An introduction

to

Version: 0.9.3

Inhaltsverzeichnis

1. Introduction....3
2. Installation....4
2.1 Linux (case: Ubuntu18.04 desktop)....4
2.2 MacOs (case: Mojave)....4
2.3 Window (case: Windows10)....5
2.4 Installation within Python2.7 environments....5
2.5 Updating MagPy....5
3. First steps....6
3.1 Graphical user interface of XMagPy....6
3.2 Loading data from various sources....6
3.2.1 Loading single or multiple files....7
3.2.2 Accessing web services and other remote data sources....8
3.2.3 Database access....9
3.3 Exporting data and Plots....9
3.3.1 Exporting and saving data....9
3.3.2 Plots....10
3.4 Platform specific aspects....10
3.4.1 MacOS....11
3.4.2 Windows 10....12
4. Panels....13
4.1 Data panel....13
4.1.1 Select columns / Drop columns....14
4.1.2 Extract values....14
4.1.3 Plot options....14
4.2 Flagging panel....15
4.2.1 Flag outlier....16
4.2.2 Flag selection....17
4.2.3 Flag range....17
4.3 Meta panel....18
4.4 Analysis panel....20
4.4.1 Derivative....20
4.4.2 Rotation....20
4.4.3 Fit....21
4.4.4 Maxima, Minima, Mean....22
4.4.5 Flags....22
4.4.6 Filter....23
4.4.7 Offset....24
4.4.8 Smooth....24
4.4.9 Resample....24
4.4.10 Delta F....25
4.4.11 Activity....25
4.4.12 Baseline....25
4.4.13 Calculate F....26
4.4.14 Frequency methods (under development)....26
4.4.15 Statistics....26
4.5 DI Panel....26
4.6 Report panel....28

4.7 Monitor panel....28
5. Main menu....30
5.1 Memory and multiple data operations....30
5.2 DI data input....34
5.3 Database connections....36
5.4 Options and Help menu....36
5.5 Specials and extra....37
6. Recipes....38
6.1 Daily data quality check and flagging....38
6.2 Basevalue analysis and adopting baselines....44
6.3 Definitive data check....49
6.4 The art of web services....53
7. FAQ....56
8. Appendix....58
8.1 Example files....58
8.2 MagPy’s DI fluxgate data format....58
8.3 Command Line Tools....59
8.3.1 mpconvert....59
8.3.1 mptest....59

1. Introduction

The analysis of geomagnetic data is a primary responsibility of approximately two hundred observatories around the globe. All procedures, from data quality control to final data submission, increasingly require more sophisticated and effective tools, especially when it comes to high resolution records. MagPy, short for Geomagnetic Python, is a platform-independent, multi-purpose software package to assist with geomagnetic data acquisition, storage, distribution and practically all analysis procedures used in geomagnetic observatories. It supports all common data formats used in the geomagnetic community, which include both instrument-specific and general purpose types such as IAGA02, WDC, IMF, IAF, and ImagCDF. Direct access to data stored in online sources using URLs is also supported, allowing the user to obtain data directly from the dedicated World Data Centers. Routines for common analysis procedures covering quality control, filtering, data series merging, baseline adoption, format conversion, and data submission are provided. Data handling functions from basic import, treatment, and export up to complex automated real-time analysis of geomagnetic data can also be easily implemented. MagPy can communicate with a MySQL database, which further facilitates effective archiving of data and all necessary meta information. MagPy is an open source project hosted on GitHub (https://github.com/geomagpy/magpy) and is available on PyPi (https://pypi.python.org/pypi/geomagpy).

The graphical user interface XMagPy provides easy access to most routines offered by the underlying python package. The basic package will hereinafter be referred to as back-end. A manual for back-end methods and functionality can be found on MagPy‘s GitHub page (see link above). In the following we will focus solely on the front-end, the graphical user interface. For better readability we are using a number of conventions. Selection paths to menu items, submenus, panels and buttons are contained in gray text boxes with arrows marking the sequence. Code fragments are contained in yellow text boxes. Button and panel names are usually written in bold face.

Example files are part of the MagPy package. They can also be downloaded directly from the github directory given above in folder magpy/examples/.

2. Installation

In the following common installation procedures for each platform are described. Extended descriptions for specific platform are given in the appendix. Although MagPy1.0 fully supports Python 2.7 and 3.x environments is is highly recommended to use Python >=3.6 as basis as any future development will be solely focused on Python3 compatibility. For all platforms you will need such a python environment. By default this is present on Linux wherefore installation is pretty straightforward using pythons setuptools. All dependencies are resolved automatically.

2.1 Linux (case: Ubuntu18.04 desktop)

open a terminal (ctrl-alt-t) and enter:

```
$ sudo pip3 install geomagpy
...
$ sudo pip3 install wxpython
```

Now you can start XMagPy from the terminal using:

```
$ xmagpy
```

If you don‘t have root access, you can install a python environment within your home directory (e.g. anaconda3, miniconda3) and then install magpy within this environment. Please check the FAQ for installation issues.

2.2 MacOs (case: Mojave)

On MacOS install a python3 interpreter. We recommend Miniconda3 or Anaconda3 (https://www.anaconda.com/distribution/)
Please follow the installation instructions there.

If successfully installed, open a terminal and enter:

```
$ pip install geomagpy
...
$ pip install wxpython
```

Thats it!
You can start XMagPy now from the terminal using:

```
$ xmagpyw
```

In order to create a desktop link got to the Finder, search for xmagpyw and copy it to the desktop.

2.3 Window (case: Windows10)

It is recommended to use the windows installer here:

https://cobs.zamg.ac.at/data/index.php/en/downloads/category/1-magnetism

The windows installer will install a python3 environment including magpy and xmagpy. It further will create links to xmagpy, a python shell and command shell within the applications menu.

If you you have any earlier version of MagPy, please uninstall to avoid any conflicts. If you don‘t have administration privileges you can install a python environment with your home directory (e.g. winpython) and then install magpy within this environment, using the same commands as shown above in 2.1 and 2.2 (see appendix).

2.4 Installation within Python2.7 environments

Please note that running MagPy with full support of CDF file formats requires appropriate libraries. Such libraries are contained within the Python3 installation. For Python 2, however, you need to install such libraries separately, namely NasaCDF tools and spacepy. Please follow the instructions given on their webpage. Another prerequisite is the outdated wxpython 3.x. You will find more details and some guidelines also in the appendix.

2.5 Updating MagPy

Once installed you can easily update MagPy to the newest version with a simple command.
On Linux and MacOs open a terminal, on Windows, open the Command shell from the MagPy menu, and enter:

```
$ pip install -U geomagpy
```

If you are interested in issues, upgrade news and further information, please visit the GitHub folder and/or subscribe to the Telegram news channel ...

3. First steps

3.1 Graphical user interface of XMagPy

The graphical user interface of MagPy is based on WXPython (https://wxpython.org/).

Figure 3.1: The graphical user interface of XMagPy consists of 4 standard sections and 1 optional part.

The main program window is separated in several parts as shown in Figure 3. Figure 3.1a contains the main menu for data access and memory/multiple data operations. Within the main menu you will find drop down lists for file operations, database issues, DI analysis, memory operations, some extra stuff, options and the help menu. File operations will be discussed in the next section (3.2). Details on all other menus will be summarized in chapter 5.

The panels (Figure 3.1b) provide access to individual methods for data analysis, interpretation and manipulation of active data sets. An individual section is devoted to each panel in chapter 4.

Timeseries plots are displayed within the plotting area (Figure 3.1c). Status information (Figure 3.1d) and current working states are found at the bottom of the window. An optional statistics view (Figure 3.1e) can be activated within the Analysis Panel (see section 4.4.15).

3.2 Loading data from various sources

MagPy supports a large variety of different data sources and formats, which are typically used within the geomagnetc community. When looking at the file menu, you see direct support for the following sources: file(s), webservices, remote data sources and database. Databases such as MySQL and MariaDB are also supported. In order to access existing data in one of the supported formats use:

MainMenu → File

This will open the menus content within a drop down list.

3.2.1 Loading single or multiple files

Most users will access data files on a local or connected file system. To do this use

MainMenu → File → Open File...

This will open a standard file selection menu for your operating system. Locate the file you want to open. No further selections are necessary. If the format type is supported, MagPy will identify it and open the data. Multiple selections are possible.
Currently supported file formats can be found here

MainMenu → Help → Read formats / Write formats...

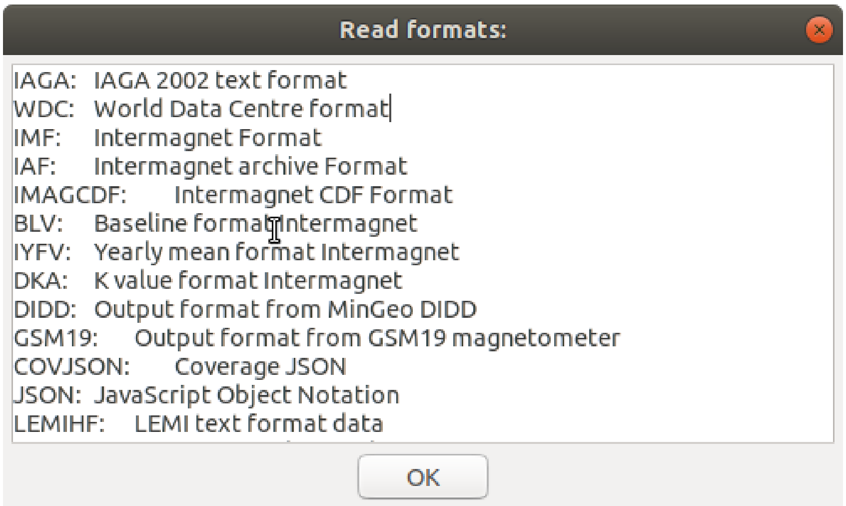


Figure 3.2.0: supported formats

You can also define a directory and then choose the time range you are interested using the following option:

MainMenu → File → Select Directory...

This option is desigend for archive directories with huge number of individual files were you don‘t won‘t to scroll through the long list of data. This method only works, if the dates are contained in the file names following the recommendations of the respective file format.

3.2.2 Accessing web services and other remote data sources

MainMenu → File → Open Webservice...

gives you access to machine readable webservices of institutions and organizations. Currently (01/2020) webservices of the USGS and the ZAMG are supported. Please check access and data licenses before using it. The webservice selection window is displayed in figure 3.2.1

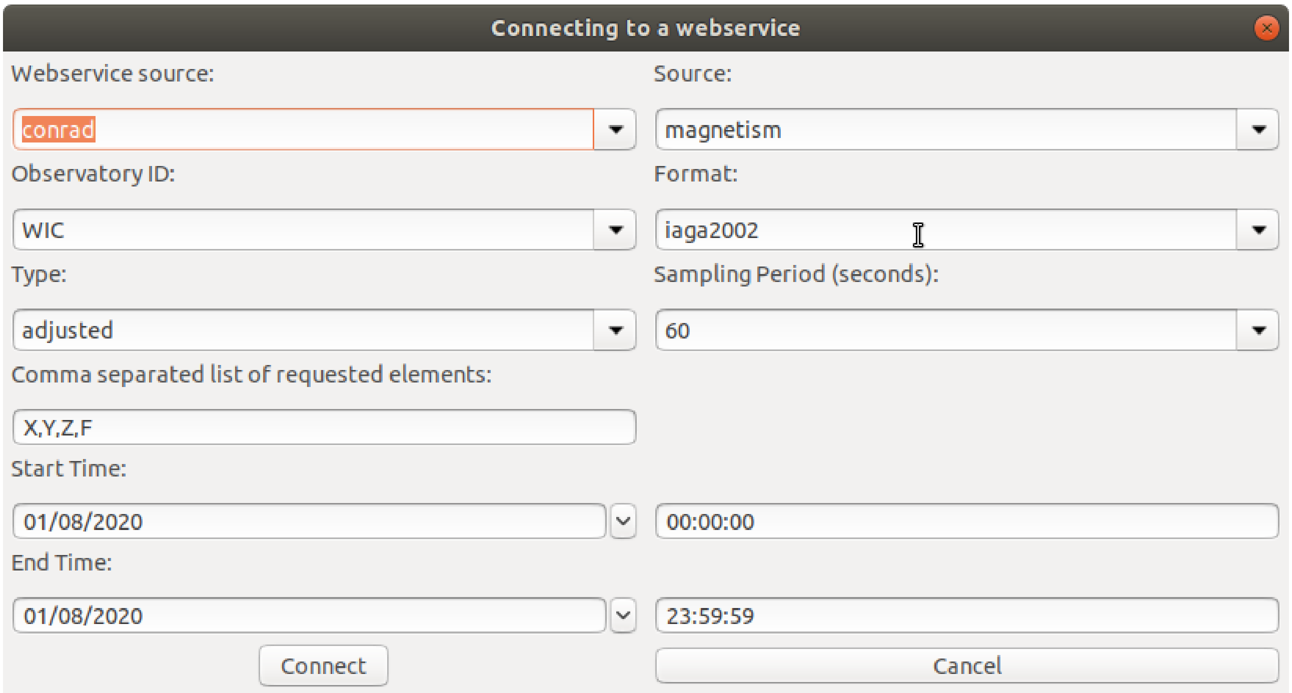


Figure 3.2.1: Webservice selection window. The data request url will be constructed from the the given parameters.

MainMenu → File → Open general URL...

allows you to access remote data provided on html and ftp servers. Some examples are contained (figure 3.2.2).

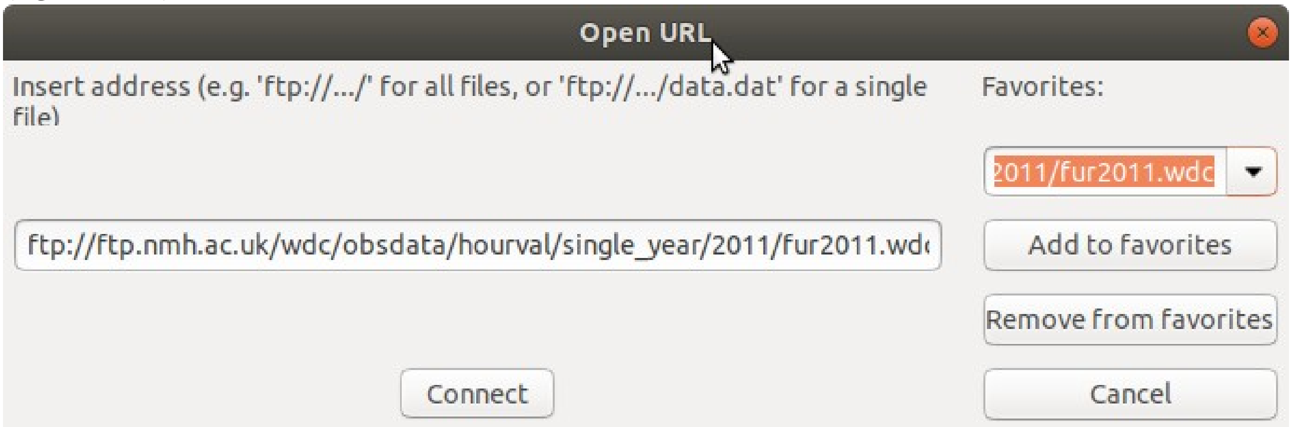


Figure 3.2.2: Input window for remote connections. The example accesses hourly wdc data from the world data center in Edinburgh via FTP. You can establish/modify also a favorite list here.

3.2.3 Database access

MainMenu → File → Select DB table...

finally is used to access a connected MySQL/MariaDB database. This database needs to be initialized using MagPy. Please check section 5.3 for details. A drop down combo box will give you access to data tables contained within the database. When opening you will further be asked to define a time range.

Figure 3.2.3: Database table access.

3.3 Exporting data and Plots

3.3.1 Exporting and saving data

You can save data by using the export method of the file menu. When choosing

MainMenu → File → Export...

a selection window as shown in Figure 3.3.1 is showing up.

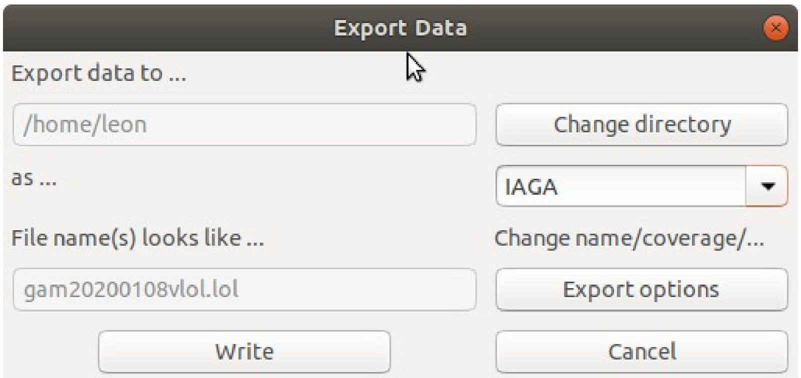


Figure 3.3.1: Export data window.

Within the export data window you can select the destination path. In dependency of the selected format, file name suggestions are constructed. You can change that, however, anytime within the Export options. Please note, that some file formats require specific meta information. Please check section 4.3 for more detailed information.
Simple format conversions can be done easily here: Just load for example your IAGA-2002 one second data for one month and then export it to ImagCDF. For correct outputs you have to add ImagCDF specific meta information before exporting as shown in 4.3. As a plot will be generated

when opening one month of one second data, this will need some time. If you have a large amount of data to be converted it is advisable to use the MagPy back-end‘s read/write methods or the MagPy command line application mpconvert (see appendix 8.3.
Please note that not all export formats make sense for all data types. Here it is assumed that the user knows about benefits and drawbacks of specific well known geomagnetic data formats. Details on these geomagnetic data formats and their contents as well as meta information requirements can be found here: https://www.intermagnet.org/data-donnee/formatdata-eng.php
There are also three MagPy specific output formats which can be used for archiving and storage. PYCDF is based on NasaCDF and can hold basically any meta information you define. It is specifically desigend to save flagging information and adopted baselines along with raw data, to allow for a single file storage (and thus reconstruction) of all analysis steps typically used in geomagnetic data analysis.
PYSTR is a comma separated ascii text file which supports all basic meta information.
PYASCII is a semi-colon separated ASCIItext file which focuses on data.

3.3.2 Plots

In the lower left section of the plotting area you will find the plotting tool box as shown in figure 3.3.2.

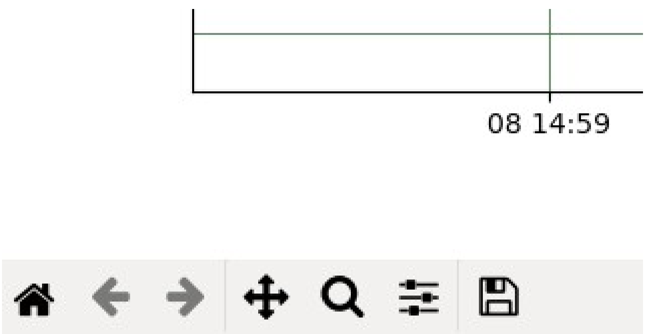


Figure 3.3.2: The plotting tool box.

The seven symbols shown here can be used as follows. The **home** button is used to go back to the full data plot. **Left** and **right** arrows allow you to go one step backward or forward, respectively. When clicking on the **move** symbol with arrows in four directions you can move individual graphs within their plotting area. The **zoom** button is also very important for data selection and flagging. Several methods define their validity range within the zoomed area of the plot. The **configure** button opens a configuration window, which allows to modify the size and position of the plot within the plotting area. Plots can be saved with the **save** option. Default output format is png. You can select however other formats as well of which encapsulated postscript (eps) and scaleable vector graphics (svg) are the most important vector formats. Png, pdf, and jpeg are the most important non-vector types.

Modifications of plot style, color, symbols, padding, etc can be performed within the plot options menu to be found on the data panel (see section 4.1).

3.4 Platform specific aspects

MagPy has been dominantly developed and tested on Linux Systems. Version 1.0 is extensively tested on Ubuntu 18.04 and other Debian like systems. Nevertheless, the software is platform independent and can be used on other systems as well (see chapter 2 for installation instructions).

Although python itself is platform independent, the realization of a graphical user interface with full functionality on all platforms is not straightforward. XMagPy looks slightly different on other systems. In order to give you an impression you will find two examples for MacOS and Windows10 below. All other examples in this manual are taken from an Ubuntu Linux machine.

3.4.1 MacOS

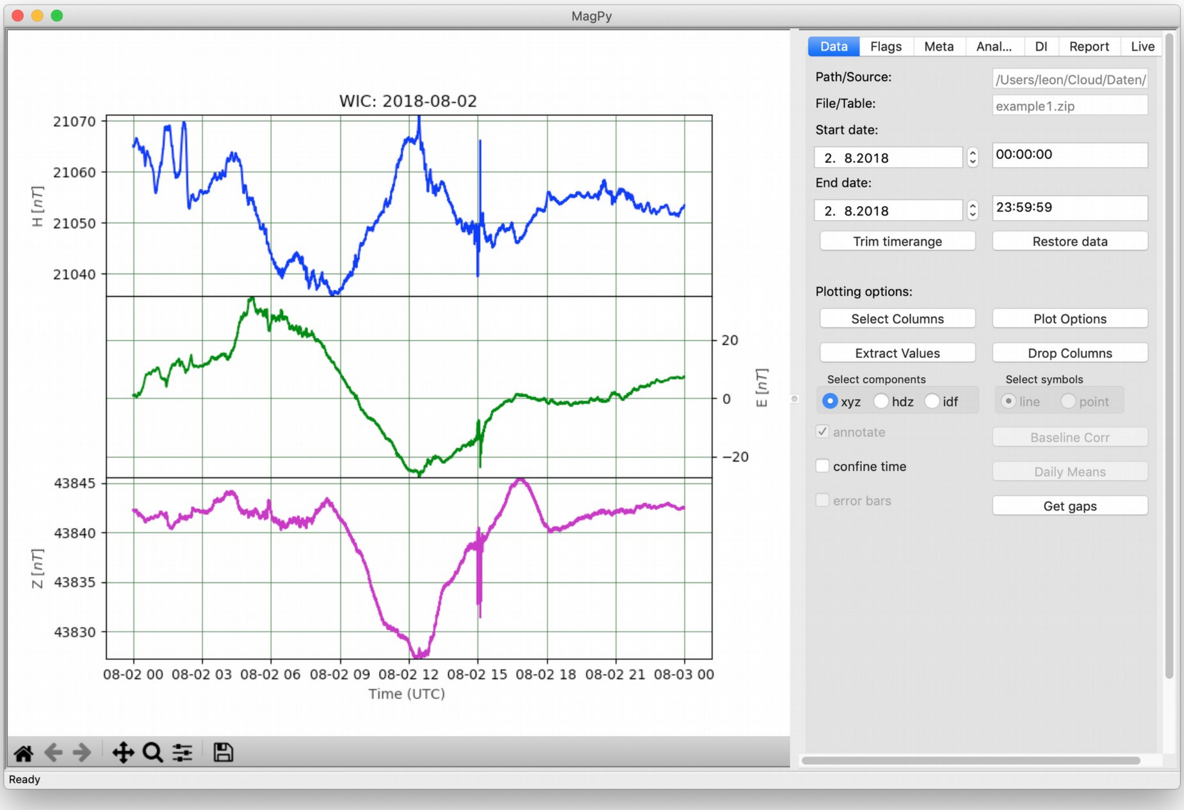


Figure 3.4.1: MagPy on a MacOS Sierra. For optimal user experience it is recommended to use a light background on MacOS. The main menu is found on the top of the screen.

