New Tools for Terrain Gravimetry
NEWTON-g
Project number: 801221

Deliverable 4.2

Data mining tools

Lead beneficiary:	Helmholtz-Zentrum Potsdam. Deutsches GeoForschungsZentrum (GFZ)
Dissemination level:	Public
Version:	Final

NEWTON-g has received funding from the EC's Horizon 2020 programme, under the FETOPEN-2016/2017 call (Grant Agreement No 801221)

Document Information

Grant Agreement Number	801221
Acronym	NEWTON-g
Start date of the project	1 June 2018
Project duration (months)	48
Deliverable number	D4.2
Deliverable Title	Data mining tools
Due date of deliverable	31 July 2020
Actual submission date	7 September 2020
Lead Beneficiary	GFZ
Type	R: Other
Dissemination level	PU - Public
Work Package	WP4 – Data analysis

Version	Date	Author	Comments
v.0	17/07/2020	M. Koymans, F. Cannavò	Creation
v.1	01/09/2020	F. Cannavò, D. Carbone	Revision
v.2	03/09/2020	E. De Zeeuw-Van Dalfsen, M. Koymans, F cannavò	Revision
v.4	04/09/2020	E. Rivalta, M. Nikkhoo	Revision
Final	07/09/2020	M. Koymans, D. Carbone	Validation

Table of contents

Introduction

NEWTON-g aims to overcome the limitations imposed by current instrumentation for gravity measurements, through the development of a field-compatible measuring system, the “gravity imager”, able to real-time monitor the evolution of subsurface mass changes. This system includes an array of low-costs MEMS-based relative gravimeters, anchored on an absolute quantum gravimeter. It will provide imaging of gravity changes, associated with variations in subsurface fluid properties, with unparalleled spatio-temporal resolution. NEWTON-g also aims to field-test the gravity imager at Mt. Etna volcano (Italy), where, due to persistent volcanic activity, measurable volcano-related gravity changes often develop, over different time scales.
The present document describes the “data mining” software tools that have been developed to automatically detect meaningful anomalies in the dataset produced by the gravity imager. Here “data mining” is used to indicate the analysis techniques that are applied to extract, from the gravity data acquired through an array of geographically distributed instruments, features that are relevant to the ongoing geophysical (volcanic) processes.
The availability of effective data mining tools may facilitate the use of gravity time series from an extended array of stations for volcano monitoring and hazard assessment purposes. Indeed, they make it possible to automatically detect anomalous changes that develop before volcanic activity shifts from quiescence to a phase of unrest.
The work described in the present document was not limited to the development of the data mining tools. We also designed software tools aimed at generating synthetic time series of gravity changes, starting from the assumption of controlled mass changes developing within simple-shaped volumes (e.g., spheres, cylinders, prisms). Assuming realistic figures for (i) the position of the mass source and (ii) the time evolution of the mass change, it is possible to generate the expected time series of synthetic gravity changes at each observation point of the gravity imager at the summit of Mt. Etna (see D3.1). This provides a controlled and realistic framework to test the performance of the data mining tools.
Real data from an extended array of gravimeters will not be available until the installation of the gravity imager at Mt. Etna is completed. This installation was scheduled to be completed within the summer of 2020 (see Annex 1 – Part B of NEWTON-g’s Grant Agreement). Nevertheless, due to the impact of the COVID-19 pandemic on the production of the MEMS gravimeters (see D2.5), it is likely that only one MEMS gravimeter will be available for installation on Etna by the end of autumn 2020, while the installation of the remaining MEMS gravimeters will occur in 2021. Under this scenario, the synthetic time series play an important role, as they allow us to assess the sensitivity and practicality of the developed data mining tools. When data from the NEWTON-g gravity imager becomes available, it will be easier to apply the already developed data mining techniques, thus extracting the most information from the available gravity data.

1. Data Mining Algorithms

In the frame of the basic anomaly detection problem applied to time series, the goal is to find outlier data points, relative to some standard or "usual" signal. The problem can be addressed in two ways: supervised and unsupervised. While the first approach needs some labeled data, the second does not, and just raw data are needed. Since we do not have much a-priori information on the characteristics of the anomalies that may affect data collected through the new MEMS gravimeters, we focus on the second approach.
Generally, unsupervised anomaly detection methods work through the following steps:

1. a generalized simplified version of the data series is built;
2. everything which diverges from this simplified model is considered an anomaly.

