

## About Me

I'm a Marcos Vinicius Silva...

## Fast geomagnetic jerk detection

---

My Master project in geomagnetism.

A python tool to study the secular variation of the geomagnetic field and accelerate the geomagnetic jerk detection.

Actually works for every INTERMAGNET geomagnetic observatory (definitive and quasi-definitive IAGA-2002 data).

There are functions to reduce the external field contribution, calculate the secular variation, detect geomagnetic jerks...

The package is separated in modules called `data_processing_tools`, `utility_tools`, `support_functions` and `main_functions`.

All the functions have a docstring with the inputs explanation and an usage example.

## Documentation summary

---

- [Setting up MOSFiT package](#)
  - [Package installation](#)
  - [Downloading IAGA-2002 data](#)
- [Package modules import suggestion](#)
- [Requirements](#)
- [Modules functions](#)
  - [main\\_functions](#)
  - [data\\_processing\\_tools](#)
  - [utilities\\_tools](#)
- [main\\_functions usage](#)
- [data\\_processing\\_tools functions usage](#)

## Setting up MOSFiT package

The MOSFiT python package was built to work with INTERMAGNET minute mean data in the IAGA-2002 format, in order to analyse the SV and check INTERMAGNET Magnetic Observatory (IMO) data quality. The definitive and quasi-definitive data are mainly used because of higher quality and reliability, especially for SV studies. However, we can also apply MOSFiT to others types of IAGA-2002 data, i.e. provisional data).

### Package installation

MOSFiT is developed in the Python 3 language. The package can be compiled in the command window or in a "jupyter notebook enviroment".

You can download MOSFiT in: <https://github.com/marcosv9/Thesis-Marcos>. In this same link, there is a documentation of how to use the package functions, with some examples.

## Downloading IAGA-2002 data

In order to use MOSFiT, it is necessary to have the data stored in a local computer.

This data can be downloaded from the INTERMAGNET website (<https://www.intermagnet.org/>), directly from the INTERMAGNET ftp server (<ftp://ftp.seismo.nrcan.gc.ca/intermagnet/>) or by using the MOSFiT function called "download INTERMAGNET file" (by choosing datatype, year, month and the list of observatories).

MOSFiT will only read filenames in the same format of INTERMAG-NET IAGA-2002 2. After the data is downloaded, the user may organize all files from different observatories in a single or multiple folders.

Most MOSFiT functions require an input called 'station'. It is the 3 letter IAGA code of the INTERMAGNET observatory. In MOSFiT there is a database with all INTERMAGNET observatories registered (IAGA code, latitude, longitude and altitude, this informations are used in many data processing functions). If you want to use MOSFiT with an observatory or location that are not registered in the database, there is a MOSFiT class called IMO that includes any location into the database. See utilities\_tools section for an explanation about how to include any location.

## Package modules import suggestion

To use the developed funtions, is necessary to import them. I suggest to use the following statements to import the modules.

```
import data_processing_tools as dpt
import utilities_tools as utt
import main_functions as mvs
import support_functions as spf
```

## Requirements

- scipy
- chaosmagpy
- numpy
- pandas
- matplotlib
- glob2
- pathlib2
- pwlf
- chaosmagpy
- scikit-learn

## Modules functions

Here are the principal functions of each module and a quick description.

Functions listed here are fundamental in the geomagnetic data processing.

There are others functions into the modules, including the support\_functions module that was not mentioned. Most of them are internally used by the package.

## main\_functions

```
import main_functions as mvs
```

Function	Description
<a href="#">mvs.load_intermagnet_files</a> (station, starttime, endtime, files_path)	read and merge IAGA-2002 file format into a pandas DataFrame
<a href="#">mvs.sv_obs</a> (station, starttime, endtime, plot_chaos, files_path)	interactive data processing workflow until geomagnetic jerk detection
<a href="#">mvs.plot_sample</a> (station, dataframe, save_plots ...)	automatically plot hourly, daily, monthly and annual means
<a href="#">mvs.plot_sv</a> (station, starttime, endtime, df_station, df_chaos, ...)	automatically plot SV with options to correct the data and plot CHAOS prediction
<a href="#">mvs.plot_tdep_map</a> (time, deriv, plot_changes, station)	plot a global map of the SV or SA as well as their changes (from CHAOS prediction)