3.4.2 Windows 10

On Windows 10 machines MagPy will look similar to the example as shown in Figure 3.4.2.

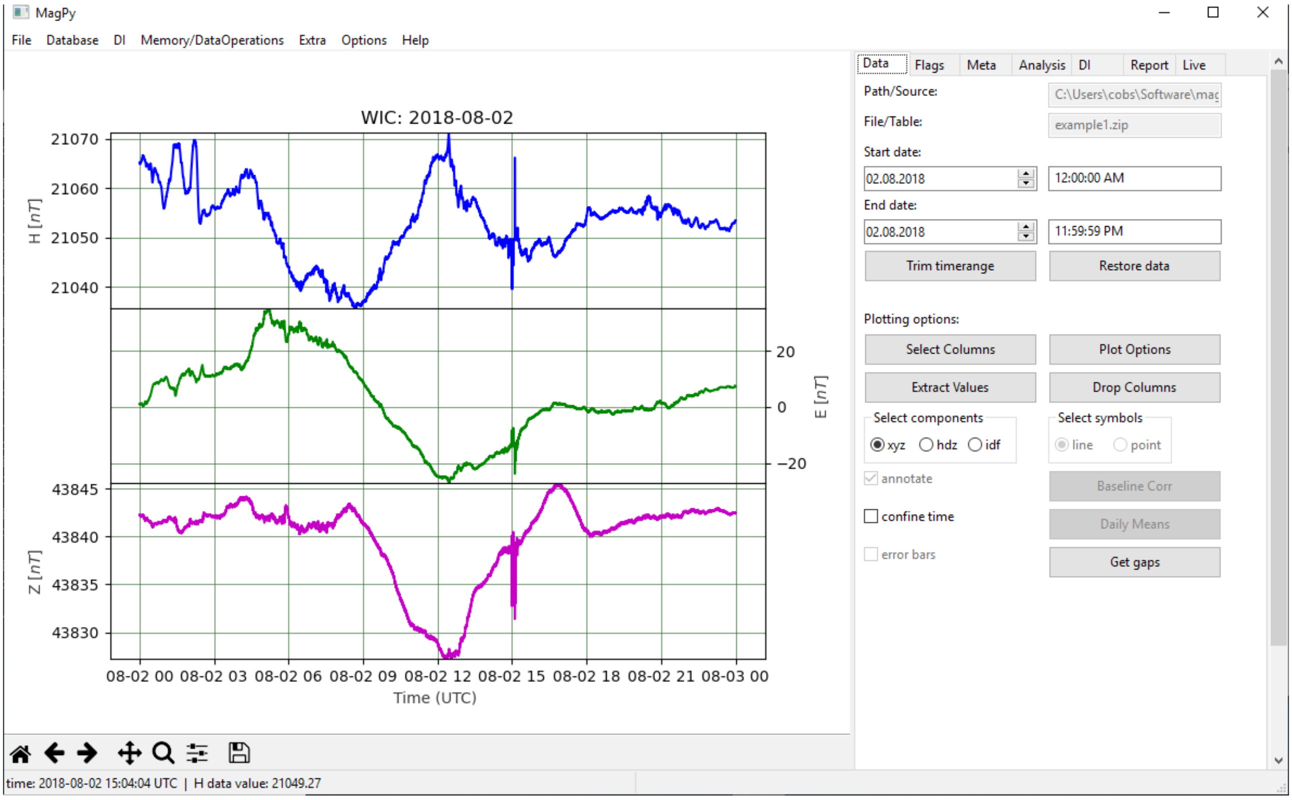


Figure 3.4.2: MagPy on Windows10.

4. Panels

Within the panels you will find methods and functions for data analysis, evaluation and manipulation. All these methods aim on individually selected timeseries or data stream. Methods for multiple stream operations or methods not related to time series analysis are found in the main menu.

4.1 Data panel

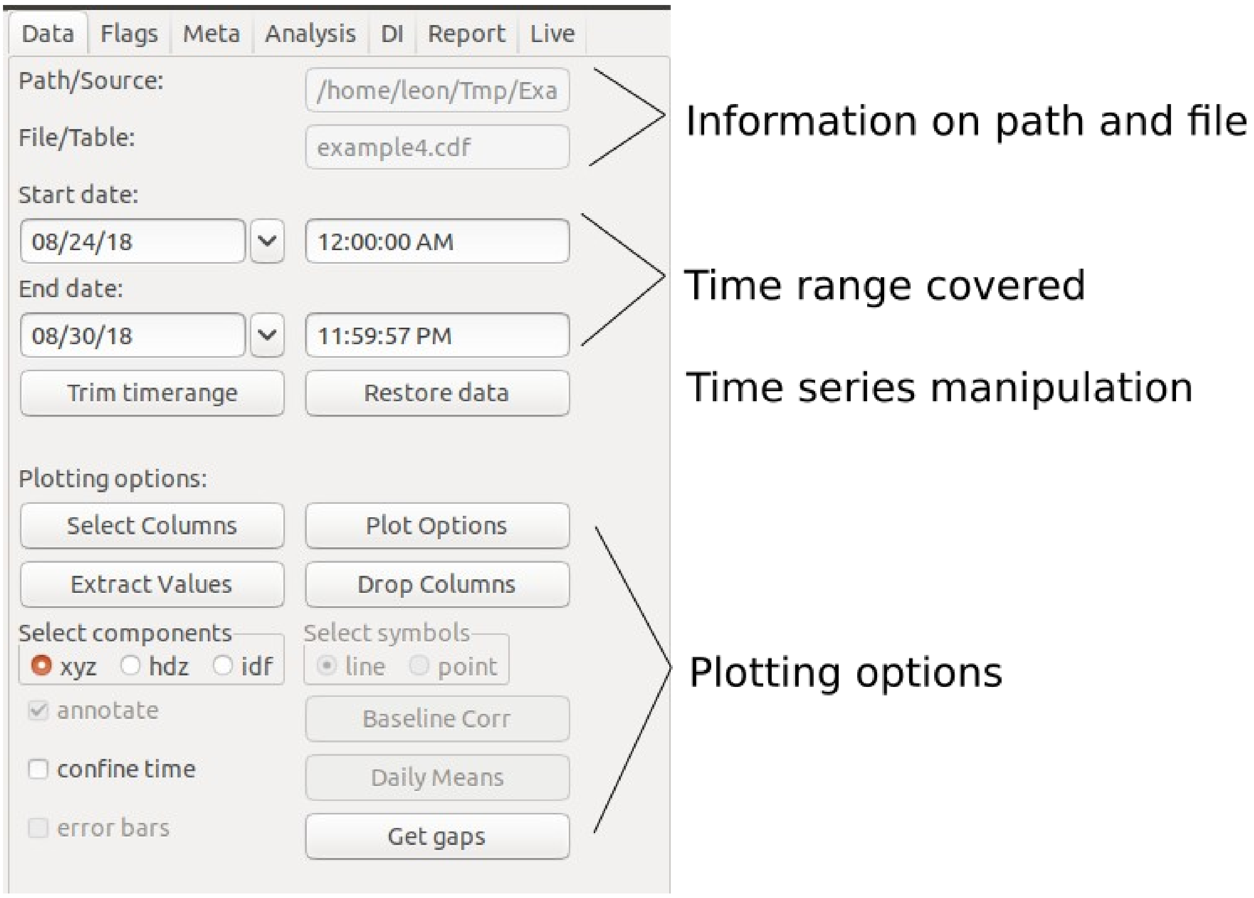


Figure 4.1.1: Data panel.

The data panel contains general methods for data manipulation and display. The source of the opened time series and its covered time range is given here. You can modify start and end time by changing times or dates subsequently using the **trim** button. The restore button is used to reestablish the originally loaded time series by undoing all analysis operations. For some operations restore is not working, you have to reload the original data source.
The plotting options allow to select shown components, extract values larger or smaller than a given threshold or even to remove individual components from the series. In case of vector data you can switch between Cartesian and other coordinate systems. **Annotate** will show flagging information, **confine time** will remove the flexible dates on the x-axis and replace them by fixed scales, which are sometimes better for printouts, **error bars** can be shown if contained in the time series. The three buttons on the lower right comprise the following methods: **Baseline corr**ection will be enabled if adopted baseline information is available for the active time series (see section 6.2 for details). **Daily mean** including uncertainty ranges can be shown for basevalue timeseries with

multiple measurements at specific days. **Get gaps** checks the time series for periodicity and eventually identifies gaps by identifying missing time steps. Corresponding values are filled by NaN. An additional column will be created with binary information (0: data available, 1: gap). Such data will result in a discontinuous plot.

4.1.1 Select columns / Drop columns

The **select columns** button opens a dialog window to select the components shown in the plot. When exporting, all data will be saved, also data currently not shown in the plot. In order to permanently remove data columns from the time series you need to use **drop columns**.

4.1.2 Extract values

The **extract value** button will open a dialog window in which you can choose up to three selection criteria to extract values from the time series as shown in Figure 4.1.2.

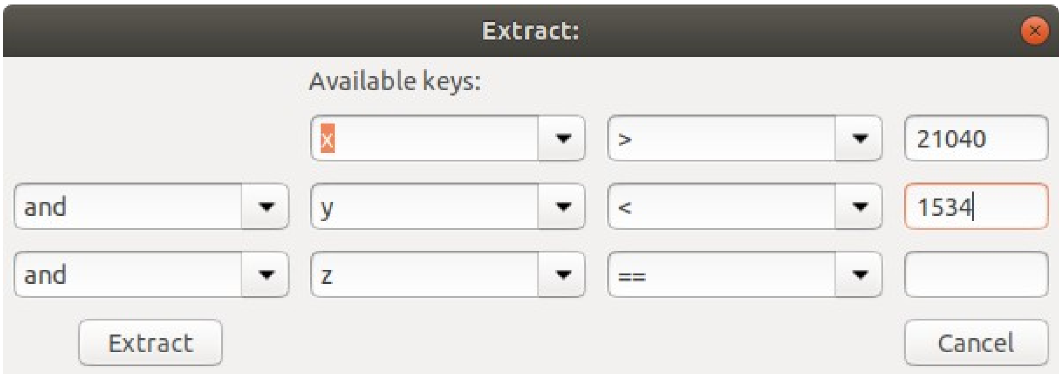


Figure 4.1.2: Dialog for extraction of values by defining thresholds.

4.1.3 Plot options

The **plot options** button gives you access to a variety of options to modify the layout of the data plots shown in the plot panel. The options will be updated soon in an upcoming version of MagPy.
Often used options are
1) padding, to add space towards the amplitude in each plot,
2) colorlist to define the colors of each shown graph (b = blue, g=green, k = black)
3) plottitle
4) flagontop: use False if the colored flagging info should be below the data points

Please refer to the back-end manual for more option descriptions or just try it.

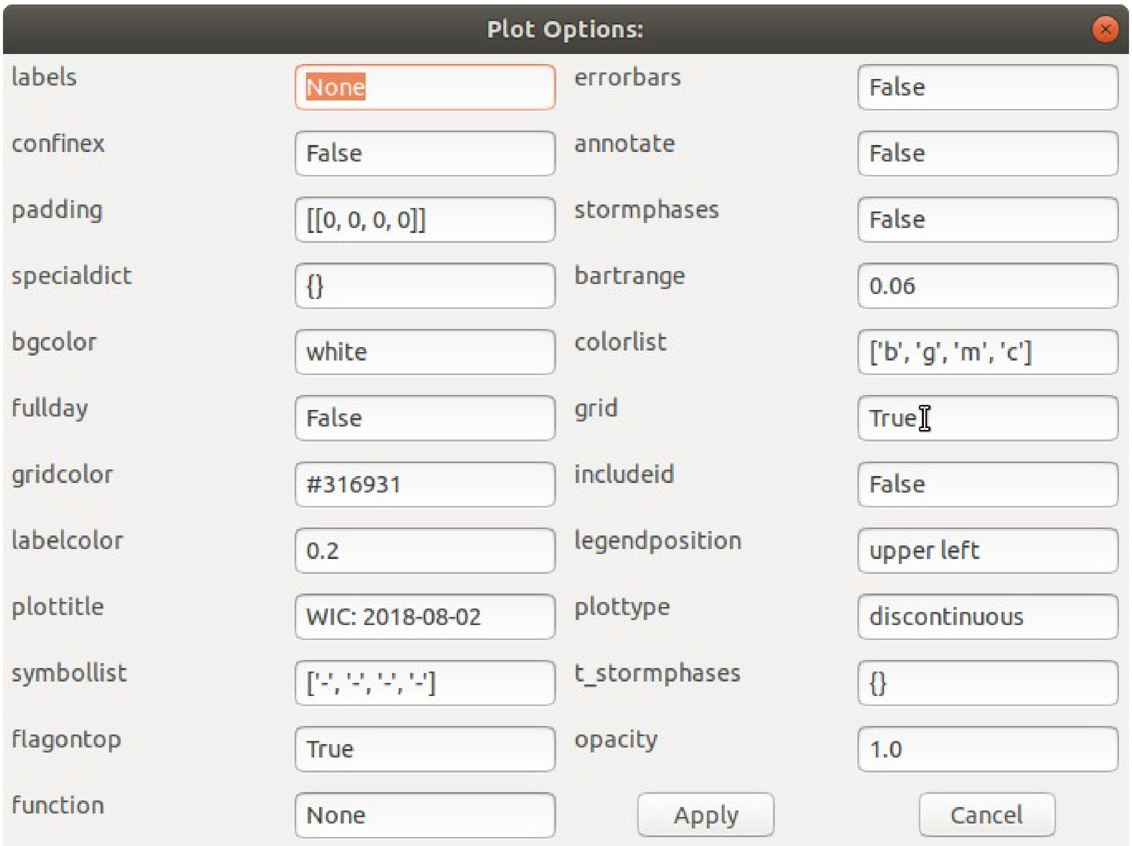


Figure 4.1.3: Selecting plotting options. Please note that this options dialog will be simplified in a future version of MagPy.

4.2 Flagging panel

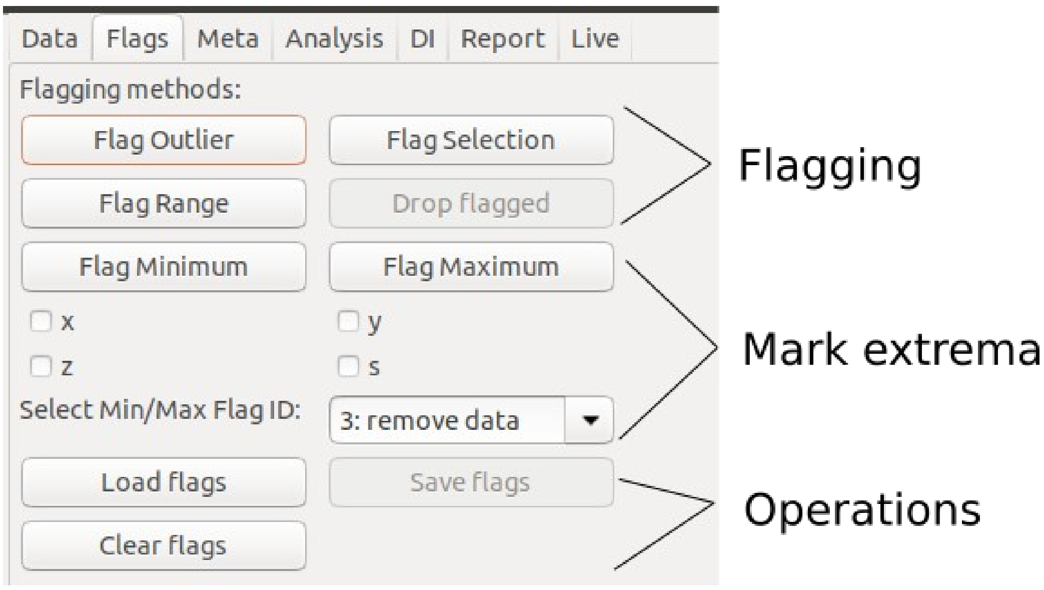


Figure 4.1.1: Flagging panel

The flagging panel (short: Flags) contains methods for flagging data, as well as possibilities to store and load such information into timeseries. Please note that flags are always connected to a specific SensorID which is related to and obtained from the data source. The main flagging functions are accessible by buttons: **Flag outlier** will scan the selected time series, which means the active shown region within the selected graph, for outliers (4.2.2). **Flag selection** will marks all currently selected data. To use this method you zoom into a specific plot area (see 3.3) and then press the Flag selection button. This is the most common method for regular data flagging. **Flag range** allows for defining either a value or a time range to be flagged. If you just want to flag maxima or minima within the time series, you firstly select the component(s) and flag ID, and then use the **Flag maximum** or **Flag minimum** button. Flags can either be saved within a connected data base (which I would recommend) or into a data file. The flag file supports two formats, pickle, a binary structure, or json, an ASCII structure.
Drop flagged will remove flagged data with ID 3 (remove data) and ID 1 (automatically detected outlier) and replacing them by NaN.
Clear flags deletes all current flagging information, keeping all data unchanged.

4.2.1 Flag outlier

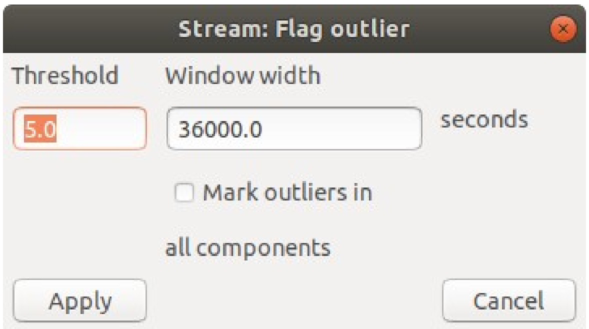


Figure 4.2.2: Flag outlier selection menu

The Flag outlier method makes use of interquartile ranges (IQR) to identify how spread-out values are within a certain **time window**, moving along the time series. Data outside the IQR range multiplied by a certain multiplier, also referred to as **threshold** value, are identified as outliers. In statistics usually a multiplier of 1.5 is assumed. I recommend larger values here. You can change time range and multipliers in a selection window when applying this method (Figure 4.2.2). Multipliers of 3 and window lengths of 30 to 60 seconds are typically good to identify lightning strikes in one-second data. Multipliers of 5 and window lengths > 300 seconds are good enough to identify spikes and keep most natural signals in. By default a time window of 600 times the sampling rate is assumed. If **marking outliers in all components** is chosen then every outlier detected in a single component is also marked in other components as well. This is particularly useful for vectorial data.
Outlier flags are „automatic flags“ and will be displayed in yellow within the plot. The flag comment (not shown in the plot) will contain threshold (muliplier) and time window length.

4.2.2 Flag selection

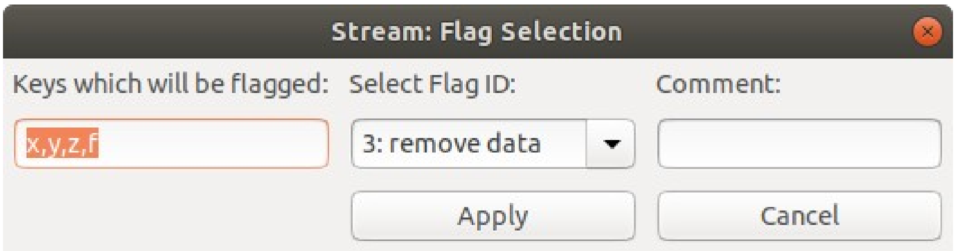


Figure 4.2.3: Flag selection menu.

In the Flag selection menu you can define the components (keys) to which this flag has to be applied. You can further select a general flag type, characterized by an ID. This type can be either

ID 0, normal data: just add a comment to the selected data
ID 1, automatic flag: should only be used by automatic processes like outlier detection
ID 2, keep data: data to be kept for definitive data production, ID 2 data cannot be be overwritten by automatic processes
ID 3, remove data: don‘t use this data for definitive data production
ID 4, special: used for example by merging data from a different source into the time series

The provided comment is free-form text, but don‘t use special characters here. If several observers check data, then it is good practice to start or end each comment with a observer identification, and to agree on certain keywords describing repeated disturbances. This way, flagging information can later be scanned again for specific disturbances. A possible input would be:
lightning RL
with RL being the observer’s identification

4.2.3 Flag range

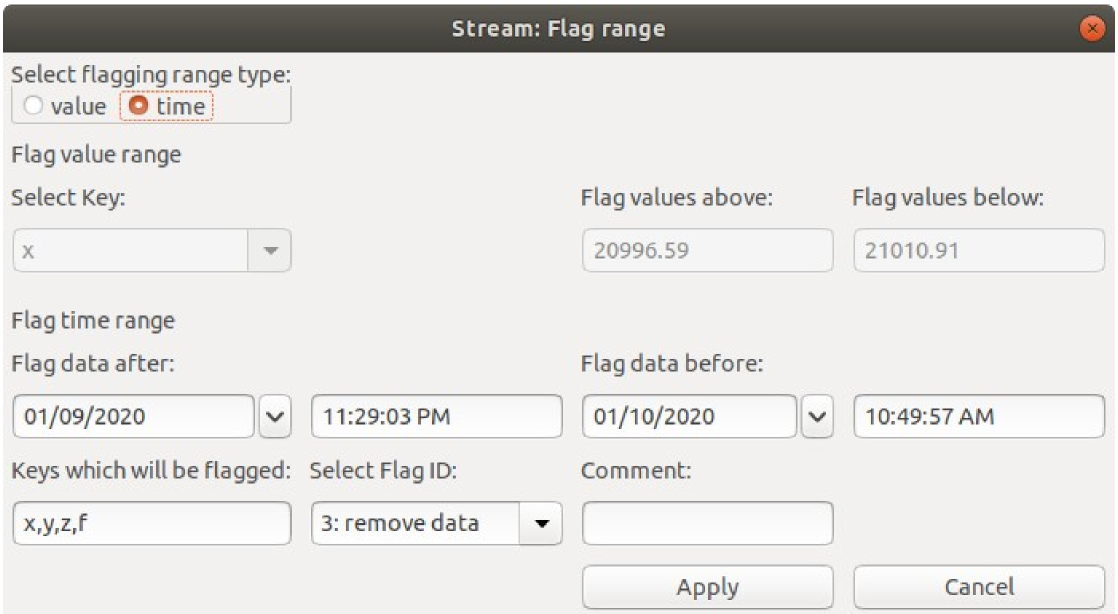


Figure 4.2.4: Flag range menu allows you to flag data between given boundaries either in time or amplitude.