This approach requires defining a model that describes the data as closely as possible. In the case of time series, simpler and more complex techniques can be employed to define the above model, from moving averages to techniques such as the Auto Regressive Integrated Moving Average (ARIMA), regression trees, neural networks, etc.
In the following section, we present the two algorithms at the core of NEWTON-g's data mining tools. The CUSUM (Cumulative Sum Control Chart) algorithm is usually implemented by monitoring infrastructures, to perform real-time anomaly detection throughout time series. The EOF (Empirical Orthogonal Function) algorithm is also considered to facilitate spatio-temporal analysis of variance in the time series from the distributed network of gravimeters.

1.1 CUSUM Change Detection

The CUSUM change detection algorithm is a technique used for anomaly detection in time-variant data that can also be applied in real-time. The approach is commonly applied to detect out-of-control behavior in industrial processes that begin to deviate from safe operational values. The algorithm uses a cumulative sum of deviations from a target mean value, and allows to detect small shifts in the mean value of a process.
For a process that is considered "in control", the sum of deviations from the target value will vary randomly around zero. When the mean of the process shifts, an upward or downward trend will rapidly develop in this sum of deviations. An alert can be issued once the drifting trend crosses an empirically selected threshold.
The procedure can be summarized as follows: a group of data values of size N is normalized by subtracting the in-control process mean and dividing by the in-control standard deviation of the group, as shown in Eq. (1). The in-control parameters ($\bar{x}, \sigma_x$) can be derived from a data set acquired when the operational process was known to be regular. Alternatively, the parameters can be based on the data set itself and continuously adjusted with incoming data. The latter approach is required when the in-control mean and standard deviation are unknown.

$$z_N = \frac{x_n - \bar{x}}{\sigma_x} \tag{1}$$

Even though the CUSUM algorithm generally uses the mean as the reference model, more complex models can be used to determine the expected target value, e.g., linear or nonlinear regression, exponential smoothing, dynamic linear models, etc. To adapt the model prediction to faster or slower process changes, the model can be estimated using data within a sliding window, whose size is tuned to match the characteristic dynamics of the anomaly to detect.
The cumulative sum of deviations is based on the recursive relationships:

$$S_{H_{N+1}} = \max(0, S_{H_N} + z_N - k)$$
$$S_{L_{N+1}} = max(0, S_{L_N} - z_N - k) \quad (2)$$
$$S_{H_0} = S_{L_0} = 0$$

Where S_{H_N} and S_{L_N} represent the positive and negative branches of the CUSUM algorithm, respectively. These branches simply represent accumulated negative and positive deviations from the target value. The parameter *k* is a configurable value that dampens the accumulation and determines the magnitude of the deviation required to trigger the detection. The values of either branch will quickly accumulate when the deviation continues to exceed the magnitude of the damping parameter. The two branches are generally examined separately in control charts, as illustrated in the example of Fig. (1): a clearly identifiable trend is visible in the positive branch, that is not easy to detect in the original data.

