithm of the prism properties. This modifies equation 2 and the problem becomes non-linear. If we define y = ln(p), the linearized forward problem becomes:

   $$d = d_0 + G\cdot exp(y_0)\cdot(y-y_0)+\varepsilon \qquad \text{eq. 3}$$

   Where y_0 is the vector of the logarithm of the actual parameter values, d_0 the theoretical data values calculated from the model parameters $p_0 = \exp(y_0)$. ε is the data misfit.
   So, G of equation 2 becomes G∗p where the multiplication in this case means that every column of G is multiplied by the actual value of the corresponding prism property (or by exp(y)).

   With this modification and replacing p by y, equation 2 is used also in the case of positivity constraint. However, every iteration is now non-linear and the optimum solution of one prism distribution is found itself in an iterative process. This process is stopped after a maximum of 50 iterations or if the relative misfit reduction $(\mathrm{std}(\varepsilon_0)-\mathrm{std}(\varepsilon))/\mathrm{std}(\varepsilon_0)$ becomes smaller than 10^{-5}.

After every iteration, the program searches first the data points where the misfit has a local maximum (all local misfit maxima larger than a threshold are determined). Then, for every one of these data points, matrix G is scanned for the parameter that has the maximum absolute effect on

this data point (where matrix G has the largest absolute value in the row corresponding to the data point). The prism corresponding to this parameter is then split into two prims of equal size in all dimensions (i.e. the prism is split into 8 smaller prisms for 3D inversion or into 4 smaller prisms in 2.5D inversion where the dimension perpendicular to the line is fixed). The derivatives for prism that are not modified are already stored from the former iteration and only the derivatives (columns of matrix G) for the new (reduced) prisms must be calculated. The columns of the original prism are eliminated and the 4 or 8 columns of the smaller prisms are added to G at the end. If in one direction subsequent prism divisions have led to the user defined minimum size, in this direction, the subdivision is not done again, but in the other direction, it may still be done. If no more subdivision for any one of the prims controlling the local misfit maxima is possible, the program stops.

If iterations may continue, the next iteration is then done using the new matrix G.