The Flag range method allows you to select specific boundary values. Data excceding these boundaries are flagged. Initially you have to choose between amplitude (**value**) or **time** range. When selecting value, you provide the component/key and threshold values. If a time range is to be flagged, as shown in figure 4.2.4, you will have to provide lower and upper date and time. If you zoomed into a graph, these values are prefilled by the boundaries of the zooming window.
For both selections you have to provide comments, ID, and components/keys to be flagged as described in section 4.2.3.

4.3 Meta panel

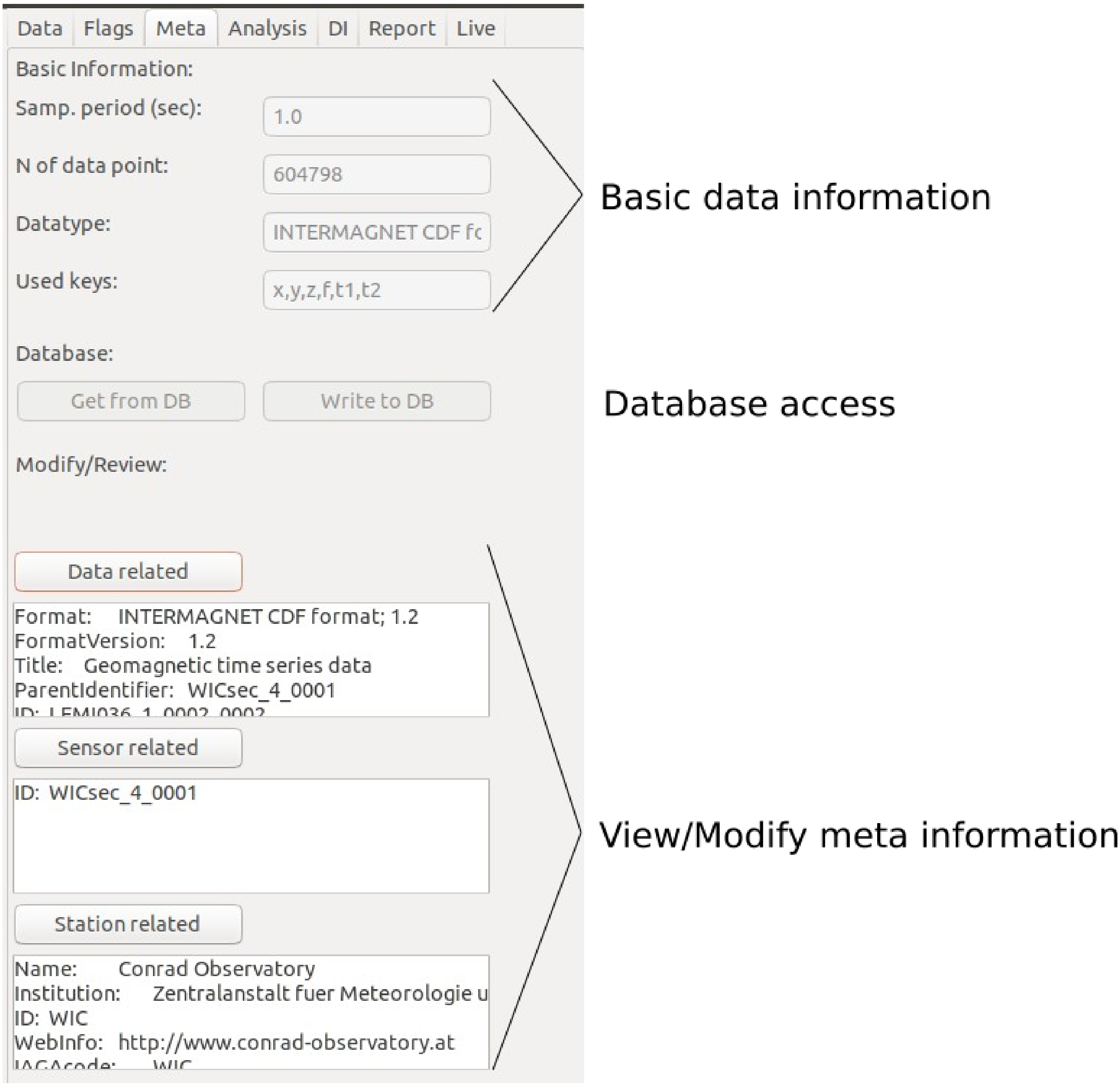


Figure 4.3.1: The meta data panel.

The meta data panel gives you an overview about all meta information connected to the currently active data set. This meta information is usually obtained from the data file. If a database is connected, and you have information for DataID, SensorID and/or StationID within your database, you can extend that information with database contents (**Get from DB**). This is useful for exporting data e.g. when converting IAGA-2002 to ImagCDF.

You can always modify or extend meta information when clicking on the buttons for **data related**, **sensor related** or **station related** meta information.

As an example the input sheet for station related information is shown in figure 4.3.2. Here you will see several field which can be filled with your information. The field names are self explaining and will be automatically converted to obligatory field names when exporting specific formats like ImagCDF. Numbers in parenthesis behind some field names indicate that this information has to be provided for the corresponding output (1 → Intermagnet Archive Format IAF, 2 → IAGA 2002 format, 3→ ImagCDF).

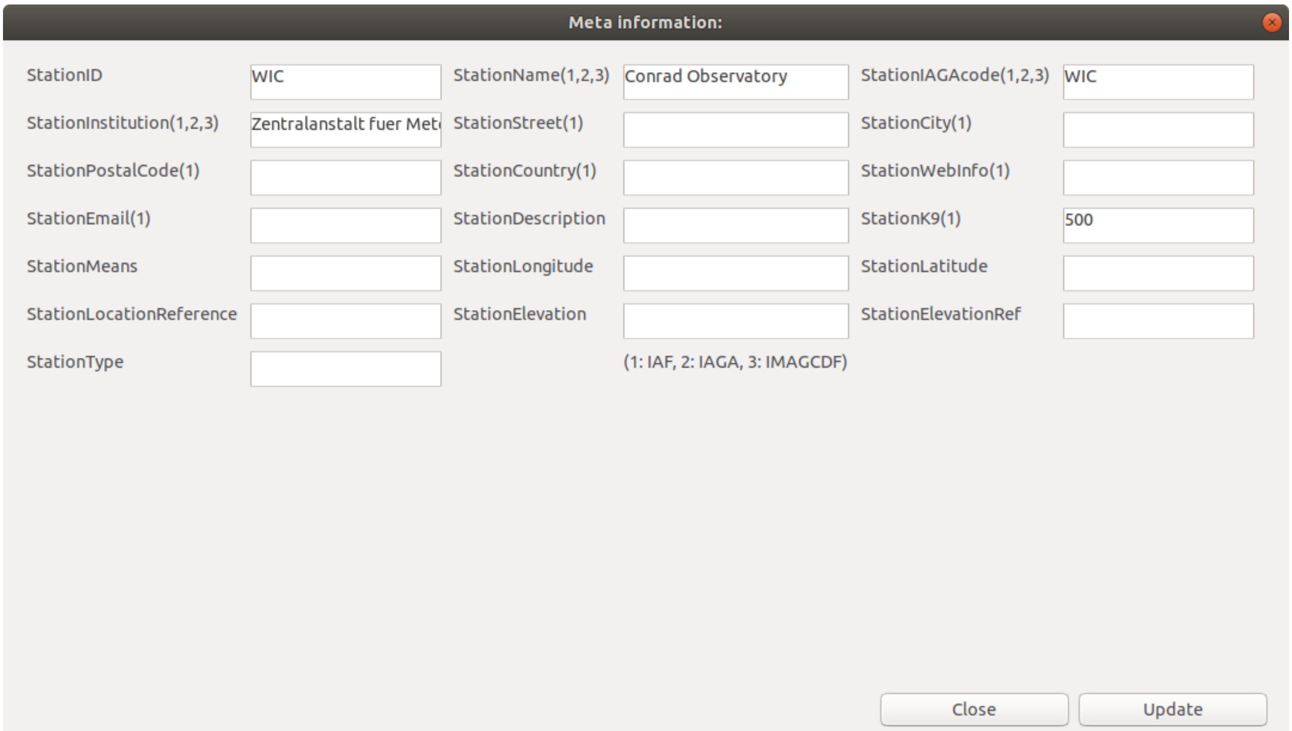


Figure 4.3.2: Input sheet for station related information.

4.4 Analysis panel

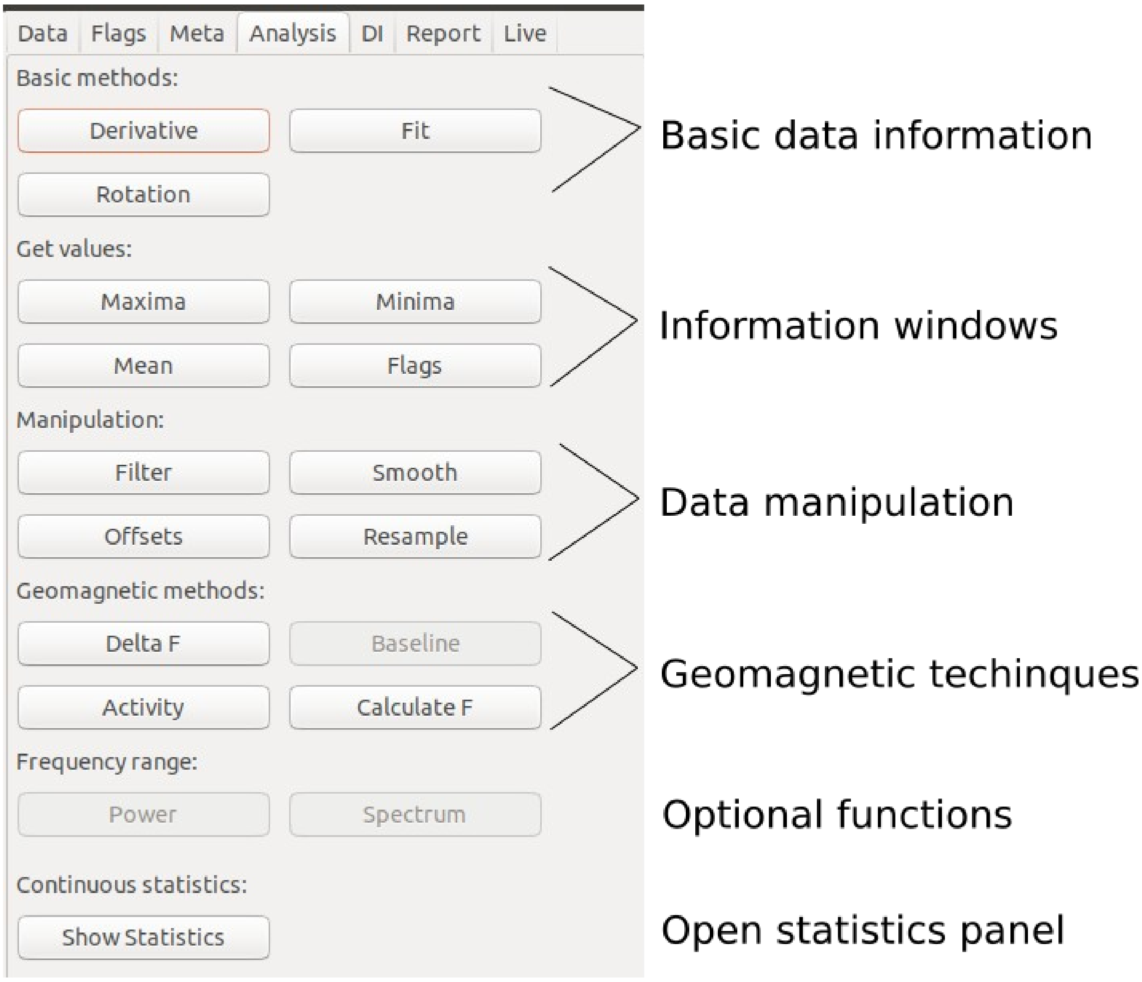


Figure 4.4.1: The analysis panel.

The analysis panel provides access to many methods which are useful for data checking and evaluation. In the following you will find subsections for buttons and underlying methods on the analysis panel.

4.4.1 Derivative

Derivative will calculate the derivative of all data columns diagrams shown in the plot. If you want the second derivative, just press the button again. The derivative method is helpful to identify strong gradients within the components, which can be related to spikes, SSC-sudden storm commencements and other artificial or natural reasons.

4.4.2 Rotation

When using the **rotation** button a window will open asking you for two rotation angles alpha and beta. The rotation function will rotate the vector (x,y,z) using the two angels alpha (a) and beta (b) into a new coordinate system (u,v,w) based on the following transformation matrix:

```
u = x*cos(b)*cos(a)-y*sin(a)+z*sin(beta)*cos(a)
v = x*cos(b)*sin(a)+y*cos(a)+z*sin(beta)*sin(a)
w = -x*np.sin(b)+z*np.cos(b)
```

When only using alpha, a „horizontal“ rotation within the x-y plane is accomplished.

4.4.3 Fit

The fit method will open an input window as shown in figure 4.4.1. The dialog window will ask you to define a fit function, its parameters as well as a time range.

When choosing **spline** as fit function, you are asked to provide a relative measure of knotsteps for the spline. The provided number defines the position of knotsteps, relative to a signal length of 1. 0.33 thus means that two knots at 33% and 66% of the time series are used. Lower numbers increase the amount of knots. When choosing **polynomial** fits, the polynomial order needs to be provided. The higher the polynomial order the more complex the fit. Further fit methods are **linear least-square**, which is equivalent to a polynamial fit with order 1, and **mean**, which will calculate an arithmetic average and use this a horizontal linear fit. Choosing **none** will remove all previously performed fits.

You can add multiple fits to a single timeseries. We will come back to this option when discussing adopted baselines.

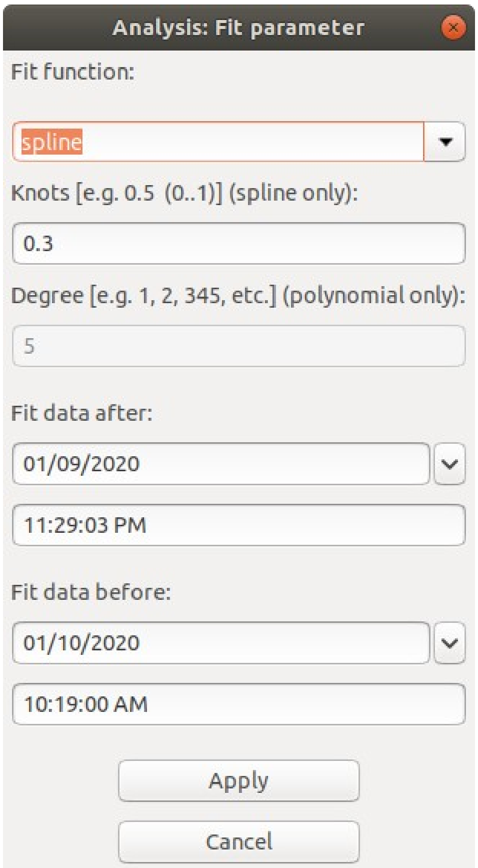


Figure 4.4.3: Fitting data dialog window.

4.4.4 Maxima, Minima, Mean

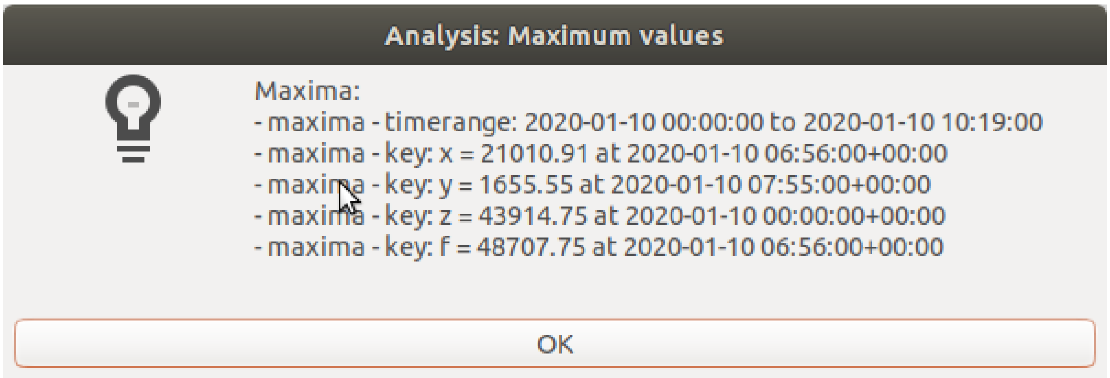


Figure 4.4.4: Information dialog on maxima values.

The three methods **maxima**, **minima** and **mean** will open an information window as shown in figure 4.4.4 providing the requested values and in case of **mean** also the standard deviation of these values for the shown data within the plotting window. You can change the active window using the plots' zoom method.

4.4.5 Flags

The **flags** button on the analysis panel will open a flag statistics dialog (figure 4.4.5). In case that no flags are currently available, the status field (figure 3.1d) will tell you and nothing else will happen.

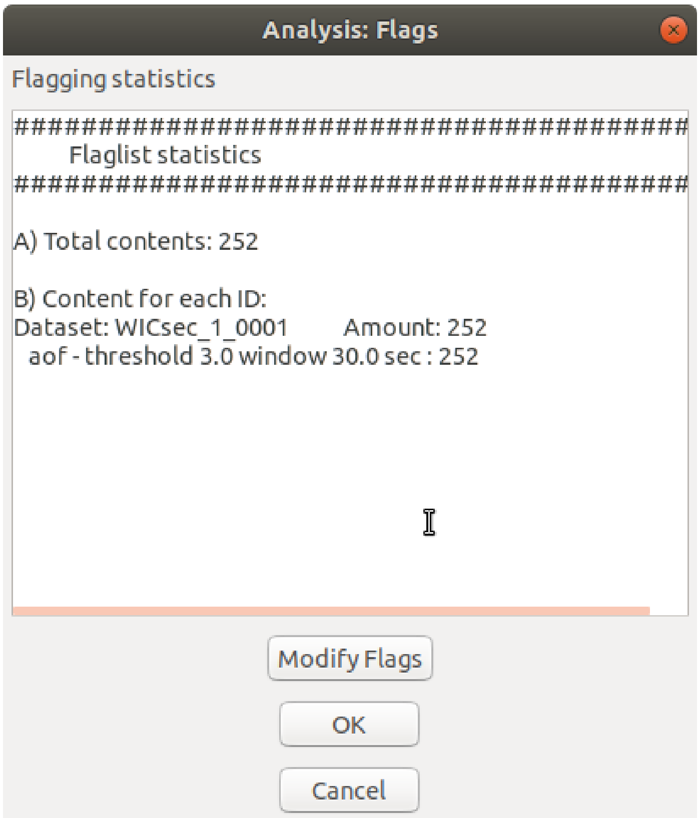


Figure 4.4.5: The flag dialog opened in the analysis panel.

The flag dialog will provide some basic information on the flags currently stored in the sensor related flaglist. As shown in the example (flagging_example.json applied to example1) , 252 individual flags are currently included. The flags have all been created by automatic outlier detection (aof = automatic outlier flag), which is also expressed by the yellow flag annotation color. 252 flags were created using the outlier detection parameters as given in the comment. Within this dialog it is also possible to modify the flagging information. Please note that this method is preliminary and more sophisticated flagging tools will be available in a future version of MagPy.

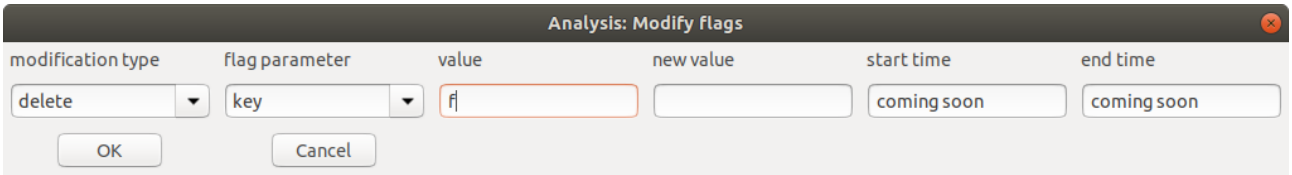


Figure 4.4.5: Modifying flags will open this dialog.

At present (MagPy1.0) you can change between different modification types (select, replace, delete), apply this type to either the flags key, comment, sensorid, or flagnumber. In above example we are deleting all flags for column key f.

4.4.6 Filter

The **filter** method will allow you apply typically geomagnetically recommended filters and also some others to your time series. Principally, filtering is a combination of smoothing using a certain smoothing window e.g. gaussian and a resampling of the smoothed time series at new intervals. The default filter parameters will be given in dependency to the sampling resolution of the time series and will correspond to IAGA/INTERMAGNET recommended filter parameters.

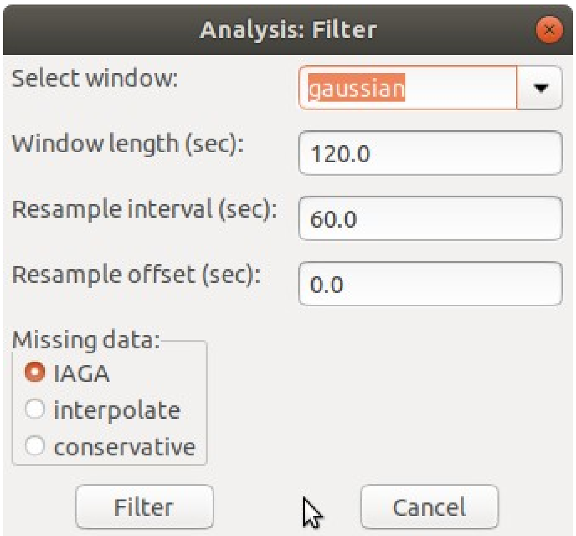


Figure 4.4.6: Default filter parameters for a one-second time series whith IAGA recommended parameters as defined in Jankowski et al. (1994).

The filter dialog allows you to select an appropriate filter window. Detailed descriptions of all possible filter windows can be found elsewhere e.g. python signal analysis. You further define a window length, a sampling interval in seconds and eventually an offset for the window center. 0.0 as shown above will center the window on the minute. Furthermore you have to select a preferred missing data treatment. By default, the IAGA recommended method will be used, which means that filtering will be performed if more than 90% of data is available within the filtering window. Please note the filter weight are always determined analytically and therefore missing data is correctly accounted for when applying the filter window. Interpolation as missing data treatment will interpolate missing data if more than 90% of data is available. If less than 90% is present than this window will not be filtered. Conservative treatment will not perform any filtering if already one data point is missing within the filter window.