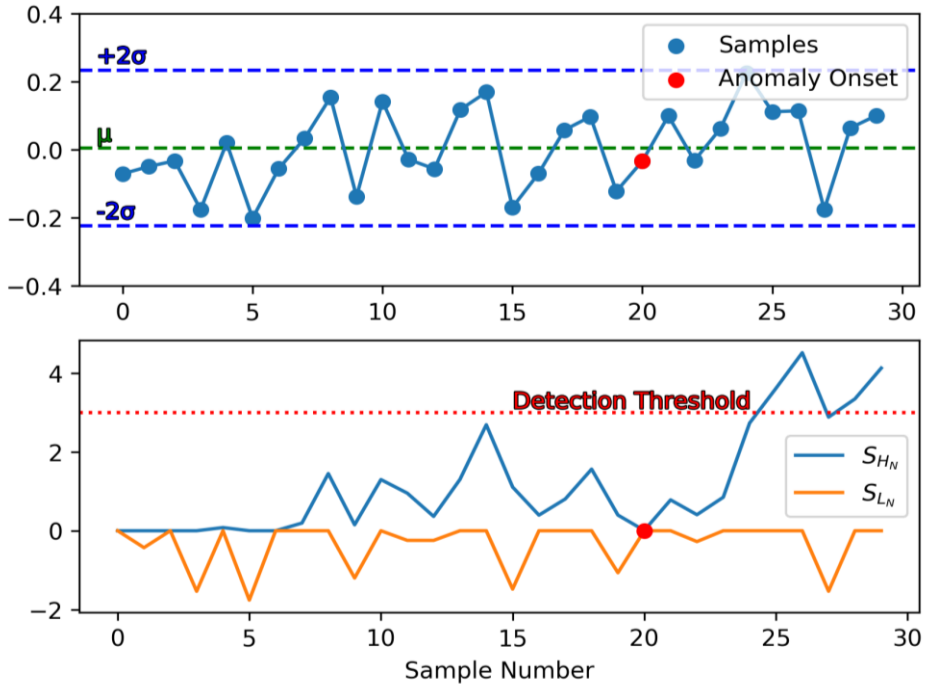

Figure 1 - CUSUM anomaly detection applied to random synthetic data using damping parameter k=0.05. Top - A process mean shift of 1σ from N(0,0.1) to N(0.1,0.1) is induced from sample number 20 onwards. All observations fall within two standard deviation of the mean of the group and the anomalous behavior fails to be detected by individually high deviations. Bottom - The CUSUM detection algorithm detects the cumulative run-away of the high (positive) branch of the process that is used to detect the anomaly after an arbitrary configured threshold is crossed.

The analysis described by Eq. (2) is defined for a single time series, but can be straightforwardly extended to combine observations from multiple instruments by evaluating, e.g., the minimum, maximum, mean or median of all instruments. Cross analysis of the control charts from the different observation points can provide insight into whether an anomaly is detected locally or affects most points of the array.

In the framework of the NEWTON-g project, the CUSUM analysis can thus be leveraged to real-time detect shifts with respect to the model-predicted value, occurring over gravity time series. It has therefore the potential to expose volcano-related gravity changes, using data from an array of gravimeters.

In the following, we report a simple Python code that can be used to apply the CUSUM algorithm on a time series:

```
from src.cusum import CUSUM
# Create a simple data model using a Pandas data frame from a file
model = Model(filepath)
plotModel(model)
# Estimate mean for the simple model used in the CUSUM algo
means, variances = model.simple(model="mean", window_length=1000)
# Get the upper and lower limits from the data frames
high, low = CUSUM(model, means, variances, k=0.5)
```

1.2 Empirical Orthogonal Functions

Besides performing "blind" anomaly detection, data mining techniques can be employed to expose the spatio-temporal features of the variation of interest. The Empirical Orthogonal Function (EOF) is a powerful technique, aimed to extract the most significant features of variability in a time-variant scalar field. It can expose spatio-temporal patterns and attributes a measure of importance to each of them (Bjornsson and Venegas, 1997). This technique is often applied to large time series in climate, meteorological, and atmospheric sciences, to study features like temperature and pressure fields; it is thus a promising technique to study continuous, geographically distributed gravity observations.

1.2.1 How the EOF works

Spatio-temporal correlations are multi-dimensional by nature, and an EOF analysis projects the data on its principal components, thus reducing the number of dimensions to only the most important modes of variability. It is important to emphasise that the analysis breaks down the data into orthogonal modes of variability that are purely statistical and thus do not necessarily represent a physical process. The results are thus primarily data modes and further interpretation is needed to relate them to physical processes. Therefore, this analysis should be used in conjunction with additional supporting observations.
Given a data set X, consisting in a (N x M) matrix, where the columns (M) represent the variable measured at a particular geographical location, and the rows the number of observations recorded at (N) different times. The data must be normalized and centered around the origin by subtracting the mean of each column:

$$\hat{X} = X - \bar{X} \tag{3}$$

The covariance matrix from the normalized data matrix is then calculated as follows:

$$\Sigma = \hat{X}\hat{X}^T \tag{4}$$

From which the principal components can be extracted using an eigendecomposition to find the eigenvalues λ_m and eigenvectors v_m of the covariance matrix Σ by solving:

$$\Sigma v = \lambda v \tag{5}$$

The solutions of Eq. (5) decompose the covariance matrix Σ into a set of orthogonal basis vectors. The eigenvector with the highest corresponding eigenvalue represents the direction of most variance in the data. The percentage of variation accounted for by each eigenvector can be found by dividing its associated eigenvalue with the sum of all eigenvalues. The

smallest eigenvalues and eigenvectors can be dropped from the analysis as they represent a negligible percentage of the overall variability in the data.
Every component of an eigenvector can be projected back on its geographical instrument location for a spatial visualization. The result is a correlation map that shows the value of the m-th mode of variability at different points. This map illustrates the spatial localization of variance. In other words, it predicts how well the observations from a mode at one grid point can be predicted from another.
To visualize the patterns over time, one can project the initial data set back on m-th particular eigenvector, commonly starting with the vector associated with the largest eigenvalue. These time series are referred to as the EOF expansion coefficients, and illustrates how the modes vary with time:

$$a_m = v_m X \tag{6}$$

The orthogonal basis remains a purely mathematical description of the variance in the data and should be treated with care. The results of this analysis do not immediately provide insight into the underlying physical processes. They merely illustrate different patterns of variability in the data that may later, in accordance with supporting observations, be attributed to certain processes. To further illustrate the technique we apply it to a simple synthetic scenario.

1.2.2 Synthetic EOF Analysis

To allow a better understanding of the EOF, we provide a synthetic case of study. In the simulation, a 20 x 20 (10x10 km) grid of 400 virtual stations is deployed on a plane. A spherical point source ascends in the center of the grid from 6 to 3 km depth and moves from 0 to 5 km eastward over an interval of 3600 steps. The gravity effect of the moving point source is calculated at the virtual stations, and white noise is added to the results. The synthetic data set X becomes a matrix where the covariance between two arbitrary stations (i.e. columns) on the grid is illustrated in Fig. (2). The change through time is represented by the change in color from blue to red.

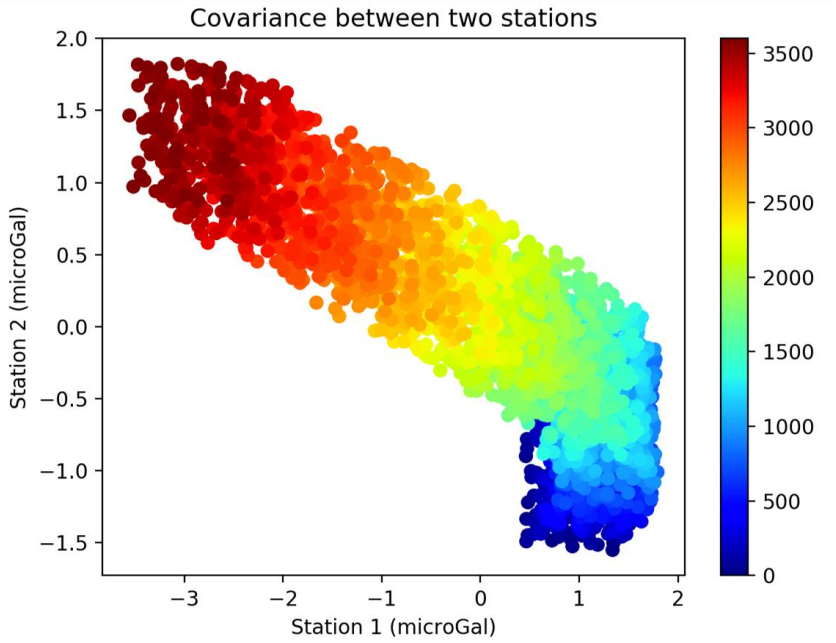

Figure 2 - Covariance between two arbitrary stations on the grid. The colors indicate the number of steps from the beginning of the simulation. From the figure there appears a small positive covariance at the start that is increasingly dominated by a negative correlation.