## data\_processing\_tools

```
import data_processing_tools as dpt
```

Function	Description
<a href="#">dpt.resample_obs_data</a> (dataframe, sample, apply_percentage)	Resample obs minute or hourly means into hourly, daily, monthly or annual means
<a href="#">dpt.hampel_filter_denoising</a> (dataframe, window_size, n_sigmas, ...)	Denoising filter based on median absolute deviation
<a href="#">dpt.night_time_selection</a> (station, dataframe)	Select the nighttime period from geomagnetic data. Default from 22pm to 2am LT.
<a href="#">dpt.keep_quiet_days</a> (dataframe)	Select only top 10 quiet days from each month
<a href="#">dpt.remove_disturbed_days</a> (dataframe)	Remove top 5 disturbed days from each month
<a href="#">dpt.kp_index_selection</a> (dataframe, kp)	Select only periods with Kp values <= the defined limit. Default is 2

Function	Description
<a href="#">dpt.calculate_sv</a> (dataframe, method, source, ...)	Calculate SV from geomagnetic data using monthly or annual means (input must be output from load_intermagnet_files or chaos_model_prediction). Default is monthly means
<a href="#">dpt.chaos_model_prediction</a> (station, starttime, endtime, n_core, ...)	Predict different sources of the geomagnetic from CHAOS-7 model (core, crust, magnetospheric (GSM + SM)).
<a href="#">dpt.external_field_correction_chaos_model</a> (station, starttime, endtime, ...)	Correct magnetospheric field from geomagnetic data
<a href="#">dpt.jerk_detection_window</a> (station, window_start, window_end, ...)	Automatically adjust two straight line segments in the SV for an user specified time window

## utilities\_tools

```
import utilities_tools as utt
```

Function or Class	Description
<a href="#">utt.download_intermagnet_data()</a>	Download observatory quasi-definitive or definitive data from INTERMAGNET fpt server and save in the computer
<a href="#">utt.hdz_to_xyz_conversion</a> (station, dataframe, files_path)	Check the existence of reported HDZ components and convert to XYZ components
<a href="#">utt.IMO</a> (self, station, latitude, longitude, ...)	Class representing IMO. Can be used to check IMO informations on MOSFiT database (IMO existence, latitude, longitude, altitude) as well as add a new IMO or delete

## main\_functions usage

The main\_functions package module consist of functions to load (read) and visualize IAGA-2002 data.

It also has an interactive function [sv\\_obs](#) that includes the most important data processing options.

### load\_intermagnet\_files

This function is the most important, since it reads any geomagnetic data following the IAGA-2002 format.

The output is a pandas DataFrame indexed by time and the columns are the X, Y and Z geomagnetic components.

Its output is used as input is most of the data processing functions.

```
load_intermagnet_files(station = 'XXX', starttime = 'yyyy-mm-dd', endtime = 'yyyy-mm-dd', files_path = 'path//to//files')
```

The returned dataframe can be manipulated by the user or processed with the others functions of the package.

You can set a specific directory or just use the default (automatically created when the files are downloaded using `download_data_INTERMAGNET` function).

## sv\_obs

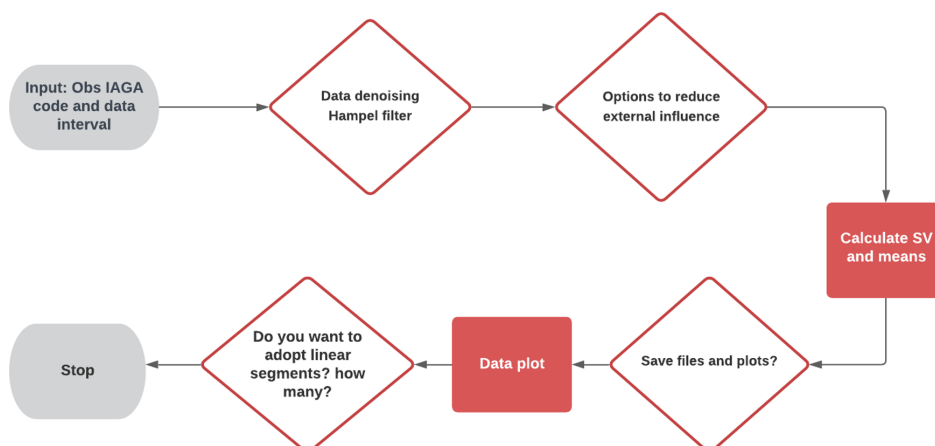
`sv_obs` is a function that includes the most important data processing options.