4.4.7 Offset

The offset method allows you to define a certain offset in time or component for either the whole time series or the selected (zoomed) time window. Figure 4.4.7 shows an example of an offset within a time series and the offset dialog to correct this offset.

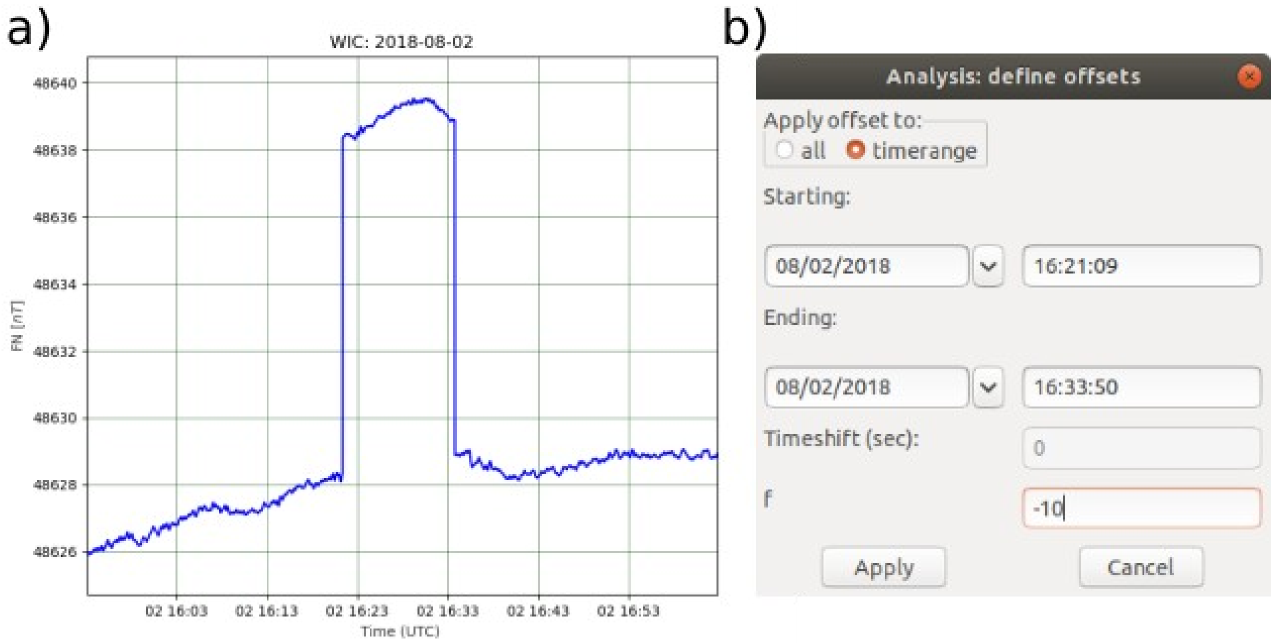


Figure 4.4.7: a) time series with an offset of 10 nT for almost 12 minutes. When using the offset button, the dialog window b) will open. Here you can provide offset values which are then applied to the time series.

4.4.8 Smooth

Smoothing data sets support the same smoothing windows as the filter method. In principle smoothing does exactly the same as the filter method without resampling the data set a new time stamps. Please check the filter function for supported methods.

4.4.9 Resample

The **resample** method can be used to subsample a time series at specific periodic intervals. If no measurements are available at the specific time stamp, than a linear extrapolation between surrounding points is used.

4.4.10 Delta F

The **delta F** method is available if vector components (x,y,z) and an independent f column is present within the active data set. When clicking on delta F, the vector sum (F[Vector]) is determined and subtracted from f (F[Scalar]).

Delta F = F[Vector] - F[Scalar]

4.4.11 Activity

The **activity** method will calculate the geomagnetic activity index k from the provided time series. The activity button is only active if geomagnetic components covering at least 1.2 days are available. Currently, the k index is solely calculated with the FMI method. Depending on the provided data eventually additional steps are automatically performed. The FMI method basically works with minute data resolution. If you have second resolution data, the filtering process is automatically performed using IAGA recommended filter options before k calculation. As the FMI method uses dynamic filter width adoption, you should provide longer timeseries as you are actually interested in, and you should be prepared for some waiting time.

You can test the activity method using example4 of the provided MagPy examples. Example4 (see 8.1) provides full vector information in one second resolution for one week, covering a weak geomagnetic storm. When applying the **activity** method, you open this file, then select (see 4.1.1) only the vector components (X,Y,Z) and the independent field strength (S) as shown in Figure 4.4.11a. Then you can press the **activity** button. The dataset will now be filtered and afterwards the FMI method will be applied resulting in figure 4.4.11b. The full process will need about 10 seconds, obviously depending on your hardware.

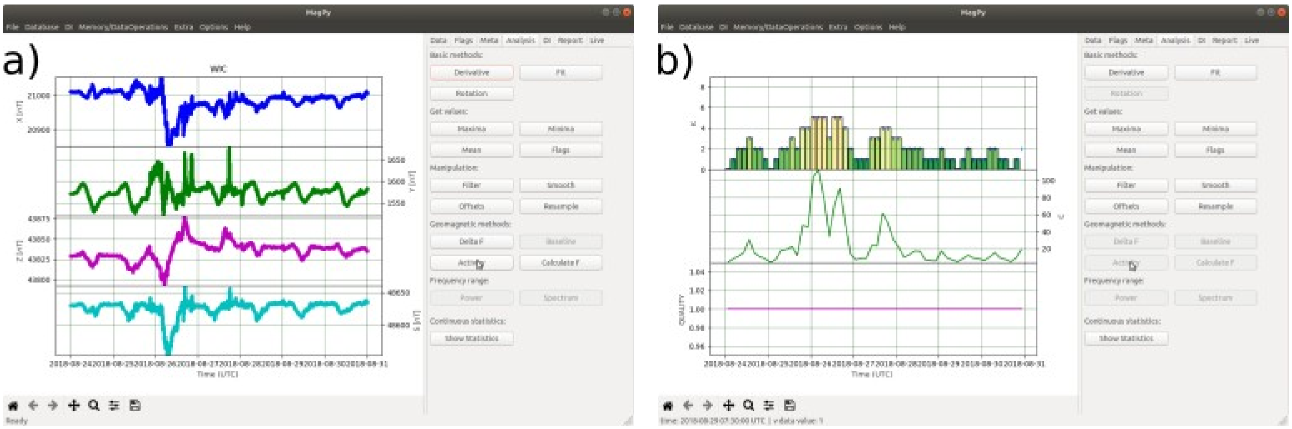


Figure 4.4.11: Application of the activity method to calculate k indicies. a) shows x,y,z and s components of example4. b) shows the k index plot after pressing the activity button. Beside k, although the daily cumulative k value and a quality factor are shown.

4.4.12 Baseline

The baseline method is only available if a basevalue data set has been opened previously and is still present in the memory. When now opening a variation time series, which is covered by the time range of available basevalues, the **baseline** button gets enabled. When pressing the baseline button you will be asked to provide fitting parameters for adopted baselines. This baseline will be calculated and a baseline correction function of the the time series will be obtained. This function will not be applied directly but stored within the time series meta information. You can apply the adopted baseline be pressing on the now available button „**Baseline Corr**(ection)“ on the **data panel**. Exporting this data set now as PYCDF will export the adopted baseline function along with the data meta info. This way, uncorrected and corrected data is not separated any more. Extensive details and a complete walkthrough with hands-on examples for this method is provided in section 6.2.

4.4.13 Calculate F

Calculate F will calculate the vector sum of x,y,z components and store the result within the reserved f column of the data object. If an independent f is already existing within this column, you will be asked whether you want to overwrite the existing information.

4.4.14 Frequency methods (under development)

The two methods **power** and **spectrum** are disabled by default as these methods are currently under development. You can activate a preliminary access in the options menu. Currently you can view power and spectral densities for individual components. Yet, it is not yet possible to access any parameters, window sizes , colors etc. This will come in a future version on MagPy.

4.4.15 Statistics

When clicking on the statistics button you can switch on/off the optional panel (e) as shown in figure 3.1e. This additional panel gives you information about data ranges in the zoomed plotting area. If less then x data points are selected you will also get indivdual values for all data points which is helpful i.e. for indepth basevalue analysis.

4.5 DI Panel

The DI panel provides all necessary functions and methods to calculate basevalues and DI timeseries from basic/residual DI fluxgate measurements.

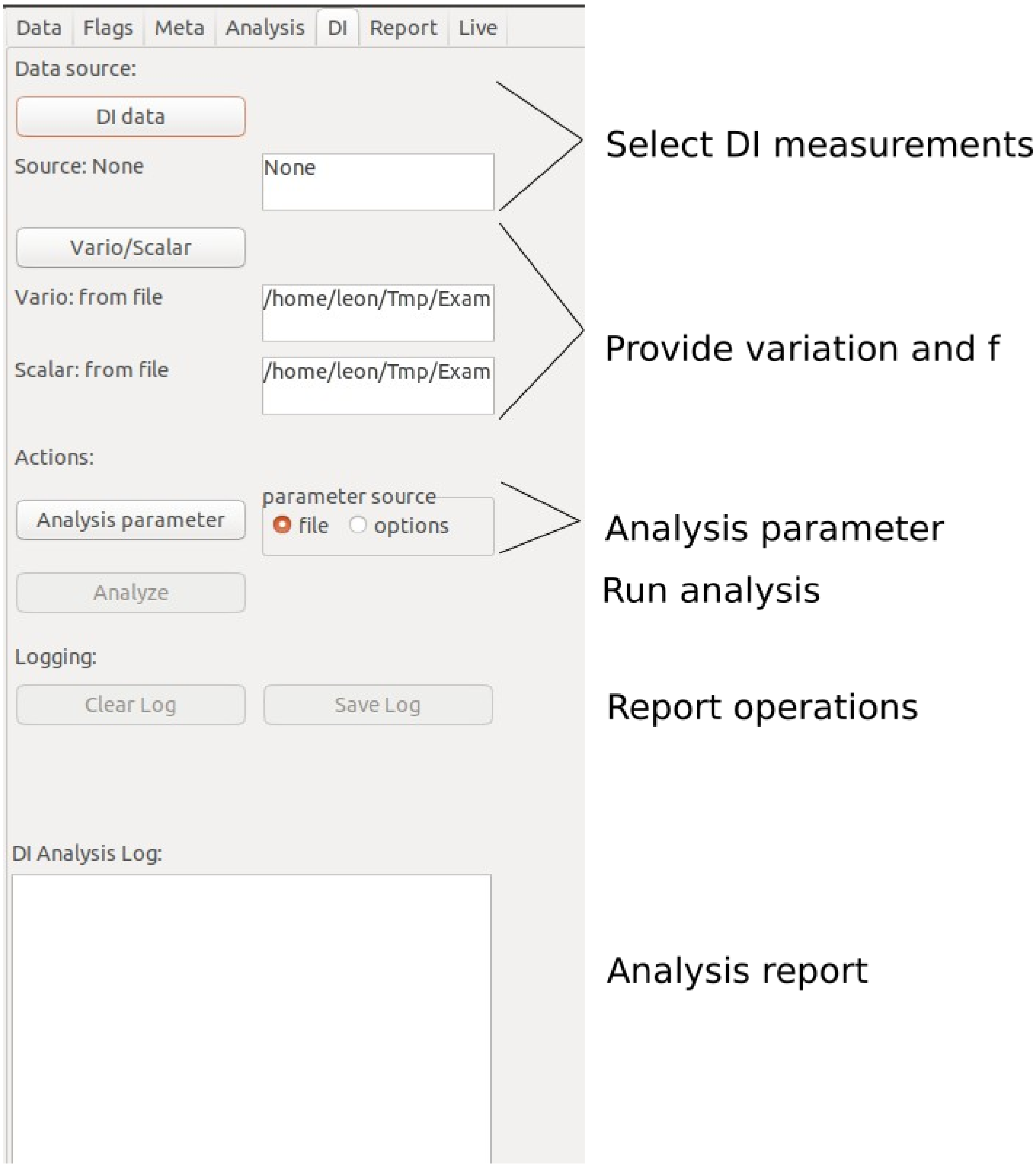


Figure 4.5.1: The DI panel. Within this panel you can access all necessary functions and methods to calculate basevalues and DI timeseries from basic DI measurements.

The upper part of the DI panel is dedicated to the definition of data sources. Firstly you need to provide access to DI measurements data (**DI data**). Such DI data can either be located on your file system, within a MagPy database, or provided by a web service. Supported data formats can be found in the appendix. When clicking on DI data, an input dialog will open to ask you for a specific data source. A detailed example application is shown in section 6.2.

In order to regard for geomagnetic variation throughout the DI measurement and obtaining basevalues you further require variation data and measurements of F. Scalar F measurements can either be provided along with the DI data set, as usually the case if you measure F before or after the DI measurements at the same pier, or you can provide access to a continuously measured F using **vario/scalar**. Access to variometer data is provided in the same way. All MagPy supported data formats are also supported for this analysis.

Before analyzing the data set you can modify/set **analysis parameters**. Such parameter include pier differences between continuously recording instruments and the DI pier, as well as threshold parameters etc. It is recommended to provide such parameters along with the meta information of the DI data.

When finally using the **analysis** button, which will be enabled if sufficient information is provided, a DI analysis will be performed, the results will be written to the logging window and can be saved as ASCII txt files, and the plot panel will show the resulting basevalues time series. By default components D, I, F as well as the basevalues baseH, baseD and baseZ will be shown. Other parameters like collimation angles are availlable (4.1.1).

4.6 Report panel

The report panel will provide a summary of all analysis steps which have been performed so far. This report might be used for better transparency and reproduceability of analysis routines. You can save the report messages as ASCII txt file by using the **Save log** button.

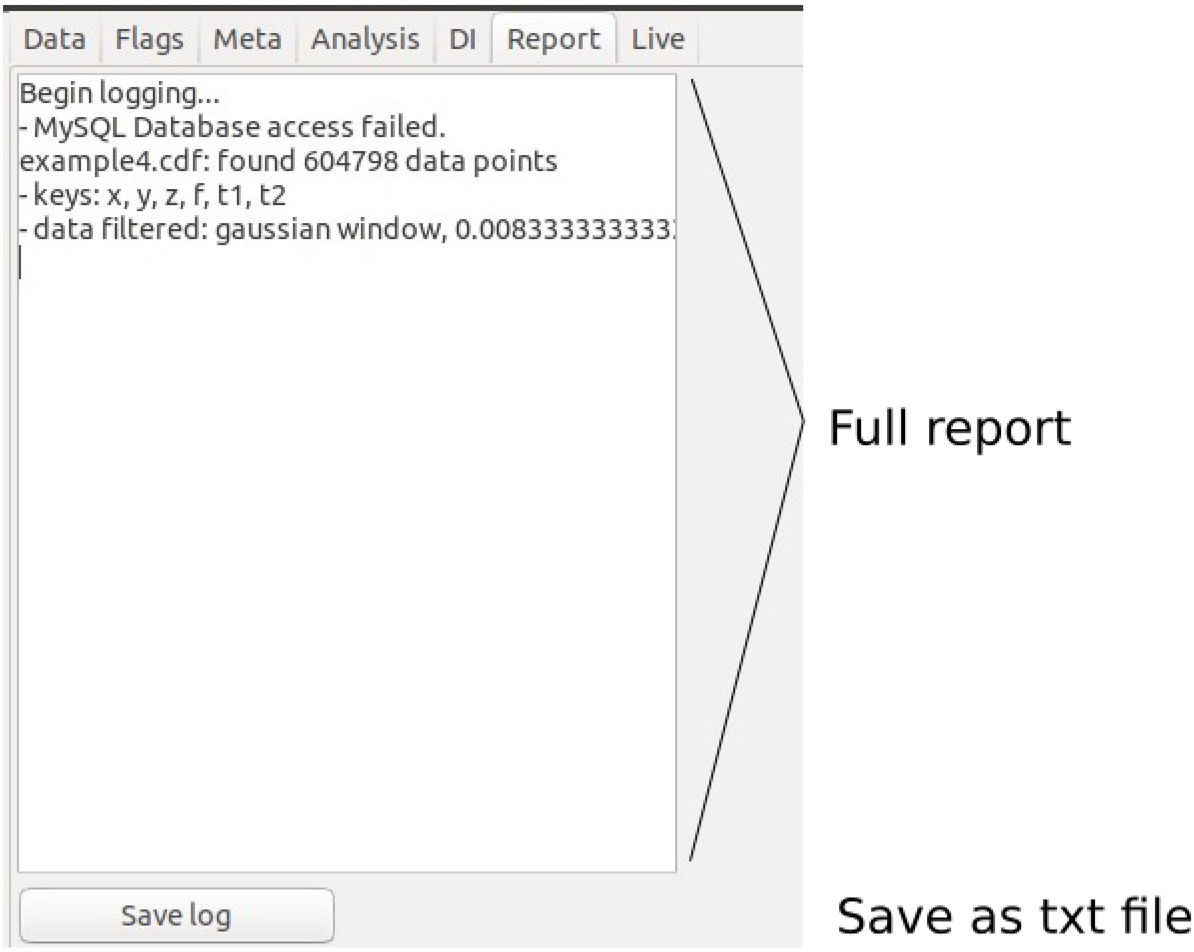


Figure 4.6.1: The report panel.

4.7 Monitor panel

The monitor panel gives you access to MagPy supported live data sources. Currently (MagPy1.0), two data sources are possible. The first one is MQTT based real-time data transmission as it is accomplished by MARTAS (**M**agPys **a**utomatic **r**eal-**t**ime **a**cquistion **s**ystem). The second data source is MARCOS which refers to an existing MagPy database. Therefore the data content of the database has to be updated in real-time by any kind of uploading process. Real time access requires a proper installation of at least one of these systems. For public usage (at present, early 2020) a test channel for MARTAS is available, providing access to real time data up to 10Hz resolution from a remote variometer station of the ZAMG. Please contact the Conrad Observatory, if you want to view this test channel.

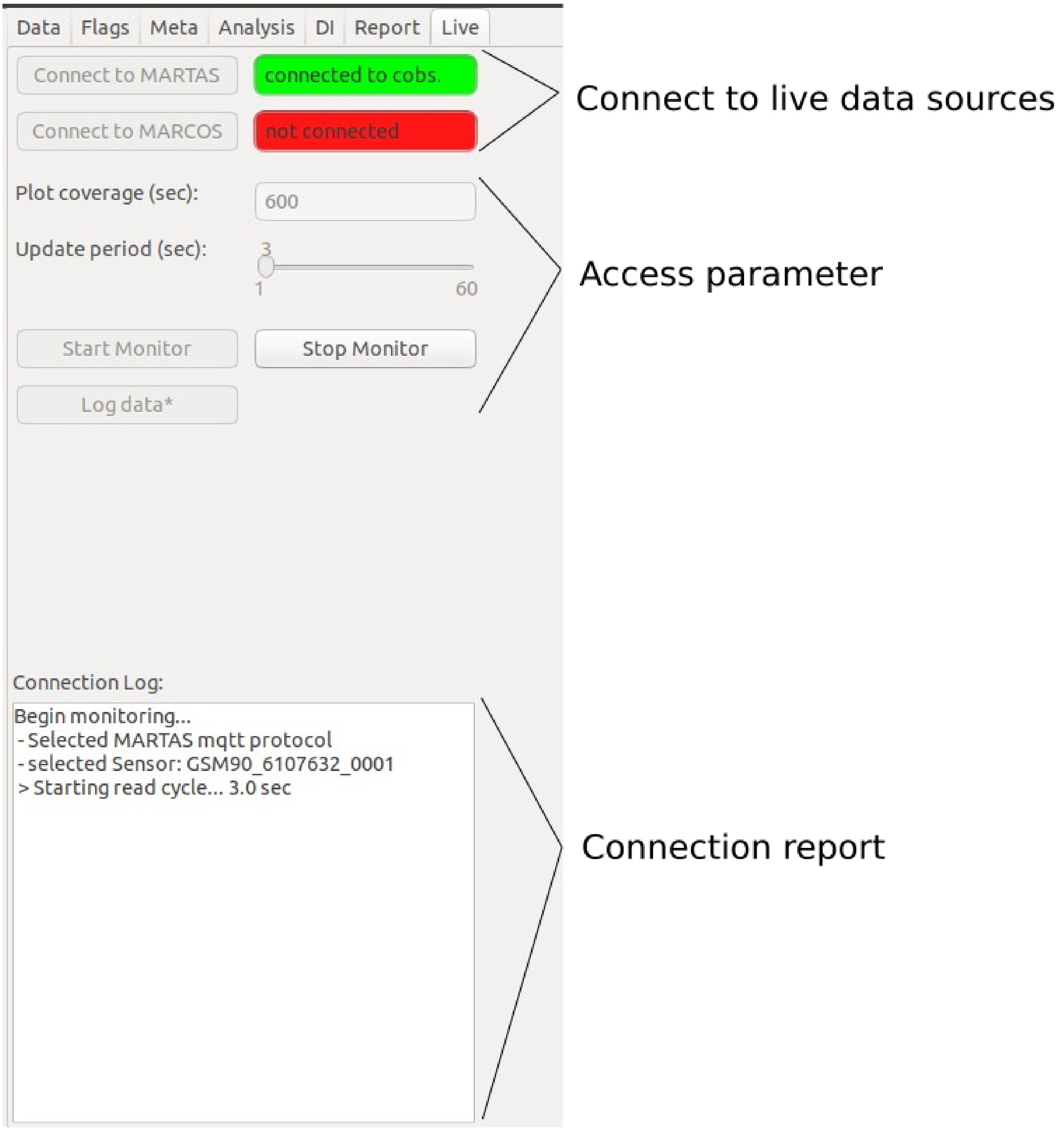


Figure 4.7.1: The monitor panel provides access to supported live data sources.

5. Main menu

The main menu is shown in figure 5.0. The main menu gives you access to general methods, options affecting all data sets and multiple stream operations. Further you will find several options dialogs, data input sheets and help.