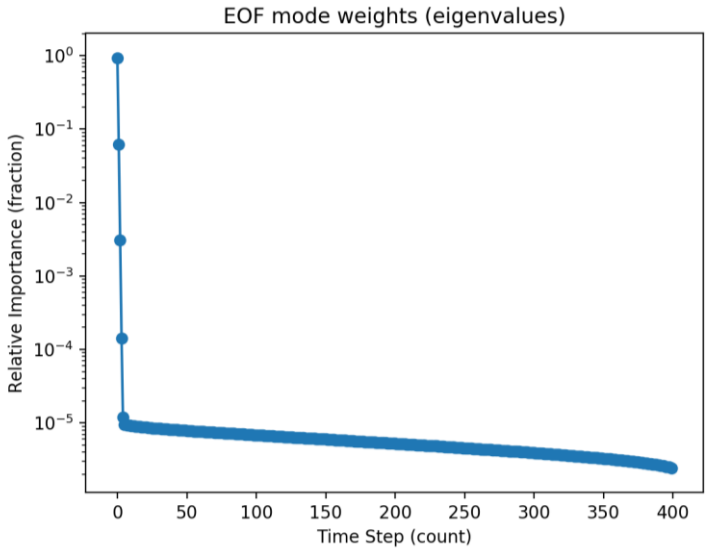

Figure 3 - Relative importance of the eigenvectors on a logarithmic scale. The first two modes, in particular, account for a significant part of the variance.

The eigenvalues (i.e. relative importance) of all eigenvectors, derived from the EOF analysis, are illustrated in Fig. 3, on a logarithmic scale. It shows five relatively important modes before nearly all variance is captured in the basis and what remains to be constrained is noise. The first two modes are illustrated in Figs. 4 and 5 and show two features that, with foreknowledge, hint at the ascent and lateral movement of the point source, respectively.

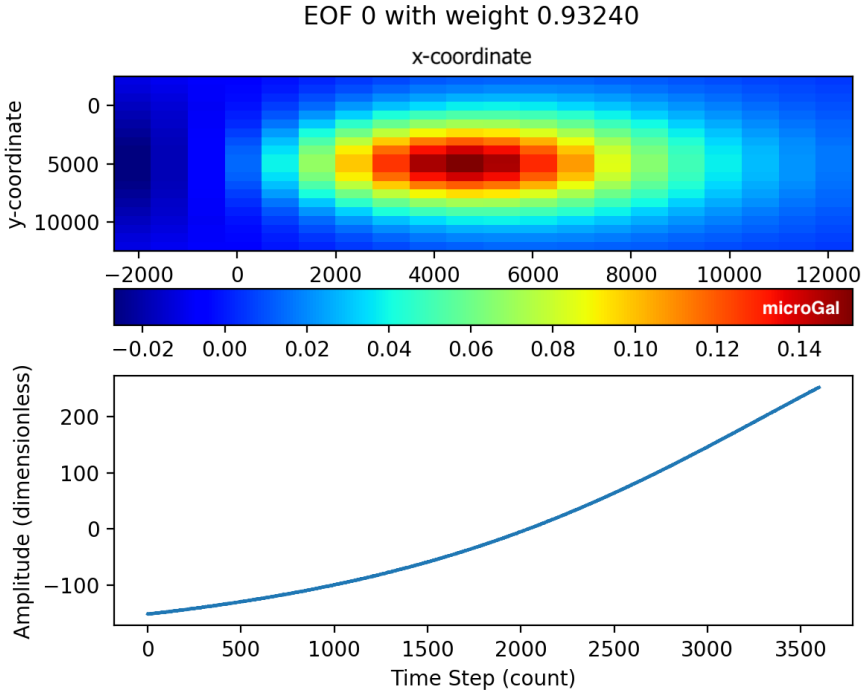

Figure 4 - The zeroth eigenvector projected back on the grid and its expansion coefficients.

In modes 3 - 4 (not shown), it appears that some variance is leaked over to these two eigenvectors in order to represent the full variance of the data set that cannot be captured in the first two orthogonal components. Fig. 6 shows that the modes starting from 5 are dominated by noise and contain no practical information at all. The rest of the eigenvectors can thus be discarded in the analysis, and we have essentially reduced the dimensionality of the problem from 400 to 5 significant dimensions.