The processing according to the figure is already implemented in a dedicated function.

However the user can combine any of the processing steps in any possible order or combination

It allows the user to process the geomagnetic data in a interactive workflow,

using most of the available `data_processing` functions.



```
sv_obs(station = 'VSS', starttime = '2000-01-01', endtime = '2021-06-30', files_path = 'path//to//files', plot_chaos = True)
```

## plot\_samples

Automatically plot hourly, daily, monthly and annual means for X, Y and Z

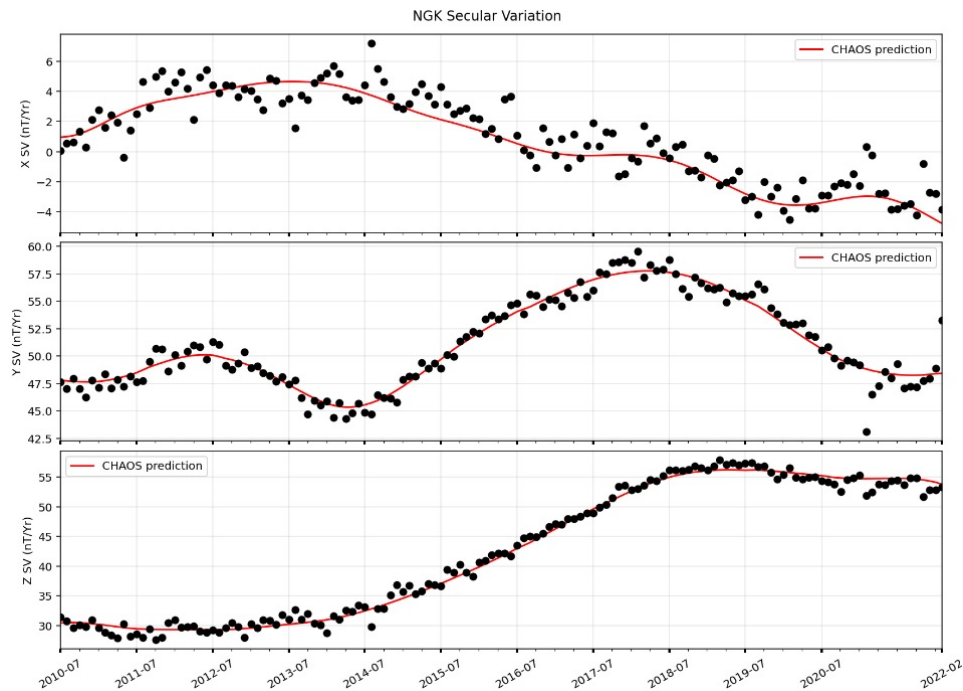
```
import main_functions as mvs
mvs.plot_samples(station = 'VSS', dataframe: df_name, save_plots = False, plot_data_type = True, apply_percentage = False )
```

## plot\_sv

Function to automatically plot the SV for an observatory

```
import main_functions as mvs
mvs.plot_sv(station = 'NGK', starttime = None, endtime = None, files_path = None,
df_station = df_name, df_chaos = None, apply_percentage = False, plot_chaos =
True, chaos_correction = True, save_plot = False, convert_hdz_to_xyz = False)
```

Example of SV from NGK automatically created using the function. The CHAOS model internal field predictions is also an option as well as correct the magnetospheric field.