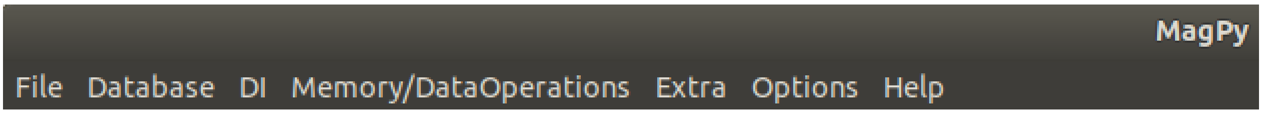


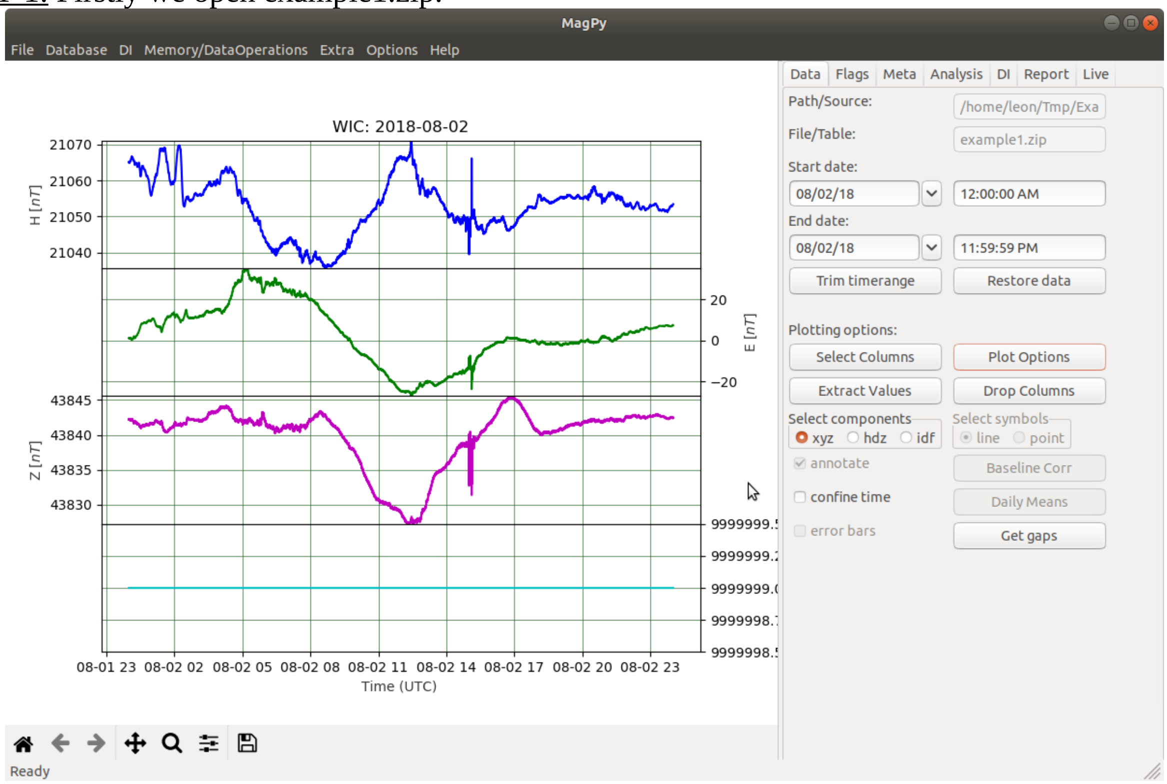
Figure 5.0: The main menu with its seven sections.

The file section has been discussed already in 3.1. In following you will now find descriptions of the other menu items. The given order is not related to the menu position, but to the frequency how often we are using them.

5.1 Memory and multiple data operations

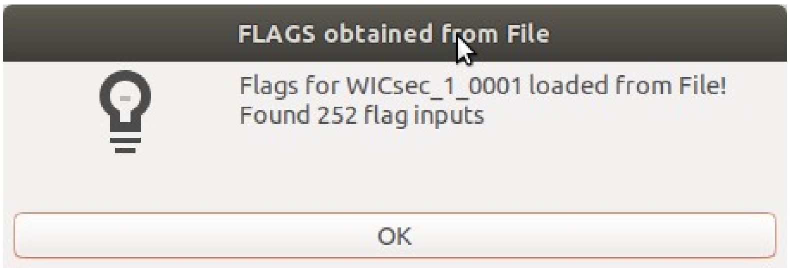
The main menu item **Memory/Data operations** provides helpful tools for multiple data operations and also provides the possibility to store certain working states and access them a later stage. When clicking on this menu item, a drop down will give you access to two submenus: 1) **Add current working state to memory…** and 2) **Multiple data operations…** . The usage of these submenus is described in the following using a commonly occurring geomagnetic analysis case, based on example1 and example2. Example1 contains plain variation data for a single day, but no scalar information. Our aim is to merge scalar information into this data set and to export a complete IAGA-2002 variation data set. For this example we also remove the lighting strikes based on existing flagging information in flagging_example.

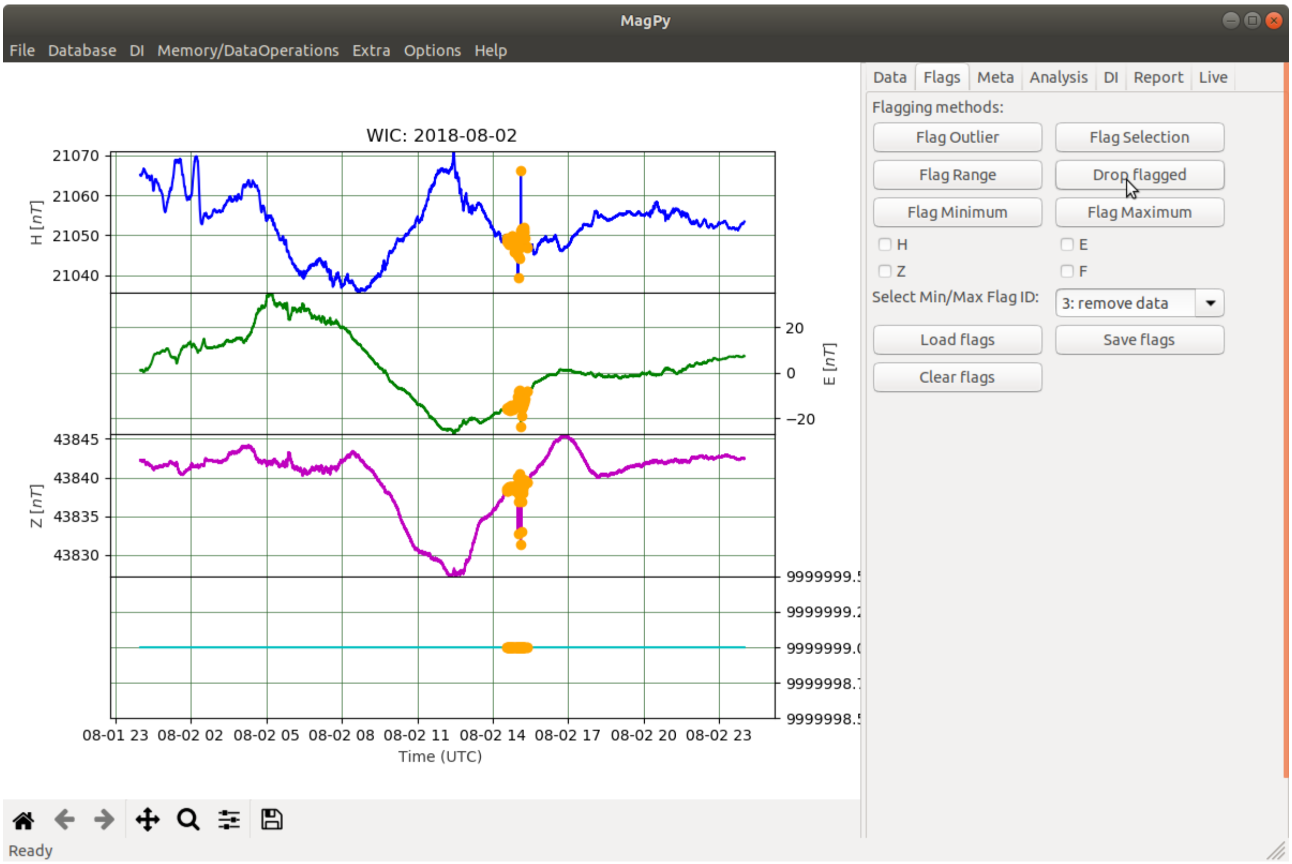
<u>STEP 1:</u> Firstly we open example1.zip.



STEP 2: Secondly we load some flagging information to mark the lightning strikes.

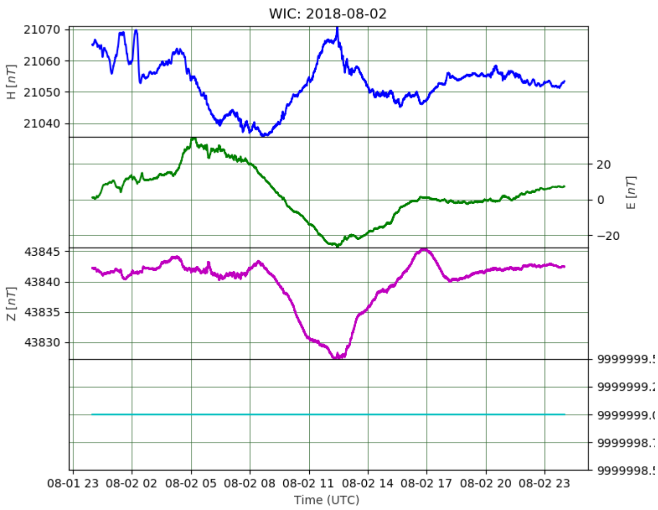
Flags Panel → Load flags → locate example directory and open flagging_example.json





STEP 3: Then we drop flagged data.

Flags Panel → Drop flagged

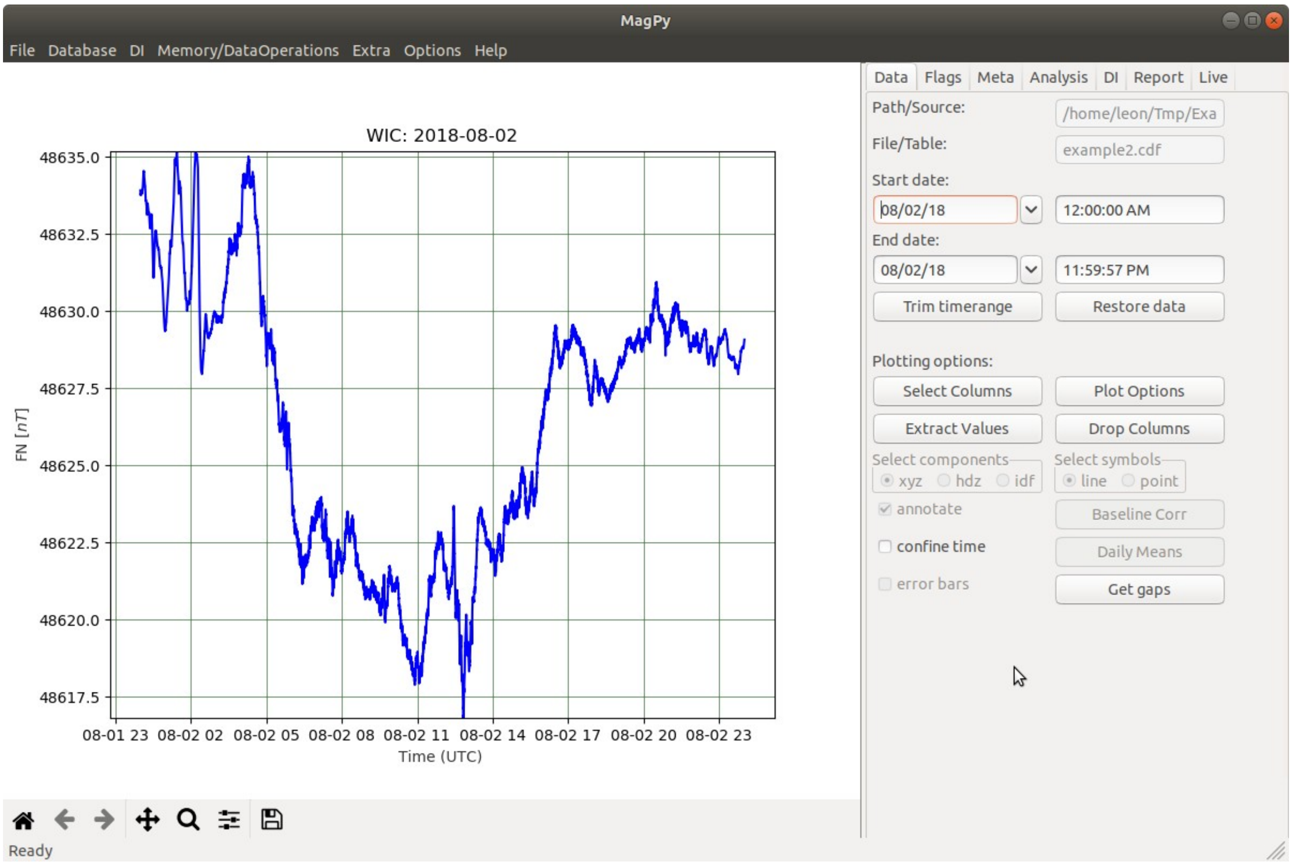


STEP 4: In the following we add the current working state to the memory.

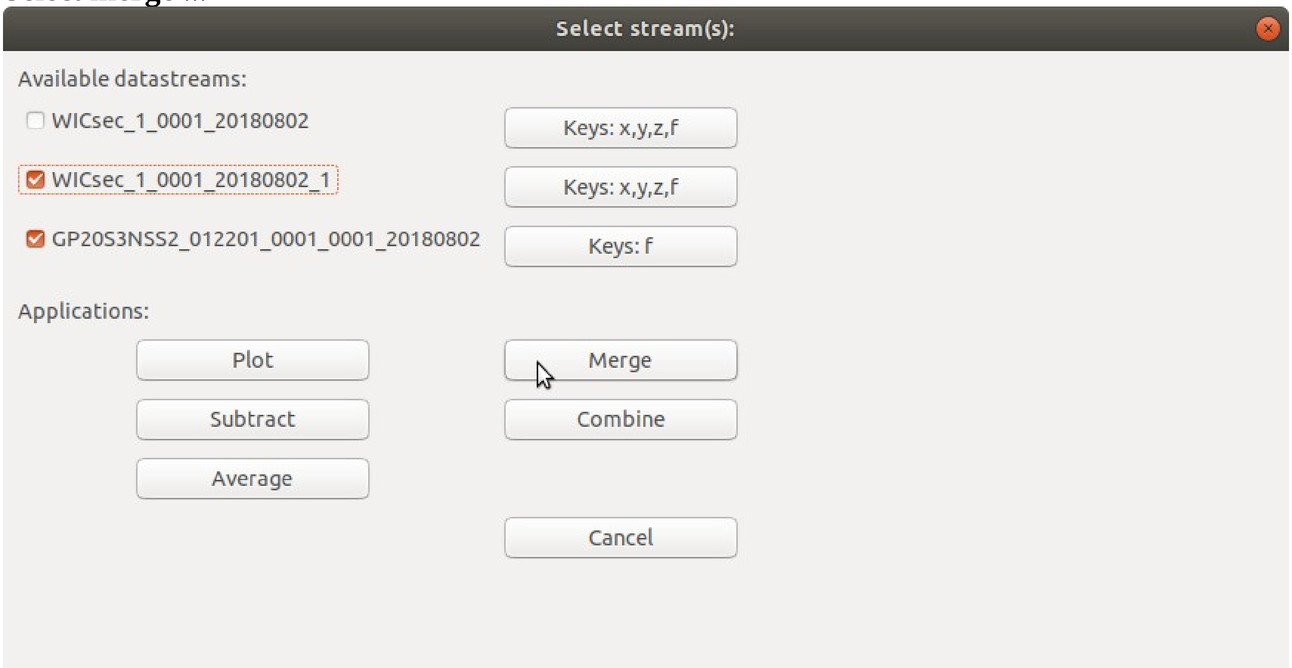
MainMenu → Memory/Multiple operations → Add current working state ...

STEP 5: Now we open scalar data for the same day: example2.cdf

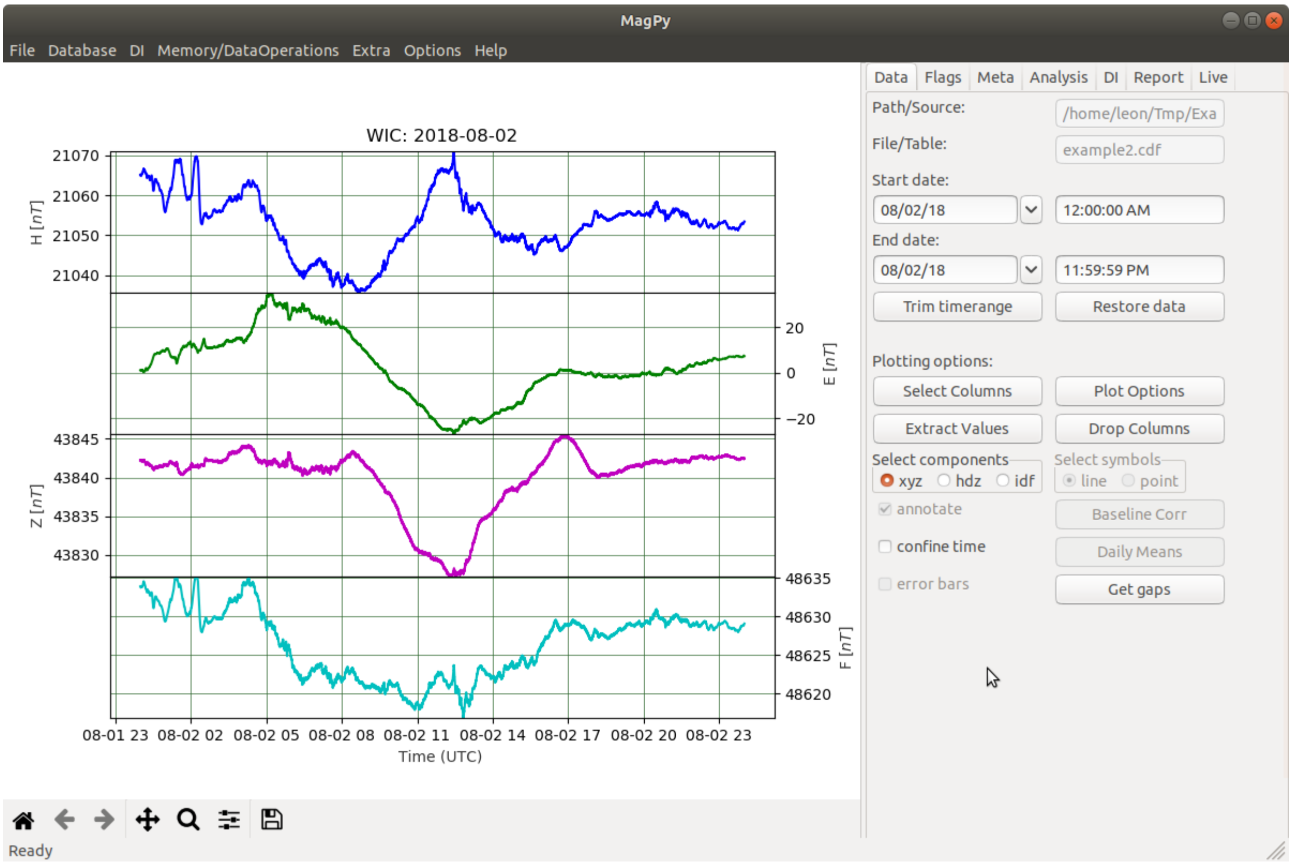
MainMenu → Files… → Open Files → *Open example2.cdf*



STEP 6: Access Multiple data operations and select the last variation data and the scalar data set. Select **merge** ...



STEP 7: That’s it. Now you can export the merged time series.



Besides the merge operation there are other options. All these options are described below:
Merge will add data from a secondary source into a primary source. If some data is already existing this will remain untouched and only gaps will be filled by the merged secondary source. Typical place holders as used in IAGA files (99999, 88888) will be treated as non-existing data. The time series will always be limited to the time range of the primary source. You cannot merge more than two data sets at once.
Subtract will subtract identical columns from two data sets. You can only subtract two data sets at once.
Combine will join two time series and fill non-existing columns. If data is already existing, this remains untouched. Join can also be used to combine two adjacent time series.
Plot will plot the selected components of the selected time series in one plot. Please note that this makes only sense if selected time series cover a similar time interval.
Average will calculate the mean of similar columns of all selected time series. If means of x,y,z and or f components are calculated, then also standard deviations (if more then 2 records) or min/max differences are provided in the data object.

5.2 DI data input

The DI data menu item has two submenus. **Load DI data …** will give you the possibility to load DI data into the memory. This method is exactly the same as the Load DI data button on the DI panel. The second submenu is **Open input sheet ...**
This option will open a dialog window which corresponds to a typical DI measurement input sheet as it it used for example in Niemegk, Conrad, and other observatories (figure 5.2.1). If you have loaded DI data already into the memory you will find a drop down list on the upper left, which gives you access to this data.

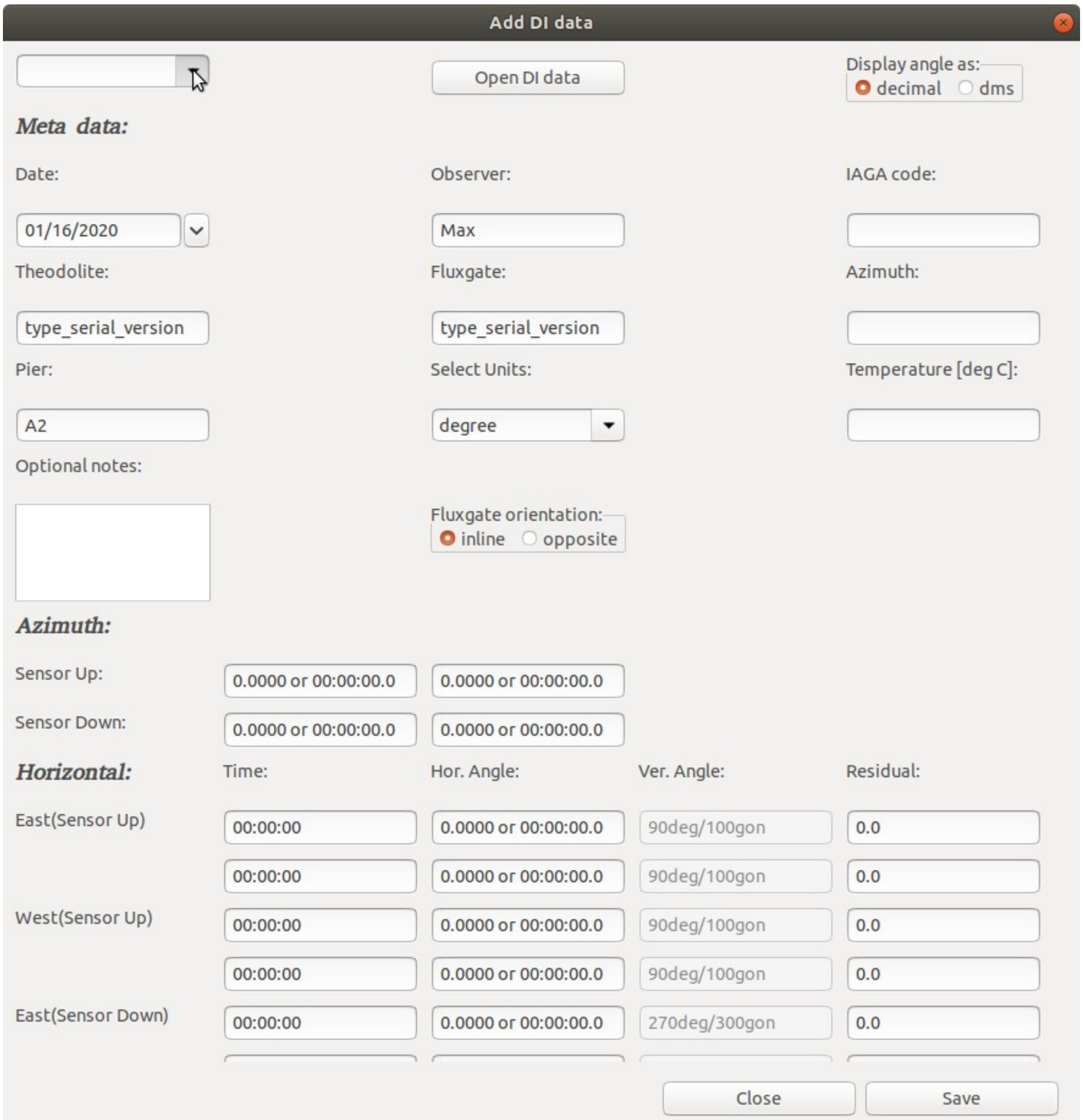


Figure 5.2.1: The Input sheet for DI data.