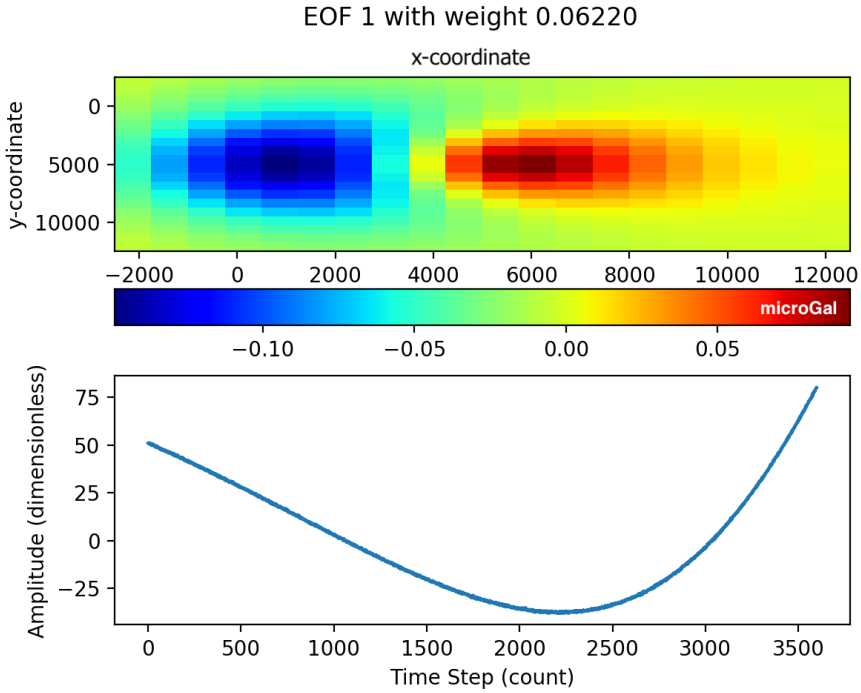

Figure 5 - The first eigenvector projected back on the grid and its expansion coefficients.

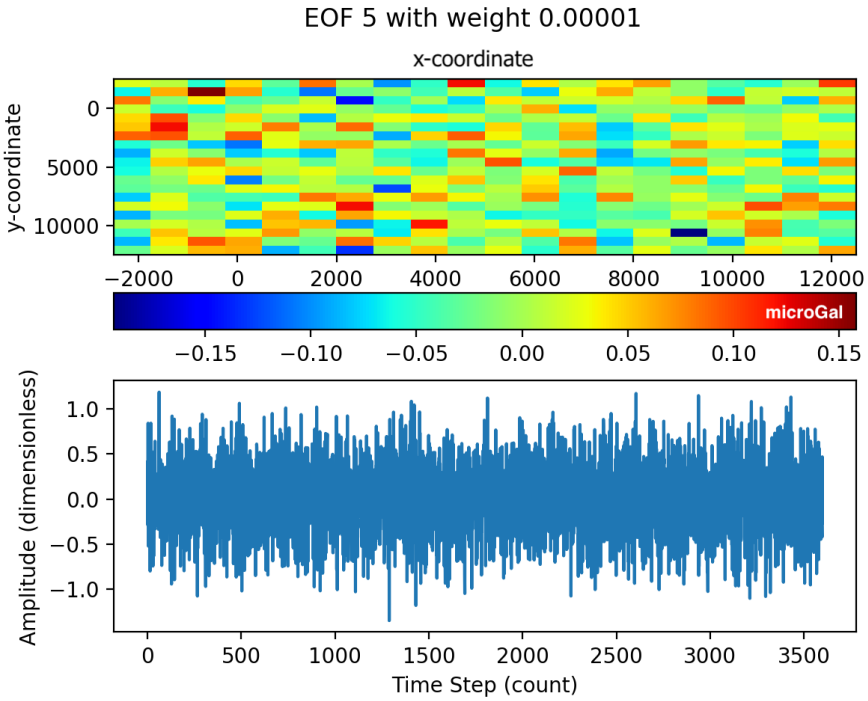

Figure 6 - The fifth eigenvector projected back on the grid and its expansion coefficients. No information can be derived from this figure or the following orthogonal components.

Being a simple analysis able to recognize coherent patterns between time series, the EOF can be used to complement change detection algorithms. In the framework of the data mining scheme of NEWTON-g, once the CUSUM algorithm identifies an anomaly, the EOF analysis is applied to extract more information, in terms of spatial and temporal patterns. Used in tandem the two methods allow to fully exploit the capabilities of the spatially distributed and continuously recording network of gravimeters.