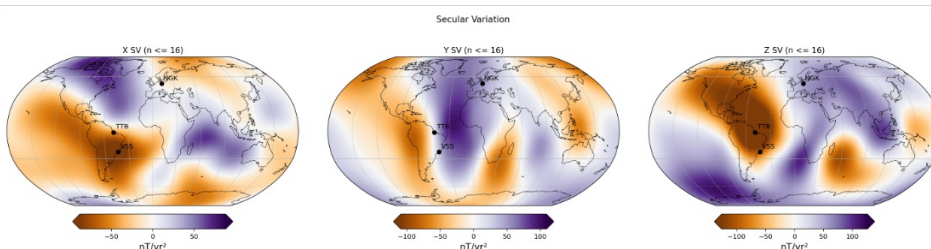


## plot\_tdep\_map

Plot a SV or SA world map for the CHAOS model predictions. It is possible to plot SV and SA changes as well as include 'stations' in the map (see figure)

```
import main_functions as mvs
mvs.plot_tdep_map('2022-07-01', deriv = 1, plot_changes=False, station =
['VSS', 'NGK', 'TTB'])
```

Example of output from plot\_tdep\_map



# data\_processing\_tools functions usage

Here I explain the principal function of the data\_processing\_tools module.

As the name says such functions are responsible for the data processing.

Most of them are methods to reduce the external field contribution from the observatory data. In order to investigate the SV.

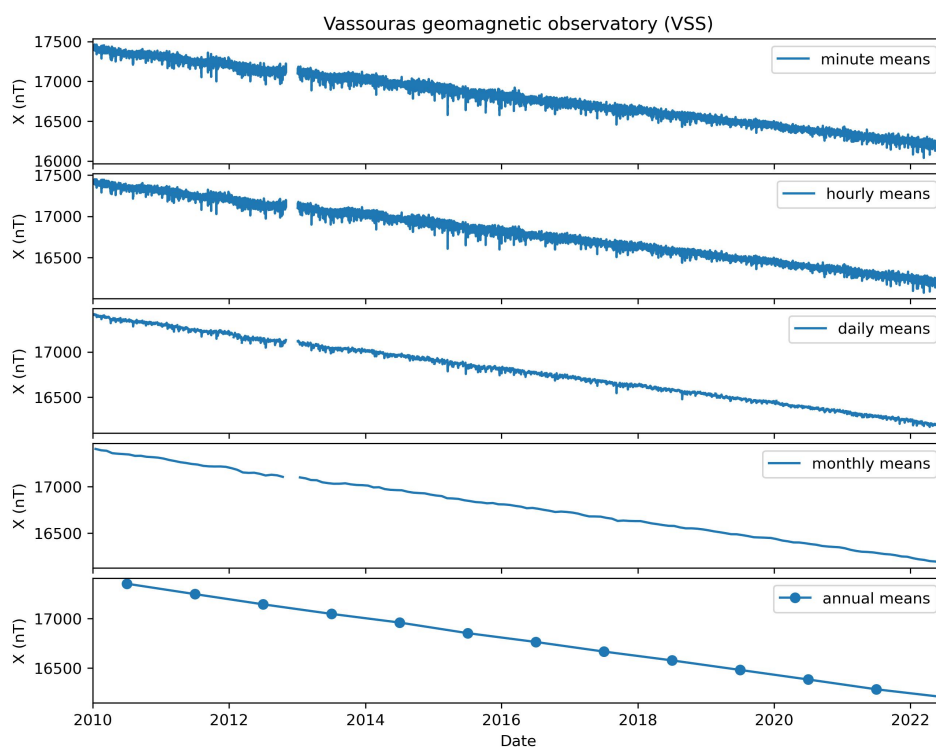
## resample\_obs\_data

This function allows the user to resample geomagnetic observatory

data into different samples (hourly, daily, monthly and annual).

```
import data_processing_tools as dpt
dpt.resample_obs_data(dataframe = df_name, sample = 'H', apply_percentage = True)
```

Example of different data samples calculated using MOSFiT.