The input sheet allows you to load already existing data or to fill-in any new measurement data and store it. If it is used for data input it is recommended to create a template file with recurring meta information and load the template first. At present you can save the data sheets to local files only. Saving will create an ASCII text file with complete information (see appendix 8.2). The input sheet can also be modified in order to reflect different measurement orders, amount of individual measurements and usage of scale value tests. If you are not using the residual method, just leave 0 as input value there.
The input sheet is modified within the general options menu as described in section 5.4.

The fields of the input sheet should be widely self-explaining for anybody experienced in DI measurements. As mentioned already, the input sheet supports residual and zero measurements. If you are using residual measurements it is necessary to consider the orientation of the fluxgate probe on the theodolite i.e. whether it is inline or opposite to the optical direction.
If you are performing F measurements at the same pier prior to or after the DI measurement you can type them in as well. Alternatively you can upload any supported F data file e.g. of a GSM19

system. You can load example6a.txt or example6b.txt into the input sheet to get an impression on how to fill in parameters.

When saving DI data to a file, the file name will automatically be generated from date and time of the first measurement, from the name of the pier and the station code. The file is an ASCII text file. If an identical file name is already existing you can either overwrite it or save the data as “alternative”. This second option will add one-second to the time in the file name (not to the data itself). Such alternative data might be used, if you have measured two different azimuth marks. Type in all data for the primary azimuth mark, save it, change the azimuth values to the second mark and save again as alterantive.

5.3 Database connections

To temporarily connect to a MagPy database you can use

MainMenu → DataBase → Connect MySQL DB…

This will open a dialog window to define location, name and credentials for the MySQL/MariaDB database. Please note that this usage requires a existing and accessible MagPy database.
For permanent database connection please use the **options** menu (5.4), **basic initialization parameters**.

If you want to create a new MagPy database you can follow the steps below to establish one. Please note that these instructions are valid for a Linux system and might be modified for other systems.

STEP 1: Install and configure MariaDB or MySQL if not already existing (please refer to related instructions)

Then run MySQL

```
$ sudo mysql -u root -p mysql
```

STEP 2: Create a new empty database (in this example called mydb, please change).

```
mysql> CREATE database mydb;
```

STEP 3: Create a new user and give him permission on this database

```
mysql> GRANT ALL PRIVILEGES ON mydb.* to myuser IDENTIFIED BY mypasswd;
```

STEP 4: Initialize this database with a MagPy supported table structure
XMagPy:

MainMenu → Database → Initialize new database…

You can now connect to this database and use it. Simply test it by storing flagging information or meta information into the database.

5.4 Options and Help menu

The options menu provides access to two submenus for **basic initialization parameters** and global **DI analysis parameters**. Within the basic initialization parameters you can for example specify a permanent database connection which then will always be started with the program. The are further self explaining default parameters which can be set there.
The DI initialization parameter option is useful, if you want to use MagPy‘s DI input sheets, as you can change the layout here (Figure 5.4.1). The text-edit provides the order of measurements as shown in the input sheet: MU (mire up), MD (mire down), EU (east up), WD (west up)… provide the current order. You can modify that for example by simply changing …, EU, WU, ... to …, WU, EU, …. Please leave the rest unchanged. If you are not using repeated measurements for each position, disable the **repeated positions** check-mark. If you are not using a residual method or do not use scale value checks then disable **scale value**.

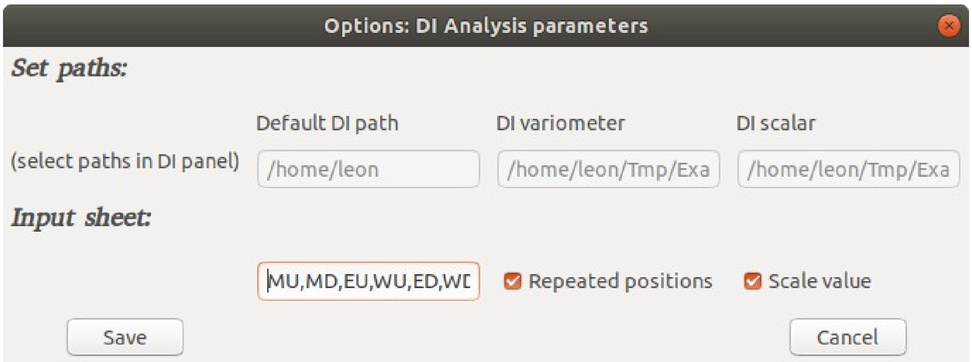


Figure 5.4.1: Modifying DI parameters for the input sheet.

5.5 Specials and extra

The extra menu has two submenus. A complete recipe is devoted to the **Check definite data** option (section 6.3). **Open MagPy log** will open a text file with the content of MagPy‘s back-end log file, which might be helpful for debugging and workflow control. If you face any problems with MagPy it is always helpful to provide this log file along with your error report.

6. Recipes

6.1 Daily data quality check and flagging

This description will cover the following aspects: open data from variometer and scalar sources, merging data sources, checking delta F, flag data, save flags and data.
In order to test recipe 6.1 we will use example1.zip and example2.cdf, as well as the flag file recipe1_flags.json. Please copy the example files into a temporary directory as we will modify some of them.

STEP 1: Loading variometer data (example1.zip)

MainMenu → File → OpenFile... → *select example1.zip*

When quickly looking at the time series you already see some increased disturbance at 16 UTC. Such rapid variations within the components can be amplified using a derivative of the data set.

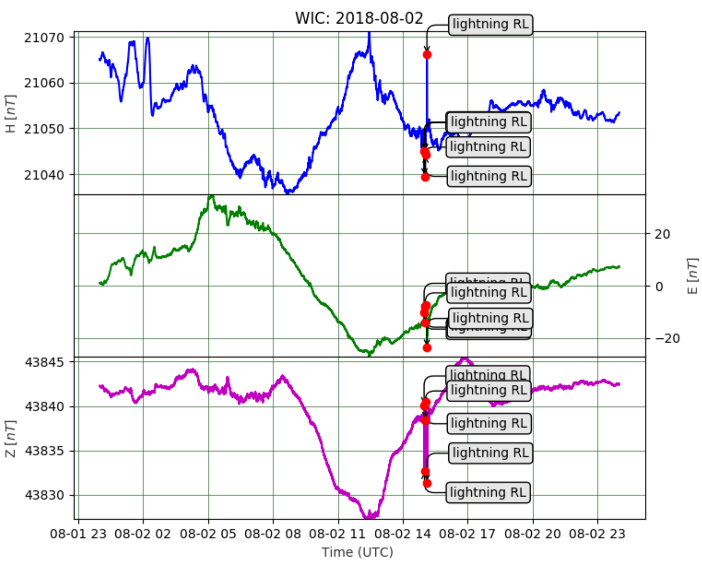
Remember the time range and restore the original data. For the plots shown below we used

Data Panel → Select Columns → *select x,y,z*

STEP 2: Initial flagging of variometer data

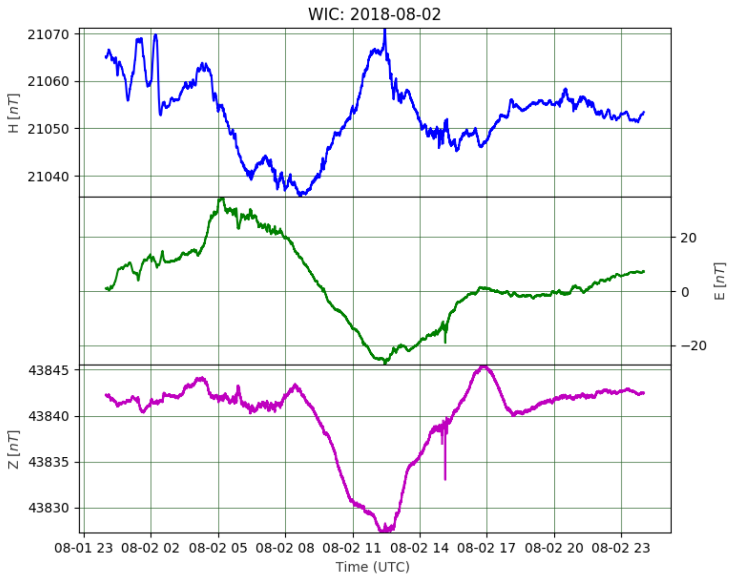
Load eventually existing flagging information (recipe1_flags.json).

Flags Panel → Load Flags → *select recipe1_flags.json*



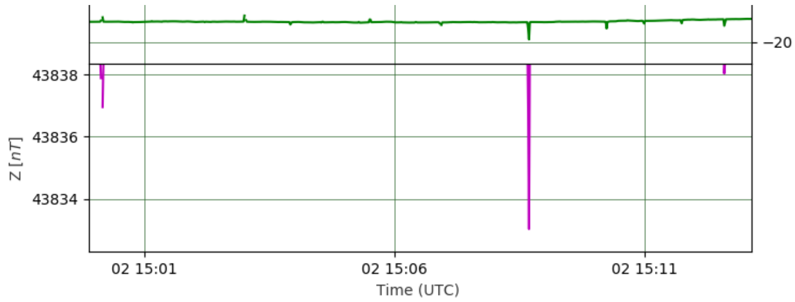
Review flags and, if ok, drop the already flagged data. If not ok, create new flags over existing. Here we first drop already existing flags.

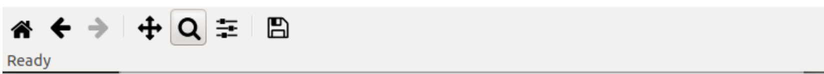
Flags Panel → Drop flagged



Zoom into Z-component and flag the spikes using **flag selection**.

Plot Area → Zoom … Flags panel → Flag Selection



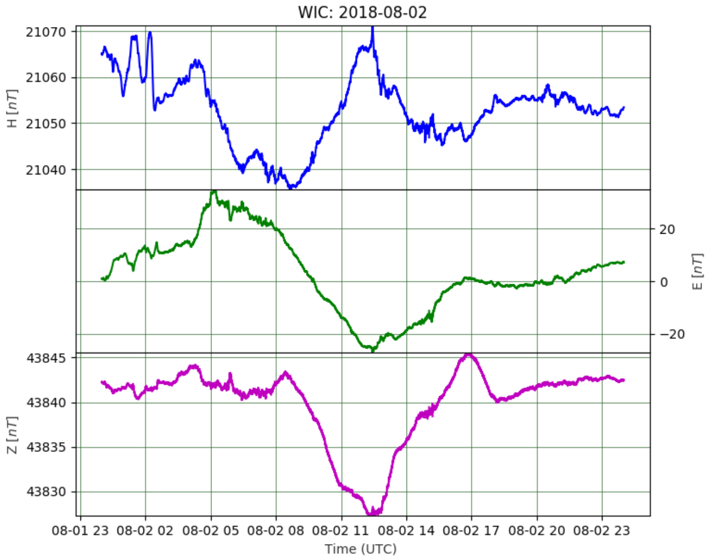


Now save the new flagging information into the recipe1_flags.json file. Please note, that although you will get a warning that the file is overwritten, this is not the case. New flagging information will always be appended to the already existing data. Each flag will also be connected with a creation date. In case you modified earlier information, the original flag will still be present in the archive. But only the ones with the latest creation dates will be active.

Flags panel → Save Flags

For continuation we will now drop the newly flagged data as well.

Flags panel → Drop Flagged



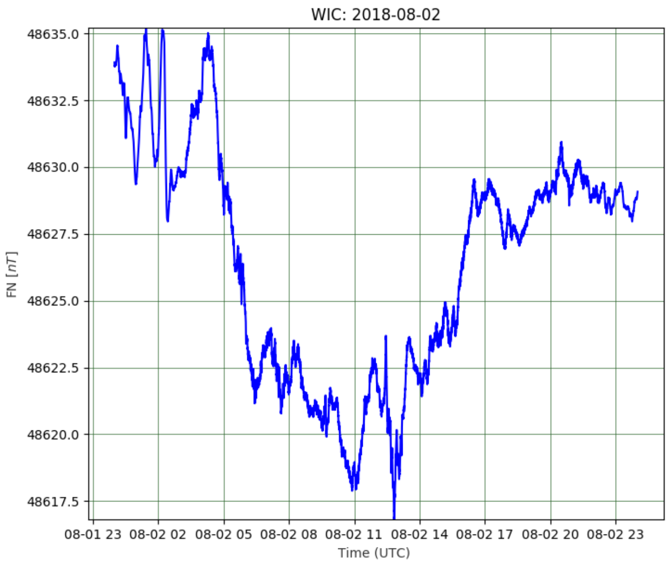
You can now reload and apply all flags any time. Just open example1.zip again and then apply the flags (Load flags) of your modified recipe1_flags.json file.
If you zoom-in again into the data set to times at about 15:00 UTC you will see that there are many more spikes due to lightning. Feel free to flag them as well or wait until Step 6 for some alternative for strongly disturbed sequences.

Finally we will store our current working state.

MainMenu → Memory/DataOperations → Add current working state...

STEP 3: Loading scalar data (example2.cdf)

Load scalar data and eventually perform similar steps as shown above. Here we will skip the flagging procedure as disturbed data has already been removed from example2.cdf.



If you have performed any flagging or other modification (offsets etc) then please do not forget to add the final working state into the memory.

STEP 4: Merging cleaned variometer data and cleaned scalar data, delta F

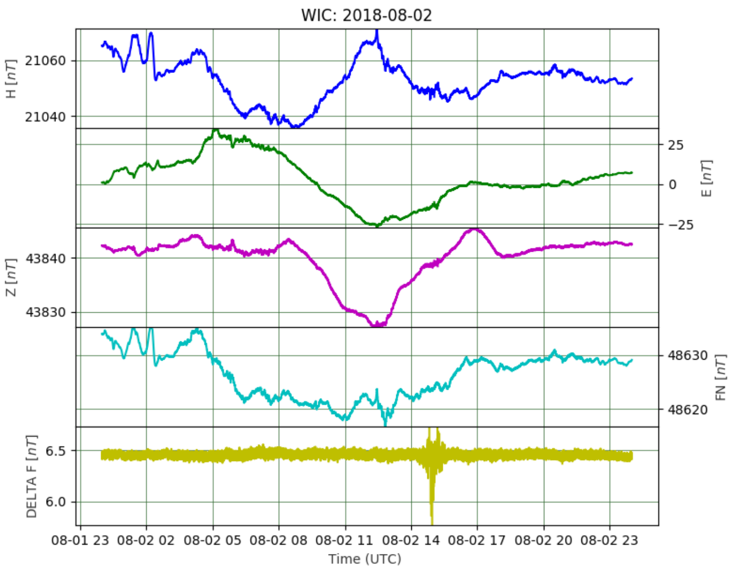
Now it is time to merge both data sets into a single time series. Please follow the instructions as given in section 5.1.

MainMenu → Memory/DataOperations → Access memory ...

Select the last variometer state and the last scalar state and select **merge** within the dialog.
Then use

Analysis Panel → Delta F

to calculate delta F between the vector sum and the independent scalar F.



The delta F value clearly shows that some significant disturbances have not yet been appropriately considered during the flagging procedure. Thus you can go back now, intensify flagging and and repeat until you are satisfied with the data quality

STEP 5: Exporting final data

We export the combined data set as follows (please note that no baseline information has been considered so far).

MainMenu → File → Export...

Please select a folder of your choice and a suitable format of your choice. We selected IAGA format. A snapshot of the resulting file is shown. Please note that delta F (G) values are used for the file, if delta F's have been calculated from an independent source.

```
 Data Interval Type      1-second (501-1500)                          |
 Data Type               variation                                    |
 # K9-limit              500                                          |
 # V-Instrument          GP20S3NSS2_012201_0001                       |
 # F-Instrument          WICsec_1_0001                                |
 # File created by       MagPy 0.9.1                                  |
DATE       TIME         DOY     WICX      WICY      WICZ      WICG    |
2018-08-02 00:00:00.000 214     21065.18      1.11  43842.24      6.44
2018-08-02 00:00:01.000 214     21065.17      1.10  43842.24      6.44
2018-08-02 00:00:02.000 214     21065.16      1.10  43842.24      6.44
2018-08-02 00:00:03.000 214     21065.15      1.10  43842.27      6.46
2018-08-02 00:00:04.000 214     21065.13      1.10  43842.27      6.46
2018-08-02 00:00:05.000 214     21065.13      1.10  43842.26      6.45
```

STEP 6: Alternative flagging procedure for strongly disturbed sequences

Open example1.zip again, load existing flags from recipe1_flags.json, drop the flagged data. Now, we trim the resulting timeseries to the range of increased disturbances.

Data Panel → Start date → *change time to 02:30:00 PM*

Data Panel → End date → *change time to 03:30:00 PM*

Then cut the time series

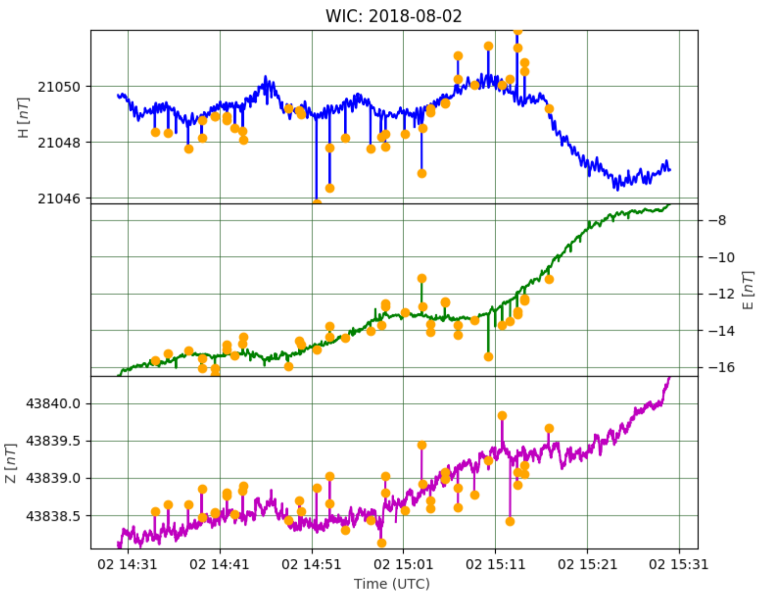
Data Panel → Trim timerange

Now go to the Flags Panel and use the **Flag Outlier** method:

Flags Panel → Flag Outlier

Modify the automatic flagging parameters as follows and **apply**:

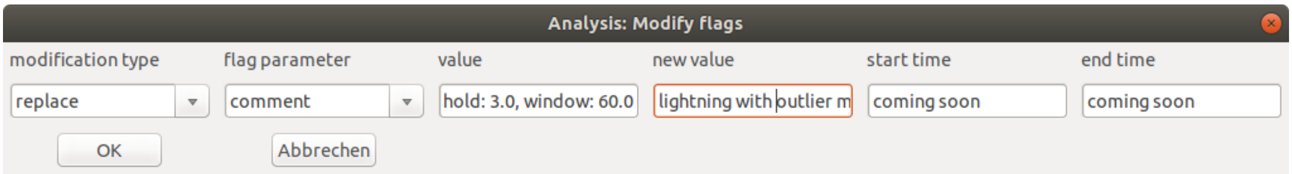
This will result in the following plot indicating that we got almost all spikes now, even very small ones.



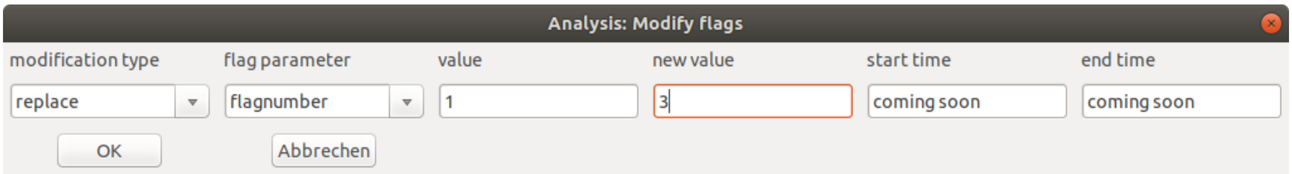
Finally, we want to change comments and flag reference.

Analysis Panel → Flags

From the flagging statistics field copy “aof - threshold: 3.0, window: 60.0”. Then go to **Modify Flags**. Choose the items in the dialog as follows:



You might also want to change the flag ID number from 1 (automatic) to 3 (observer reviewed, not usable for definitive analysis). Use the **Modify Flags** Button again.



You will see the updated information in the statistics field and the when pressing OK, flagging information in the plot will be updated. Save flags

Flags Panel → Save Flags → *use the modified recipe1_flags.json again*

Then reload the full example1.zip (or use Data Panel → restore), reload flags, etc...

6.2 Basevalue analysis and adopting baselines

Recipe 6.2 will give you an overview about analyzing DI measurement data, obtaining basevalues, getting adopted baselines and applying baseline corrections to variation data in order to produce adjusted, quasi-definitive and definitive data sets. For this recipe we will use example3.txt, example4.cdf as well as example6a.txt and example6b.txt.

Firstly we will look at the input sheet.

MainMenu → DI → Open Input sheet ...