The following code is an example of use of the developed package for EOF&PCA analysis:

```
from import src.linalg import getFEOF
# Create a simple data model using a Pandas data frame from a file
model = Model(filepath)
# Normalise the data (eq. 3)
model.normalise()
# Get orthogonal components and normalise the eigenvalues: then plot
w, v = EOF(model, normalise=True)
plotEOFEigenvectors(w, v, model, weight_cutoff=0.05)
```

2. Gravity changes simulator

The deployment of the NEWTON-g gravity imager was scheduled for the summer of 2020 (see Annex 1 - Part B of the Grant Agreement). The shut down in many European countries to fight the spread of COVID-19 pandemic, has caused delays in several project activities and it is likely that the deployment of the gravity imager will not be completed until the summer of 2021. In order to check the performance of the data mining tools, before real data from the array of gravimeters become available, we developed further software tools able to generate realistic synthetic time series of gravity changes. This forward simulation software is written in Python and designed to offer ease of use and versatility. The equations for the vertical gravity effect of bodies with simple geometric shapes (sphere, cylinder, thin rod, prism) and the formulations of Mogi (1958; point source) and Okubo (1992; shear and tensile faults) are used as forward models. Besides choosing the mass source type, the operator can modify the characteristics of the source (e.g., position, density, size) over time, through a linear property iterator, thus obtaining time series at different points of the considered array. The results can be either visualized in the application, or saved to a file, both as a map of gravity change over a given interval, and as a collection of time series from the available points.
Here we report an example of the Python code behind the gravity changes simulator:

```
from src.anomalies import SphericalAnomaly
#create the spherical anomaly source
denscontrast = -100
radius = 500
position = Position(500076, 4176413, -3000)
source = SphericalAnomaly(position, denscontrast, radius)
# simulate the source
results = source.simulate(receiver)
```

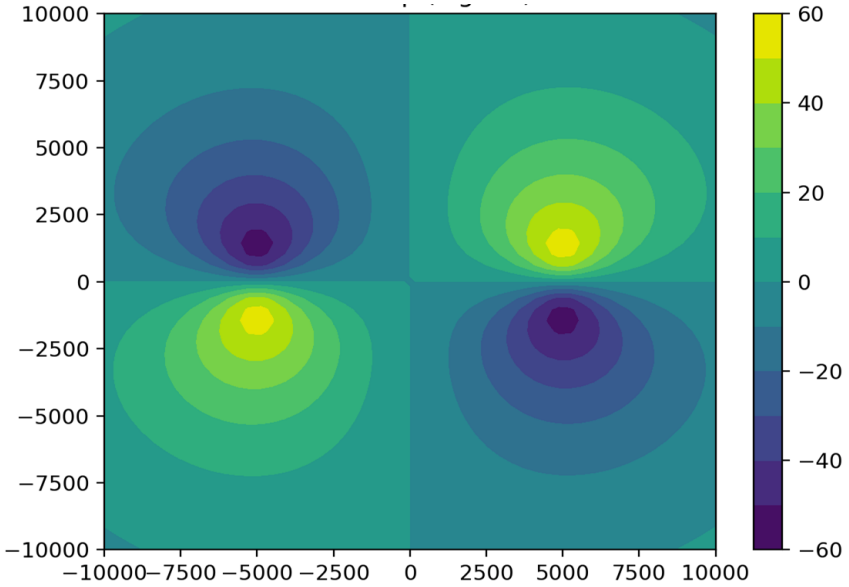

Figure 7 - Modeled gravity change (in μGal) of a simulated 10×10 kilometer strike-slip fault with 5 meter displacement, using the same parameters as in Fig. 4a of Okubo (1992).

Software benchmarks were run, aimed at validating the accuracy of the implemented models. Figure 7 shows the results obtained assuming an Okubo (1992) strike-slip fault, buried in a half-space with a flat surface. For calculations where the real topography can be safely taken into account, a digital elevation model (DEM) of Mt. Etna is included in the software. The output format of the gravity change simulator software is fully compatible with the developed data mining tools, implying that checks on the latter can be performed directly once a forward calculation from the simulator is available.