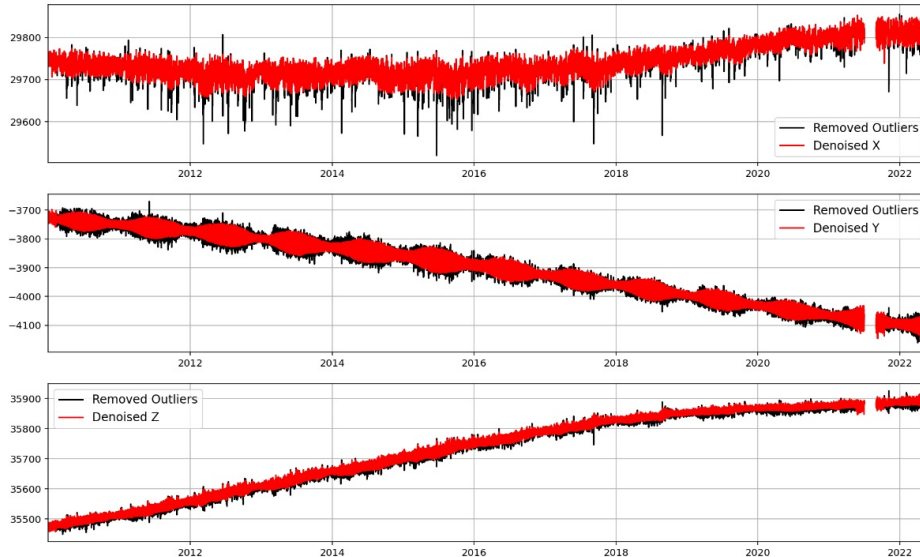
## hampel\_filter\_denoising

This function to denoise geomagnetic data based on a median absolute deviation filter

In order to reduce computational coast the function automatically resample the minute mean data (default from IAGA-2002 data and output from load\_intermagnet\_files) into hourly mean values

```
import data_processing_tools as dpt
dpt.hampel_filter_denoising(dataframe = df_name, window_size = 200, n_sigmas = 3,
apply_percentage = True, plot_figure = True)
```

Example of denoised hourly mean data.

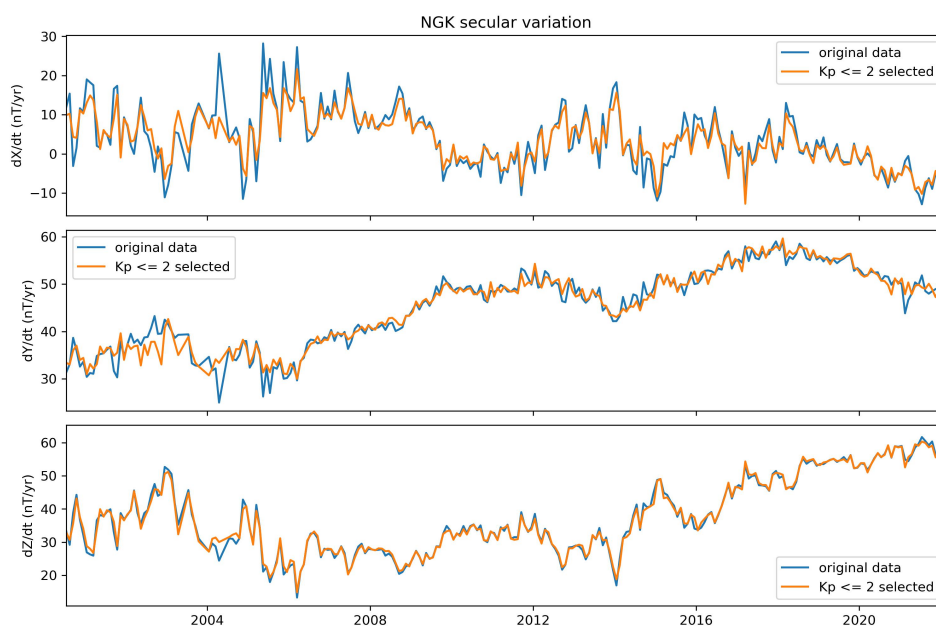


## kp\_index\_correction

The function removes periods with Kp index values above user input limit from the geomagnetic components

Find the Kp index on <https://kp.gfz-potsdam.de/en/>

```
import data_processing_tools as dpt
dpt.kp_index_correction(dataframe = df_name, kp = 2)
```