Please check out section 5.2 for more details on the sheet itself. In the Input sheet dialog we click on **Open DI data** and locate/open example6a.txt. Here you will have an example of a measurement using the residual method. It is recommended to use serial number for unique identification of theodolite and fluxgate. Also a self-defined revision number, for the theodolite might be useful to verify e.g. maintenance changes for fluxgate alignment and thus the collimation angles.

DI data files usually contain only measurements directly connected to the DI pier and thus theodolite measurements as well as F data prior to or after the directional measurements. In order to analyze such data and to calculate absolute values of declination D and inclination I for a specific time we will need to consider magnetic field variations throughout the DI measurement period. For our example data we need both, a continuously recording variometer and also scalar data, as well as the pier differences between continuously recording instruments and the DI pier.

In order to analyze DI data files we now go to the DI panel, and firstly load DI data files into the memory.

DI Panel → DI data → *select both example6a.txt and example6b.txt*

The source information will then show "file" and you will see that two data sets from station code "WIC" have been loaded. Next, we will have to access variometer and scalar data for this time range. Such data is contained in example5.sec.

DI Panel → Vario/Scalar → *select "file" and the example folder for both and choose *.sec*

Example5.sec contains variation data and independent F from the observatory. In order to consider the pier difference between F and DI pier, please modify **Analysis Parameter**.

DI Panel → Analysis Parameter

For our analysis the delta F values have already been considered in the F record and other deltas are negligible. We can directly use the **Analyze** button. This will result in a plot of D,I,F and basevalues as well as a detailed report on the results in the **Analysis Log** window of the DI panel (Figure 6.2.1).

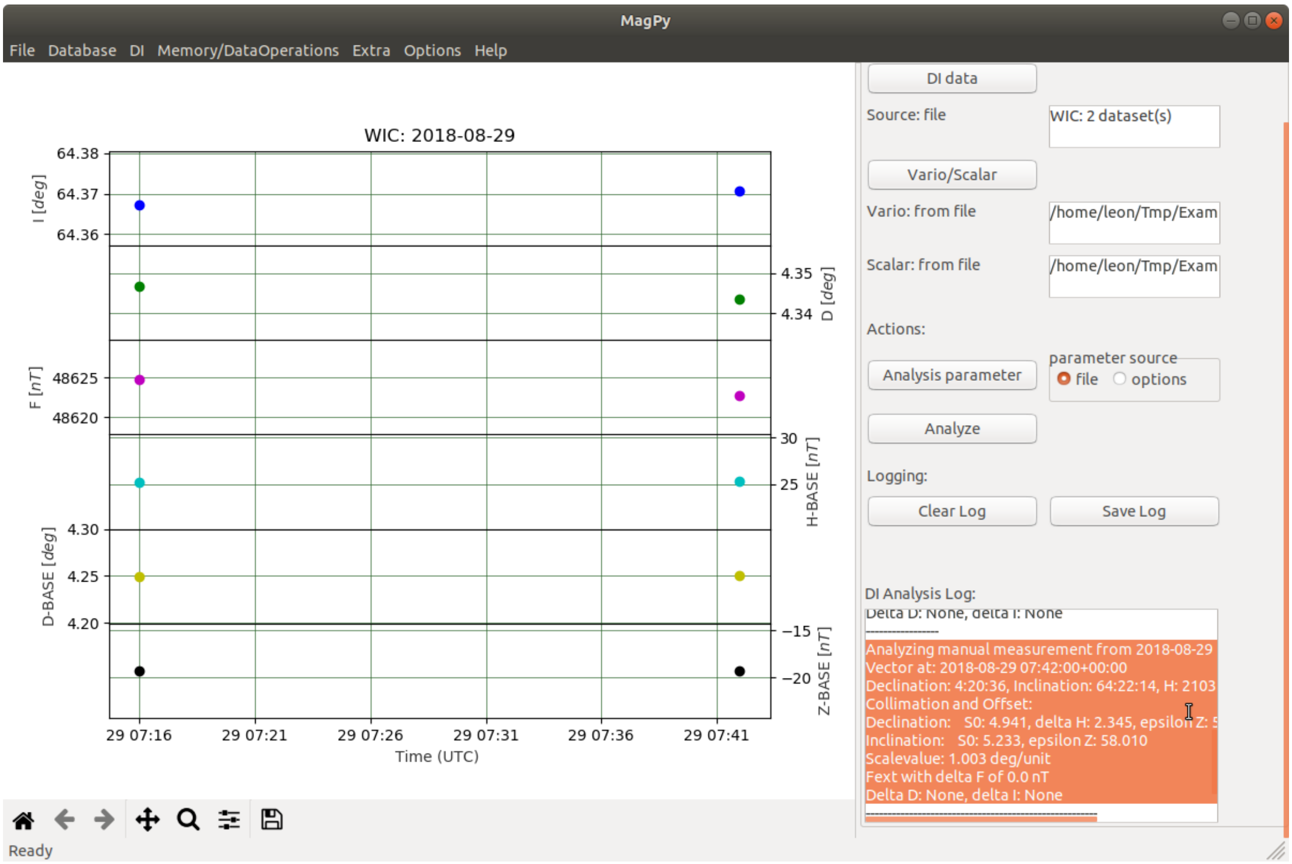


Figure 6.2.1: After DI analysis is finished, the screen will look similar as in this figure. You can change padding ranges in the graph using the plot options method of the Data panel.

We can also export all information using

MainMenu → File → Export → *choose PYSTR format*

Please choose PYSTR as output format, modify name, coverage and path, and check the contents of the newly created file using a text editor of your choice. You will see that all analysis information is contained in this time series file. For further analysis we will now switch to example3.txt as this file contains such DI analysis data for one year.

MainMenu → File → Open File → load *example3.txt*

When opening this data file you get figure 6.2.2. You will find many individual DI measurements for one year. All these measurements are performed at a single pier ("A2") and for analysis we used the same variometer/scalar instruments, thus obtaining base values for these instruments. Please note that measurements from other piers are saved in separate files.

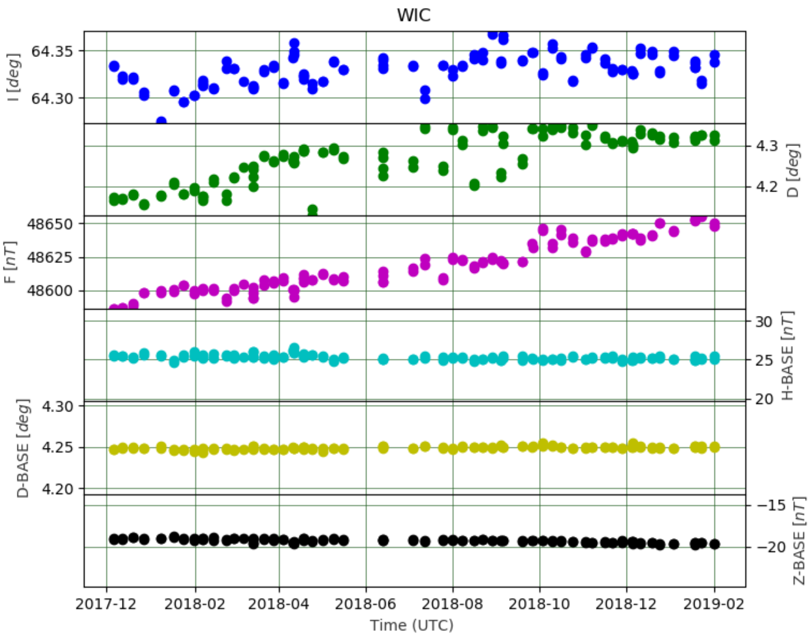


Figure 6.2.2: Basevalues for 2018 for pier A2 using a Lemi036 variometer and a GP20S3 potassium scalar magnetometer as reference. Only measurements of s specific theodolite are shown. The gaps in May and June are related to the usage of various other DI instruments in preparation of an IAGA workshop. Checkout the Conrad Observatory Yearbook 2018 for complete data.

In a next step we want to fit an adopted baseline. It is obvious that a simple linear fit will do a good job here. Anyway, for demonstration purposes we will use two separate fits to describe the baseline.

Analysis Panel → Fit

In the fit dialog we add the first fit, a cubic spline, as shown in figure 6.2.3a. Then we add a second fit, a linear least-squares according to figure 6.2.3b.

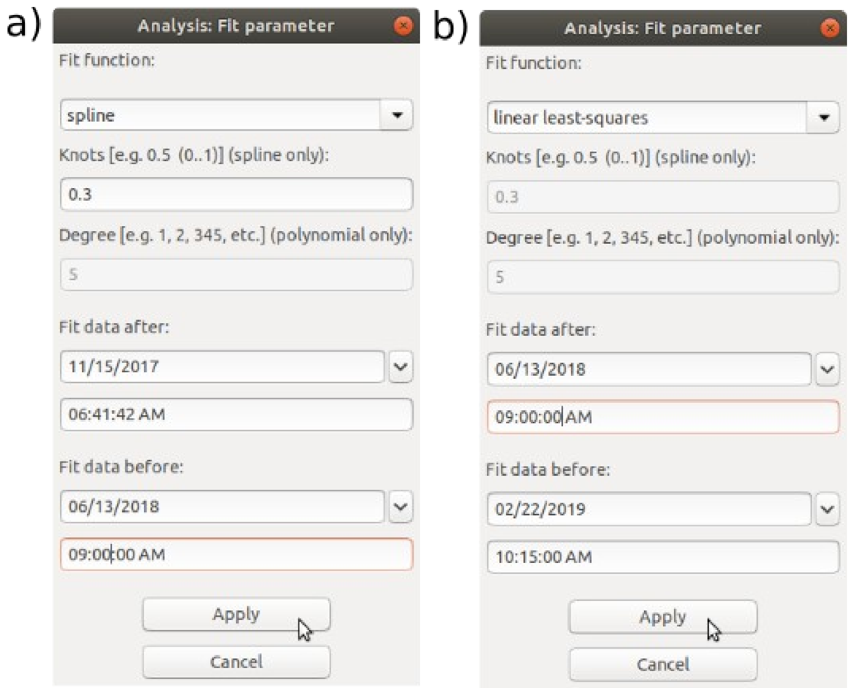


Figure 6.2.3: Fit dialog for first and second fit.

Both fits will be shown in the plot (Figure 6.2.4) and also be automatically recognized as fits to basevalue data in MagPy's memory.

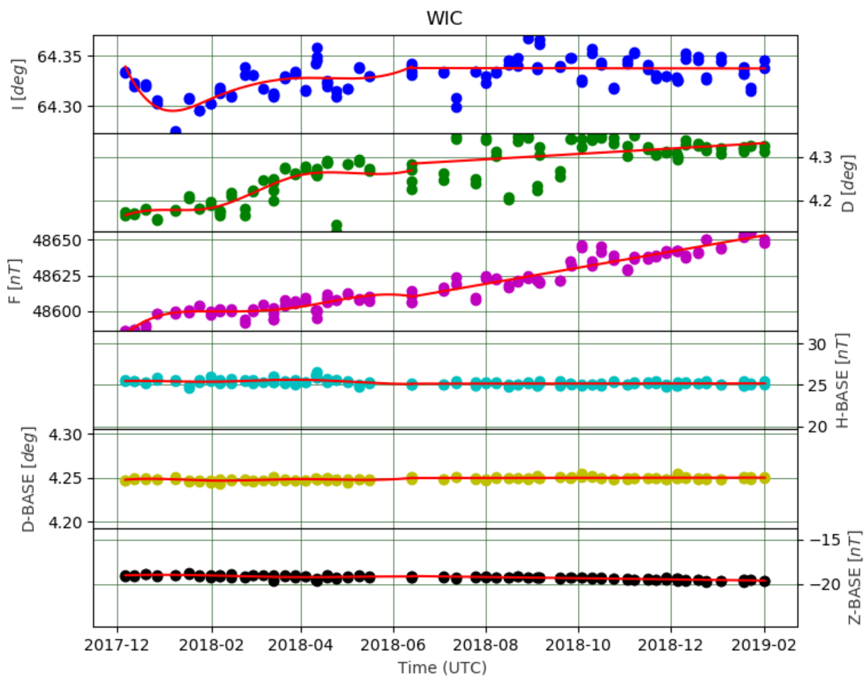


Figure 6.2.4: Adopted baseline.

If you are satisfied with that you can save a BLV file. Please note that the file name will automatically be set with the correct year.

Now we want to use the adopted baseline for baseline correction of the variation data. Therefore we load such variation data (example5.sec). After that you will find the button **Baseline** on the Analysis page enabled. Time to use it.

Analysis Panel → Baseline

A dialog will open as shown in figure 6.2.6. This dialog will let you choose from different baseline data sets and their fitting options within the memory from the upper drop down box. You can also redefine an adopted baseline by using **Change Fit**. For our baseline we just use the latest (final) input in the drop down menu, telling us that this will use a linear-least square based on example3.txt contents for our data set.

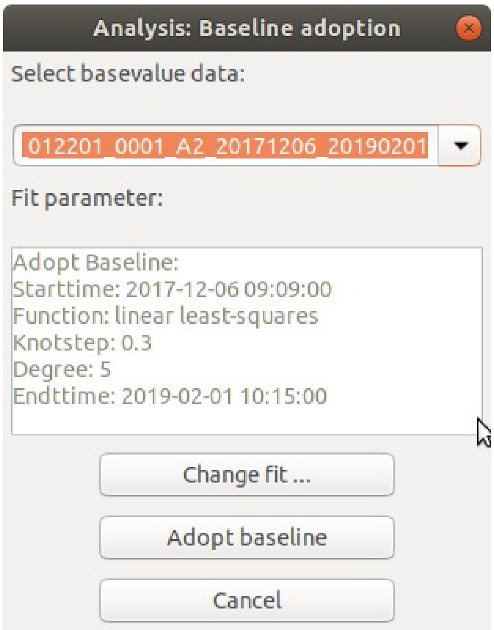


Figure 6.2.5: Baseline adoption dialog.

After using Adopt baseline an information dialog (Figure 6.2.6) will pop-up telling you that the baseline correction has been calculated. Its functional parameters are now contained in the meta information of the variation data set. Yet, the correction has not yet been performed. If you want you can now save your variation data set as an PYCDF archive which will store the functional parameters and, if available, also any flagging information currently connected with the data set. This way you can keep a single archive file which contains basically every data analysis step between raw data and definitive products.

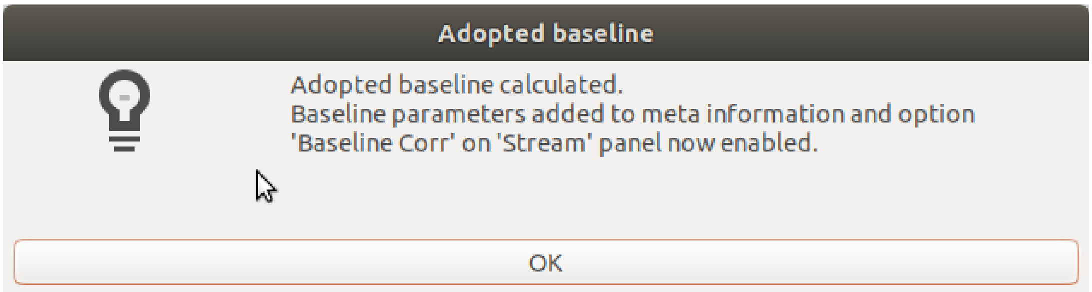


Figure 6.2.6: Information pop-up that an adopted baseline has been calculated and added to the variation meta data.

In order to apply baseline correction go to

Data Panel → Baseline corr

This will apply the functional parameter and you will now have a baseline corrected record for publication.

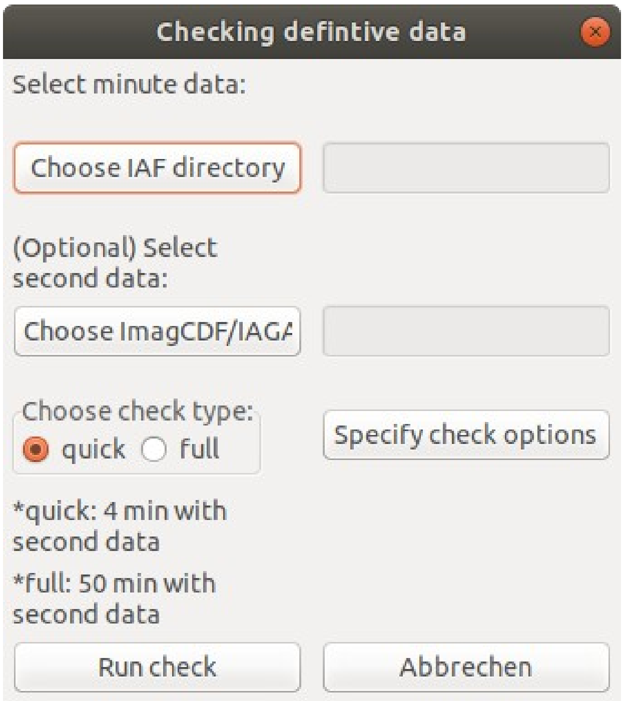
6.3 Definitive data check

XMagPy comes with some methods and functions to check the validity of definitive data in comparison to INTERMAGENT standards. Although these methods can also be used by official data checkers, the main purpose is to provide an easy testing environment for all users to check their data sets **before** submitting them to INTERMAGNET. For non-INTERMAGNET observatories, the testing routines are helpful for data consistency checks and provide the possibility to identify critical deviation values relative to the IM standard.
Please note, successfully passing the definitive check routines does not automatically mean, that your data is acceptable for IM. Further tests and final decision is always the duty of an official data checker. Using MagPy's definitive data check, however, will dramatically reduce the work load of the voluntary IM data checkers, as most typically occurring problems are tested, and therefore, all sides will benefit.

To access the data checking routine go to

MainMenu → Extra → Check definitive data ...

This will open the following dialog



You can choose source data too be checked and also the checking intensity. You can provide data in minute resolution and data in second resolution. If both data sets are provided please make sure that

they are from the same year. As source you define a directory which contains the INTERMAGNET requested data structure for definitive data submission (see below). If you just want to test either minute or second data, please leave the other source field empty.
Two check types are possible, quick and full. For the quick test a single randomly selected month will be tested. For full analysis, the validity tests will be performed for all months of one year.
You can also specify check options. By default all possible tests as shown below will be run. The checks will be performed one after the other and reports will be generated.

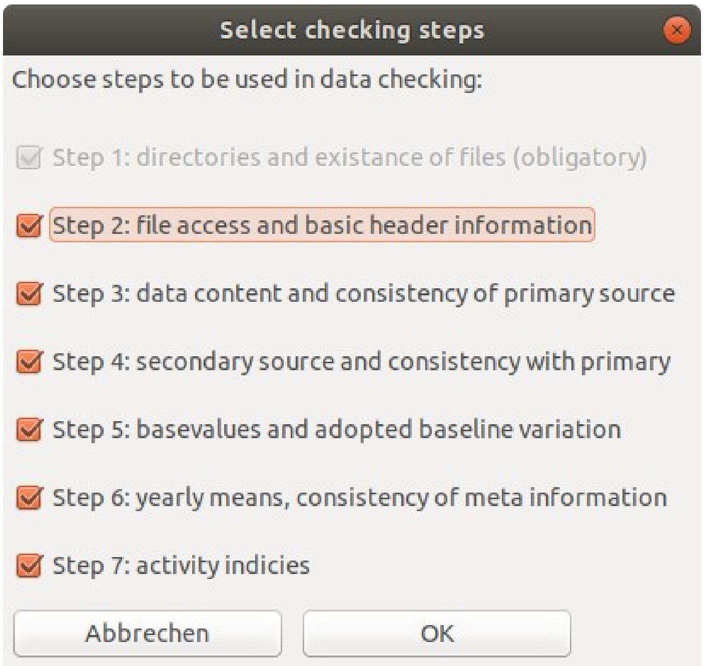
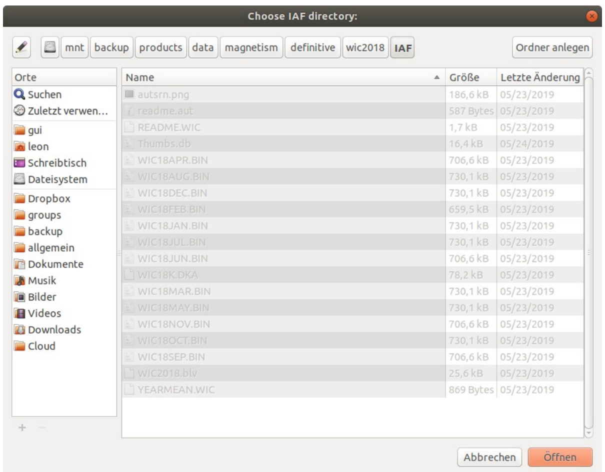
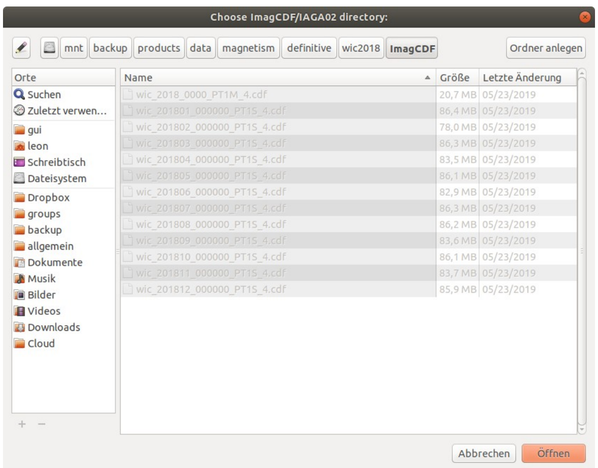


Figure 6.3.2: Checking steps

For this example we will use all checks and use a quick analysis. Valid directory structures for source directories will look similar to

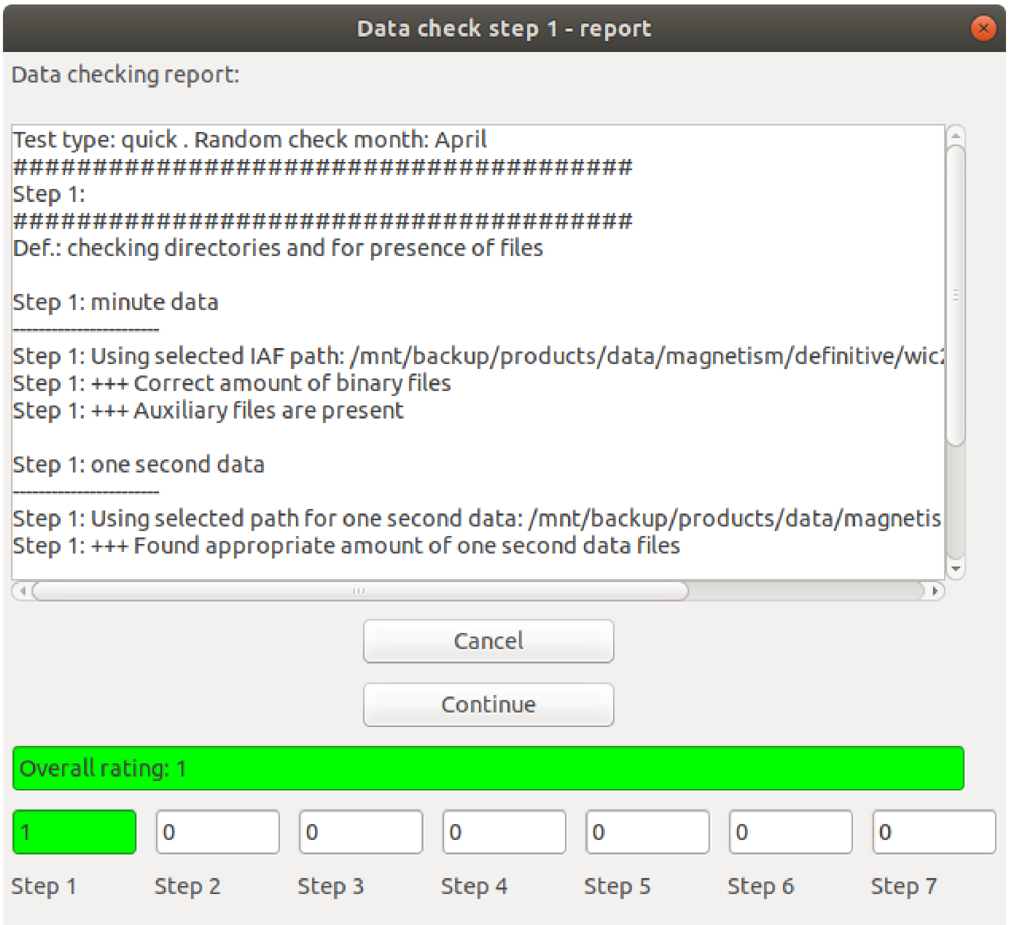




Afterwards you can select **Run check**

After each step, as defined above in figure 6.3.2, a report window will pop up, giving you a summary of the check results. The color and rating helps to guide your eye and eventually indicate

points which need your attention. Orange or red colors do not necessarily indicate that the data is not ready for IM submission.
The pop-up report looks like



The report window will contain some details on each performed check. It will also contain explanations if problems occurred. Warnings (orange) and errors (red) are found at the end of the report window. This report window will be extended with the results of every new checking step, so please scroll down for new information after you continued. At the end you can save the summary to a txt file. If you are checking one-second data then steps 2 and 4 are the ones which need longest. In summary the following checks are performed at each step:

Step 1: Test whether selected directories are existing and they contain expected data files (amount and types) for definitive data submission

Step 2: A random data file from each source directory is tried to be opened. If successful, then the basic required meta information of the file is analyzed.

Step 3: Content of minute resolution data is analyzed. This includes a coverage and sampling period check plus an analysis of delta F and average delta F. Furthermore, IAF data files contain hourly and daily filtered data. It will also be checked, whether these products have been filtered using INTERMAGNET recommended procedures. Step 3 will generate a plot of one minute data for the tested time series.

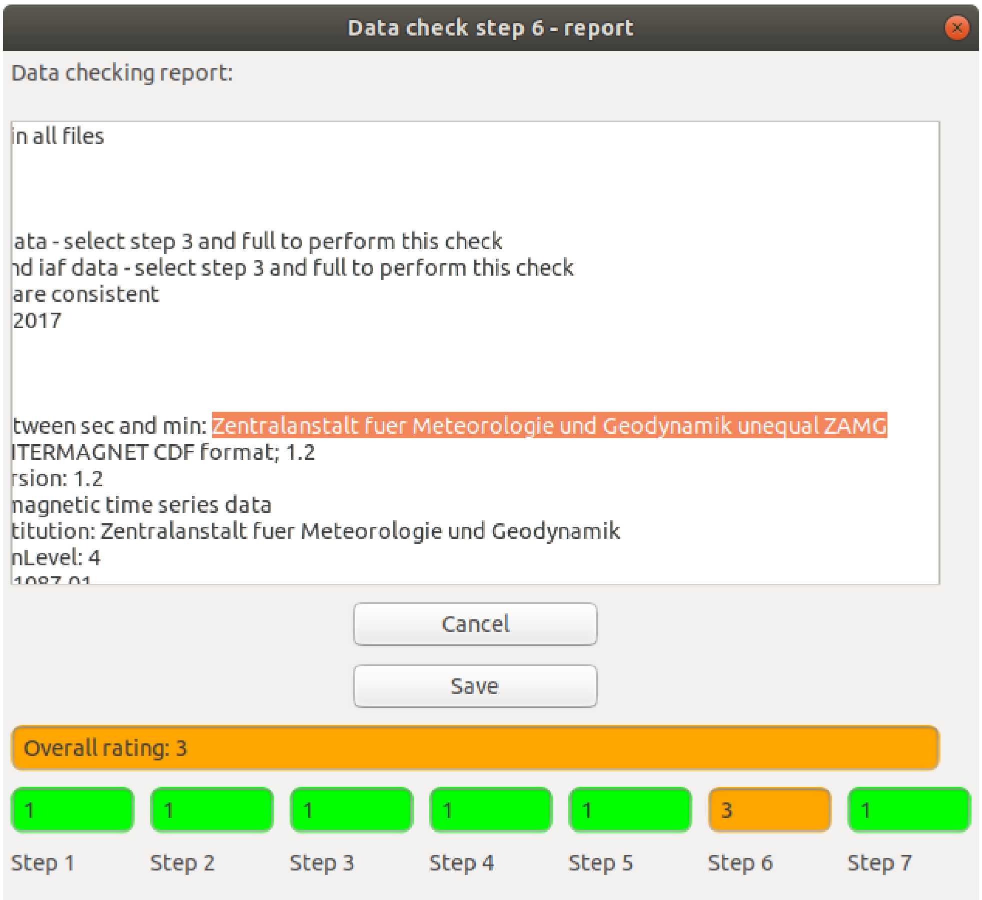
Step 4: One second resolution data is now analyzed. Firstly, coverage, sampling period and, eventually, delta values are checked. Then the time series is filtered to one minute resolution and, if one minute data is provided as source, the filtered one-second data set is directly compared to the provided one-minute data. If the differences exceed the numerical noise by a factor of 1.3, i.e. 0.13 nT, a warning is issued. The numerical noise is related to 0.1 nT rounding within the IAF data set. If a warning is generated then a plot with the differences will be created. Please note that the error report level is rather rigorous here, so you will easily get an orange warning. This particularly happens if your one minute data set is a composite with gaps filled from another instrument. If the mean difference value is close to zero, the great majority of data is well within an 0.1 nT variance, and the few maximum differences reported are not larger than 0.2-0.3 nT then there is no reason to panic. This is still excellent data.

Step 5: Now the basevalue (BLV) file will be opened and also shown in the plotting area. The adopted baseline will also be checked. The quality analysis starts with an average periodicity, i.e. how often have measurements been performed. If several measurements are performed each day, then the average intra-day deviation of measurements is analyzed. Afterwards the residuals between adopted baseline and basevalues are calculated. Finally the amplitude of the adopted baseline is determined. If any of these parameters exceeds 5 nT than a warning message is issued. The report contains all numerical values. When accessing the BLV plot later, you can also select the delta F component there, providing information on the F baseline.

Step 6: Meta information and header content from all provided data sets is now checked. Besides the data files (IAF, IAGA, ImagCDF) also the BLV and yearly mean files are investigated. All header information will be listed in the report. If differences of identically expected information is found, then a special notification will be created. Some notifications are uncritical as e.g. IAF only contains abbreviations of institute names whereas other format contain the full name. Sometime you might run into problems with this test, as MagPy tries to open and interpret a yearly mean file which actually is free form. You can test that before by just opening the yearmean file alone. You might want to disable step 6 if you have problems.

Step 7: An overview about k values as contained in the IAF data file is shown.

When all selected steps are finished you will end up with a full final report and the Data check window will look similar to



The orange warning indicator points to an uncritical difference between the ImagCDF one second data set and the one minute IAF data set, related to the institute name, which is only contained as abbreviation in IAF. You can save the report as soon as all selected steps are finished. You can use copy-and-paste anytime to extract information from the report window.

6.4 The art of web services

In this section a short overview on how to use webservice data is presented. A major aim of MagPy and its future development will be easy access and maximum support of web-based data management. Excellent examples for such features can be found on at the USGS. For this recipe we will get DI data from an USGS observatory and calculate basevalues with variation data provided by the webservice as well.

DI Panel → DI data → Select Webservice/Remote → usgs

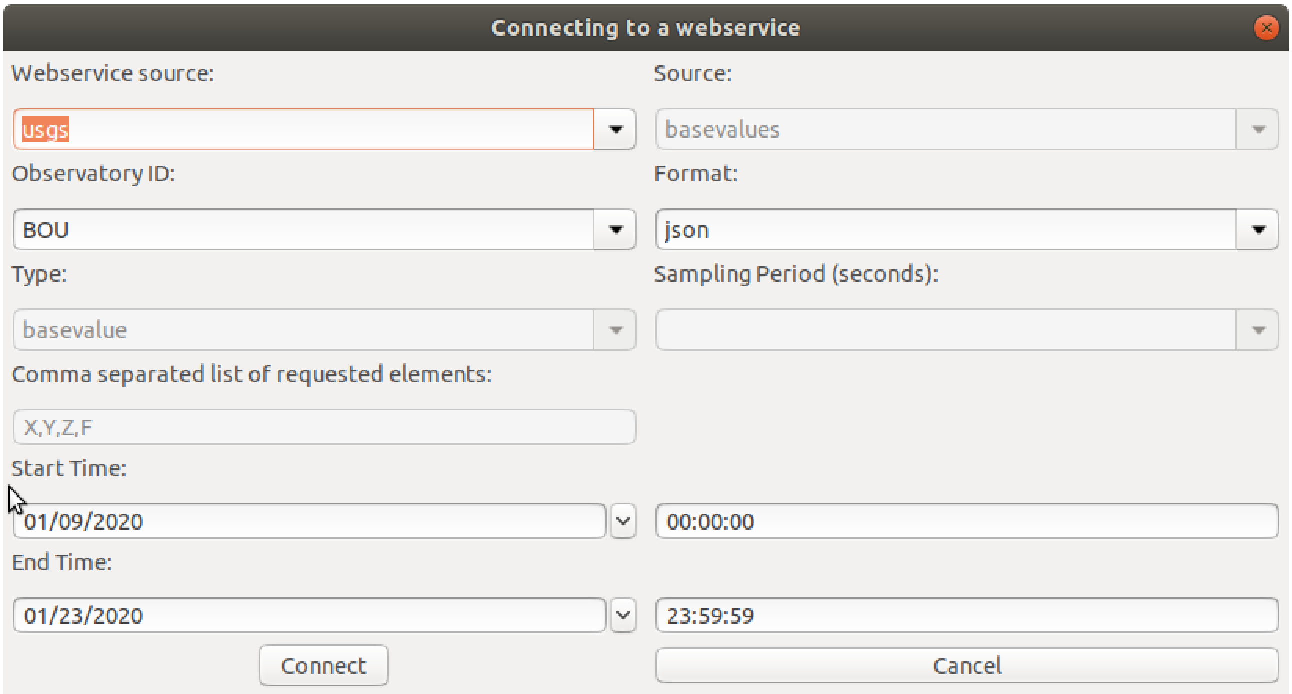


Figure 6.4.1: DI data selection from a webservice.

This will bring up the dialog shown in Figure 6.4.1. We select BOU and leave all other parameters unchanged. This we will load DI measurements for the last two weeks into the memory. The source field on the DI panel will read "webservice" and you will see the number of available records.

DI Panel → Vario/Scalar → *check webservice* → *select usgs*

The next step will open a webservice access to variation data. Please select the usgs webservice and the same observatory code as used above. For tests it is also recommended to use one-minute data only.

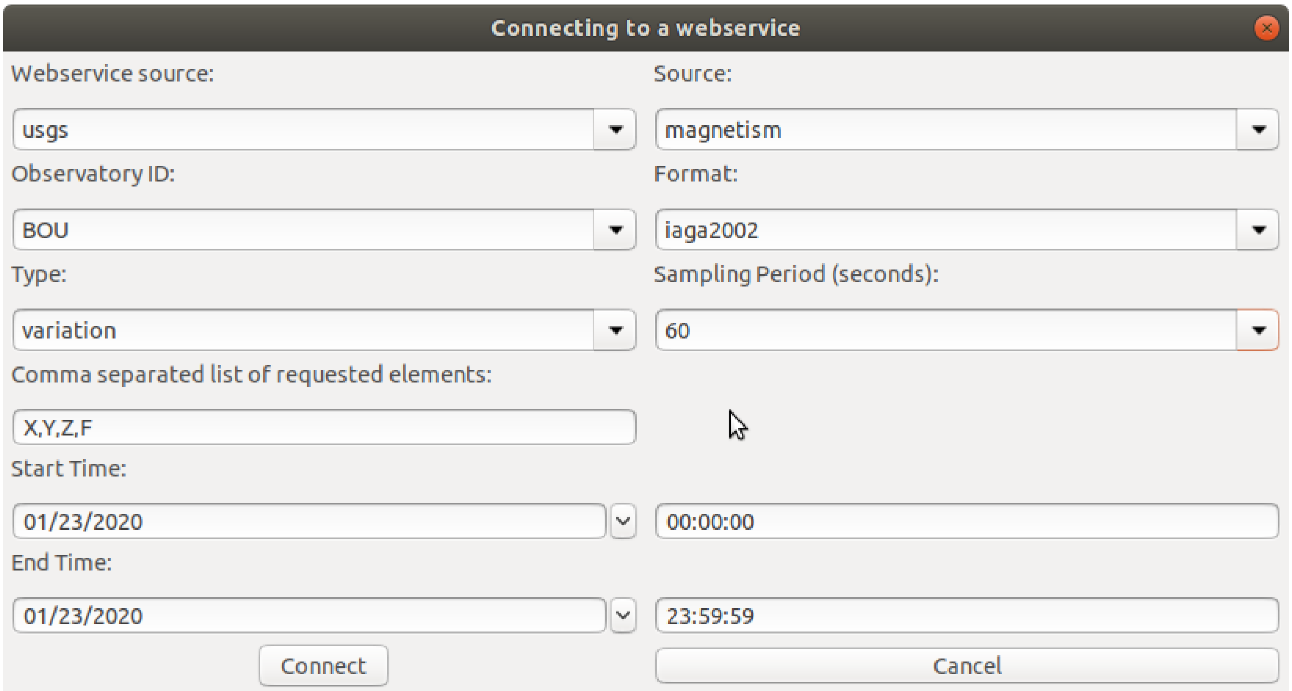


Figure 6.4.2: Selecting variometer and scalar source.

Modifying the time range is not necessary. The analysis process will access the necessary time intervals automatically. You have to repeat the same selection process for scalar data as well. When finished you can now run the **Analyze**. This will result in a plot similar as shown in Figure 6.4.3. You can also load longer time ranges, calculate adopted baselines and follow the instruction in 6.2.

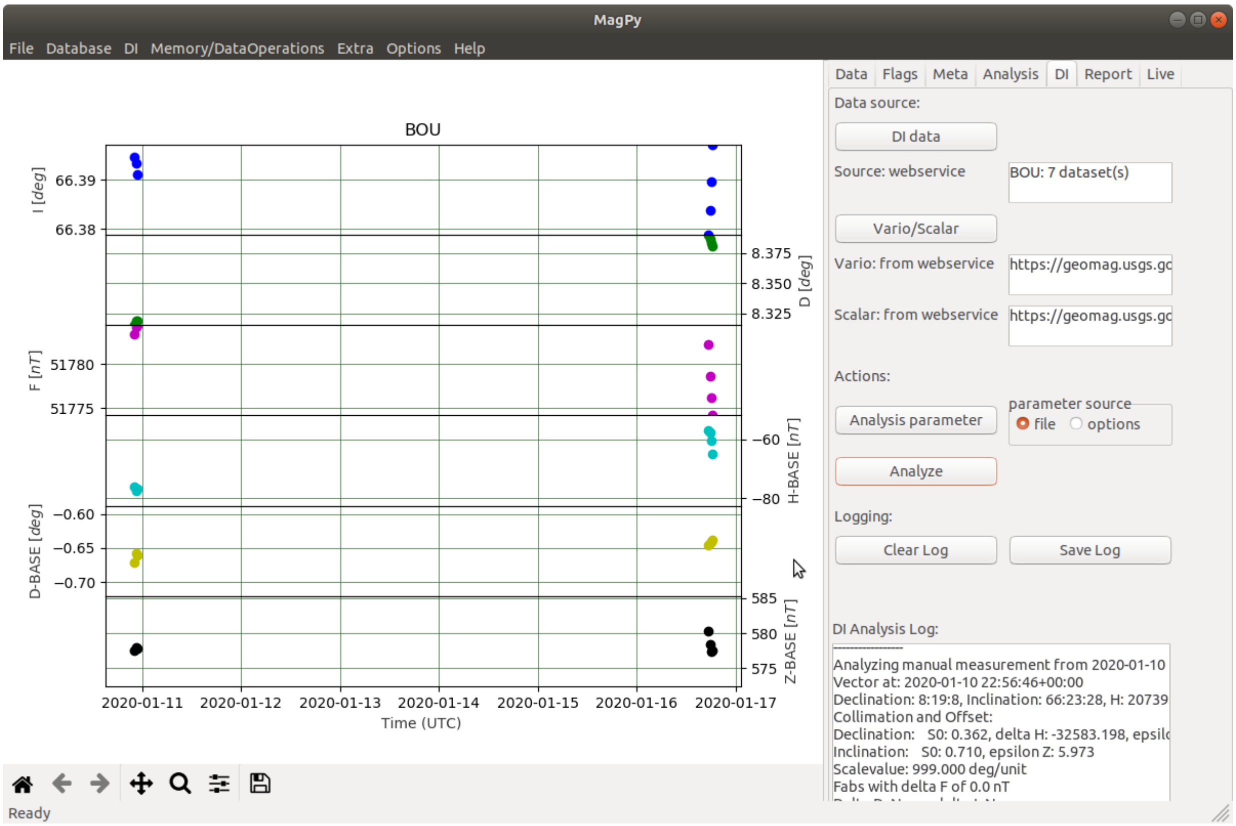


Figure 6.2.3: DI analysis with webservices.

7. FAQ

Where do I find the example files after installation?

Examples are always stored in the python path:
PYTHONPATH/lib/pythonx.x/site-packages/magpy/examples

You can, however, also download them directly from GitHub (https://github.com/geomagpy/magpy/tree/master/magpy/examples) and store them wherever you want.

Is it possible to add or modify webservice definitions?

At present, you cannot add or modify webservice definitions within the graphical user interface. It is however possible to add them directly into MagPy initialization file, which however requires some basic python knowledge, particularly python dictonaries. To access the initialization file you start a Python environment and issue the following commands:

```
import pickle
filename "/home/you/.magpyguiini" # locate this file on your computer
with open(filename, 'rb') as f:
  data = pickle.load(f)

ws = data.get(„webservices")

# modify ws dictionary to new_ws
data[„webservices"] = new_ws
with open(filename, 'wb') as f:
  pickle.dump(data,f,pickle.HIGHEST_PROTOCOL)
```

Webservices are stored within a python dictionary which can be modified and extended as described on all basic python introductions. Please only modify this information if you really now what you do.

Definitive data checking fails for step 6 – meta information check

During step 6, yearly mean files will be analyzed and their content compared to BLV and other data files. The yearlymean data format (IYVK), however, is not accurately defined wherefore a correct interpretation is not straightforward. Please try to load the file alone using **Open file**. If failing try to remove the comment section and reopen. Compare your file contents to the IM example on the webpage and check for additional blank lines, etc. If all this does not work, uncomment step 6 for definitive analysis, send us your yearlymean file and a short error report.

I have observed instabilities. Sometimes MagPy hangs up or just crashes?

Please always report such instabilities to the developers. This will help us to improve the software. If you are working with many large data files or perform analysis of more than a month of one second data with low-memory systems, you might observe such instabilities, because MagPy is memory hungry. In that case, splitting up your analysis and open new instances of MagPy for each analysis will help.

8. Appendix

8.1 Example files

example1.zip:

Example1 is a zipped file with data inIAGA-2002 format. It contains full vector information in one second resolution for one day (in magnetic coordinates HEZ). Fluxgate compensation values have been added to the respective component. Scalar data is not enclosed. This data set also recorded a thunder storm passing near the sensor location at about 15:00 UTC.

example2.cdf:

Example2 comes in a MagPy CDF (PYCDF) fromat which is based on NasaCDF. Example2 contains scalar data from a potassium sensor covering the same time range as example1. The sampling rate is one-second. Outliers due to a thunder storm have already been removed from example2.

example3.txt:

Example3 holds basevalues for more than one year obtained at the Conrad Observatory.

example4.cdf:

Example4 is a geomagnetic time series in ImagCDF format, which is based on NasaCDF. Example4 provides full vector information in one second resolution for one week, covering a weak geomagnetic storm. Additionally, temperatures of sensor and electronics are contained.

example5.sec:

Example5 is a IAGA-2002 one second data file. Example 5 contains raw variation data and an independent record of F. A F basis has already been applied so that F values are representative for measurements at the observatories main DI pier.

example6a.txt,example6b.txt:

Example6a and b are two DI measurements as obtained from the input sheet.

8.2 MagPy's DI fluxgate data format

MagPy supports DI fluxgate data as provided by the AutoDI flux (citation), the USGS webservice format and a simple ASCII format as given below. This ASCII format will also be generated when saving DI fluxgate data from the MagPy Input sheet.

8.3 Command Line Tools

MagPy comes with two command line tools: **mpconvert** is used to automatize and speed up file conversion, **mptest** provides a test suite for developers. After installation of MagPy these command line tool are available from the command prompt/terminal.

8.3.1 mpconvert

The command line tool **mpconvert** is a python script which is designed to quickly convert between different geomagnetic data formats.

```
$ mpconvert -h
```

will give you an overview about usage and options.

To convert one month of IAGA2002 one-second data (in folder /home/iagaseconds) into INTERMAGNETS ImagCDF format (to be found in /home/imagcdf) you could use the following command:

```
$ mpconvert -r "/home/iagaseconds/wic201701*" -f IMAGCDF -c month
            -w "/home/imagcdf"
```

In order to add some necessary meta information you would issue the following additional option:

```
$ mpconvert -r "/home/iagaseconds/wic201701*" -f IMAGCDF -c month
            -w "/home/imagcdf"
            -m "DataStandardLevel:Full,IAGACode:WIC,DataReferences:myref"
```

Please refer to help (-h) for more detailed information and other examples.

8.3.1 mptest

The command line tool **mptest** is designed for developers and can is used to perform reading and writing tests for specified data formats.