## keep\_quiet\_days

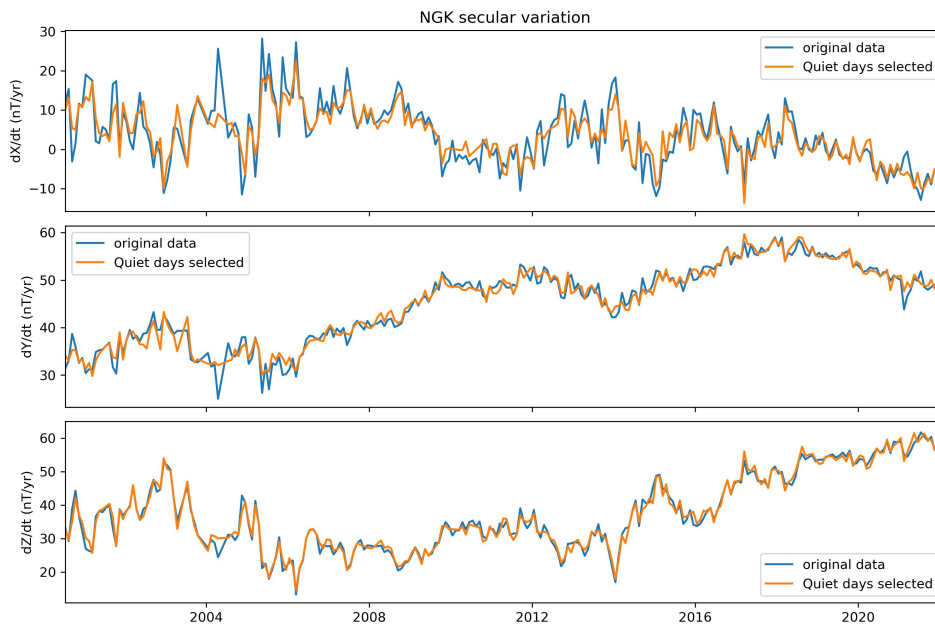


The function select only the top 10 international quiet days from each month

Find the list of quiet days for each month on <https://kp.gfz-potsdam.de/en/>

```
import data_processing_tools as dpt
dpt.keep_quiet_days(dataframe = df_name)
```

Example of SV calculate using normal data and selecting quiet days for each monthly.



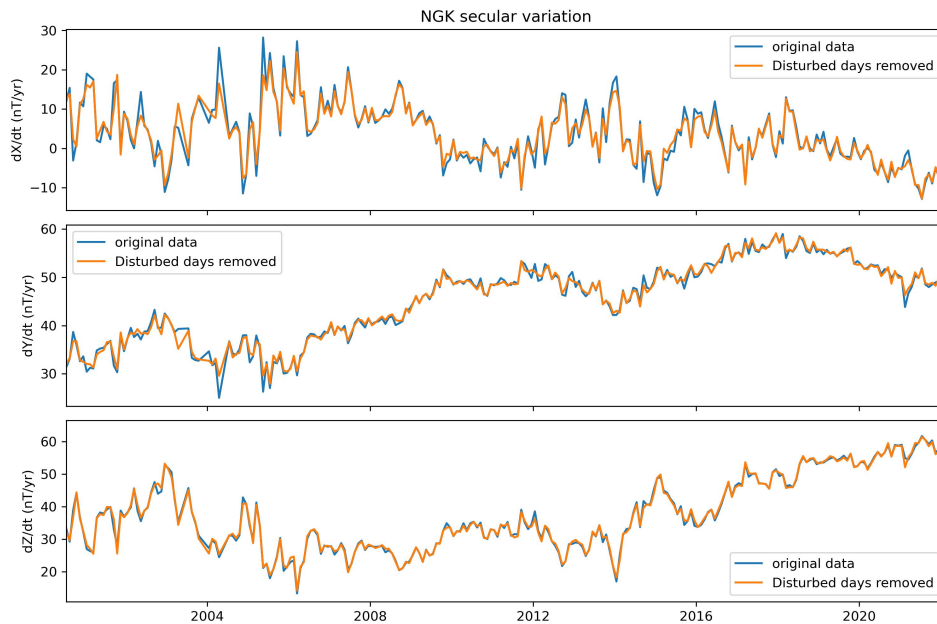
## remove\_disturbed\_days

The function remove the top 5 international disturbed days from each month

Find the list of disturbed days for each month on <https://kp.gfz-potsdam.de/en/>

```
import data_processing_tools as dpt
dpt.remove_disturbed_days(dataframe = df_name)
```

Example of SV calculate using normal data and removing the top 5 disturbed days from each month.



## night\_time\_selection

The function select the nighttime period from the geomagnetic data (default from 22pm to 2 am LT)

```
import data_processing_tools as dpt
dpt.night_time_selection(station = 'XXX', dataframe = df_name)
```

Example of SV calculate using normal data and selecting only nighttime period.

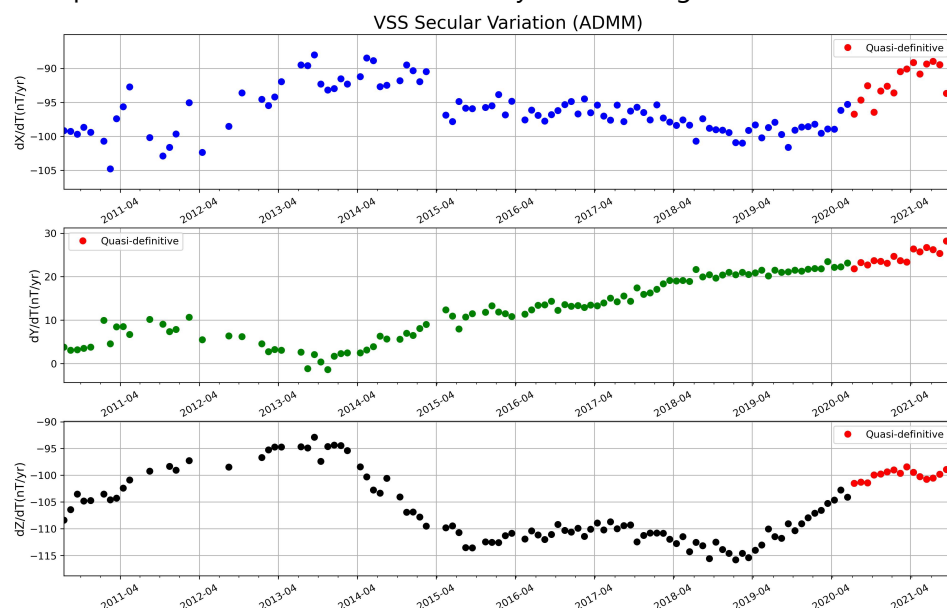


## calculate\_sv

Calculate geomagnetic secular variation using monthly means or annual means

```
import data_processing_tools as dpt
dpt.calculate_sv(dataframe = df_name, method = 'ADMM')
```

Example of SV calculate from VSS monthly means using MOSFiT.



## chaos\_model\_prediction

Predict core fiel, crustal field and magnetospheric field (GSM and SM) from CHAOS-7 model predictions in a hourly rate

find the model realease on <http://www.spacecenter.dk/files/magnetic-models/CHAOS-7/>

References Finlay, C.C., Kloss, C., Olsen, N., Hammer, M. Toeffner-Clausen, L., Grayver, A and Kuvshinov, A. (2020), The CHAOS-7 geomagnetic field model and observed changes in the South Atlantic Anomaly, Earth Planets and Space 72, doi:10.1186/s40623-020-01252-9 [pdf]

Finlay, C.C., Kloss, C., Olsen, N., Hammer, M. and Toeffner-Clausen, L., (2019) DTU Candidate models for IGRF-13. Technical Note submitted to IGRF-13 task force, 1st October 2019 [pdf]

Example of how to use MOSFiT chaos\_model\_prediction. The station (3 letter IAGA code) must be in the MOSFiT imos database. All INTERMAGNET observatories are included in the database automatically. If you are interest in predict the field for other observatory or location, use the 'IMO' MOSFiT class to add the location in the database. See utilities\_tools section for an explanation about how to include the location.

```
import data_processing_tools as dpt
dpt.chaos_model_prediction(station = 'XXX', starttime = 'yyyy-mm-dd', endtime =
'yyyy-mm-dd', n_core = 20, n_crust = 110, n_gsm = 2, n_sm = 2)
```

## external\_field\_correction\_chaos\_model

Subtract the magnetospheric field (GSM and SM) from CHAOS-7 model predictions from the observatory data

find the model realease on <http://www.spacecenter.dk/files/magnetic-models/CHAOS-7/>

References Finlay, C.C., Kloss, C., Olsen, N., Hammer, M. Toeffner-Clausen, L., Grayver, A and Kuvshinov, A. (2020), The CHAOS-7 geomagnetic field model and observed changes in the South Atlantic Anomaly, Earth Planets and Space 72, doi:10.1186/s40623-020-01252-9 [pdf]

Finlay, C.C., Kloss, C., Olsen, N., Hammer, M. and Toeffner-Clausen, L., (2019) DTU Candidate models for IGRF-13. Technical Note submitted to IGRF-13 task force, 1st October 2019 [pdf]

Example of how to use MOSFiT external\_field\_correction\_chaos\_model.

```
import data_processing_tools as dpt
dpt.external_field_correction_chaos_model(station = 'XXX', starttime = 'yyyy-mm-dd', endtime = 'yyyy-mm-dd', df_station = None, df_chaos = None, n_core = 20, n_crust = 110, n_gsm = 2, n_sm = 2)
```

Example of SV calculate from VSS monthly means using MOSFiT magnetospheric field correction from CHAOS predictions.

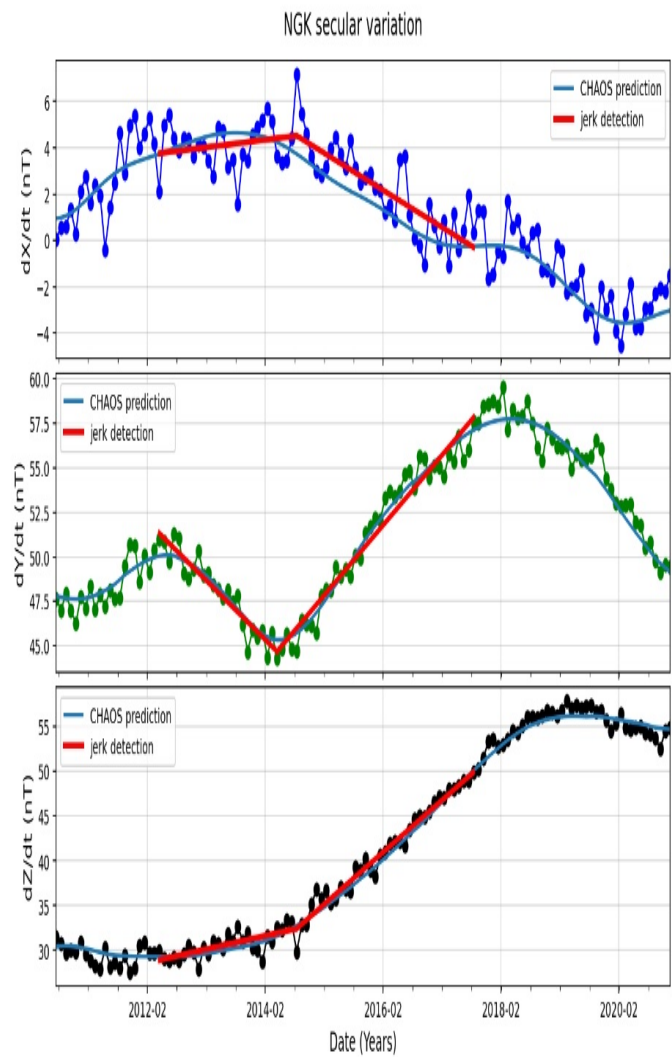


## jerk\_detection\_window

Automatically fits two straight line segments for an user selected window. Determine the geomagnetic jerk occurrence time ( $t_0$ ), amplitude (A) and coefficient of determination ( $R^2$ ).

The function uses the occurrence time as input in the [plot\\_tdep\\_map](#) to plot the SA changes for the jerk detection.

```
import data_processing_tools as dpt
dpt.jerk_detection_window(station = 'NGK', window_start = '2012-04', window_end = '2017-08', starttime = '2010-01-01', endtime = '2021-06-30', df_station = None, df_chaos = None, files_path = None, plot_detection = True, chaos_correction = True, plot_chaos_prediction = False, convert_hdz_to_xyz = False, save_plots = False)
```



NGK Jerk statistics for the X component.

Jerk occurrence time -t0-: 2014.62  
Jerk amplitude: -1.93  
R^2: 0.68

\*\*\*\*\*

NGK Jerk statistics for the Y component.

Jerk occurrence time -t0-: 2014.28  
Jerk amplitude: 7.26  
R^2: 0.95

\*\*\*\*\*

NGK Jerk statistics for the Z component.

Jerk occurrence time -t0-: 2014.62  
Jerk amplitude: 4.32  
R^2: 0.98

\*\*\*\*\*

SA changes for the occurrence time