3. Assessing the performance of the data mining tools using synthetic time series

Two scenarios were simulated with the source modeling package to assess the detection capabilities of the data mining tools. The two scenarios include (Fig. 8):

- Dyke opening in the flank of Mt. Etna (Okubo tensile-like structure; depth = 3 km, dip = 90°, strike = 90°, length = 5 km, width = 5 km, host rock density = 2780 kg/m^3, fill density = 2700 kg/m^3)
- Density change within a deep spherical source.

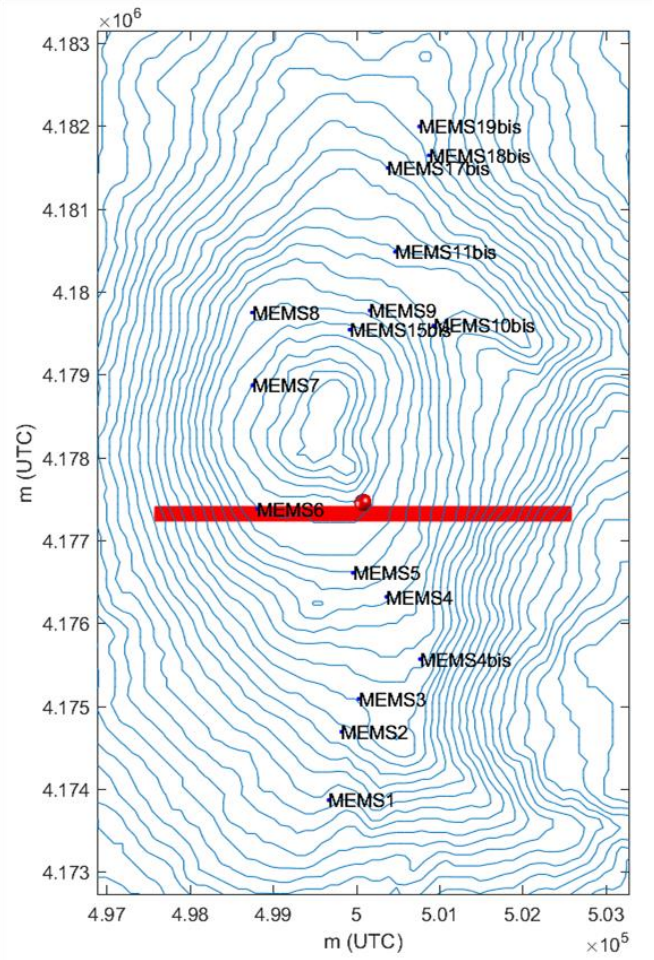

Figure 8 – Position of the source models that are used to generate the synthetic time series of gravity changes.

The stations used in the simulations (Fig. 8) are the ones selected by the algorithm for optimizing the configuration of the gravity imager, which was developed in the framework of WP4 (see D2.1 and the Part B of the 1st Technical Report of NEWTON-g).

These simulations were run for different rates of change and using different ramping functions, in order to assess the performance of the CUSUM detection algorithm under different conditions. Since the characteristics of the instrumental noise affecting the signal from the MEMS sensors during deployment at Mt. Etna are unknown, either white or auto-regressive (AR) noise is added to the synthetic time series, in order to study its effect on the performance of the data mining algorithms.

The following Python code is used to create a time series of AR noise:

```
from src.noise import autoRegressionModel
# Offset and an array of coefficients: the number of coefficients
determine AR(N)
coefficients = [-0.9]
offset = 0
sigma = 0.01
# t = 1000, 4 stations
shape = (1000, 4)
noise = autoRegressionModel(offset, sigma, coefficients, shape)
```

The algorithm performance is estimated by calculating the detection delay. The synthetic time series are built assuming (i) a quiet period of 5000 samples (ii) an anomaly simulated through linearly changing a model parameter (opening of the dyke-like source and density of the spherical source), (iii) other 5000 samples of quiet signal at the new level, after the development of the main change.
A white noise with amplitude equal to about 40% of the peak-to-peak amplitude of the main gravity change is added to the synthetic time series. Relying on a synthetic simulation, the parameters of the CUSUM algorithm were set to 100 samples for the sliding window length and 40 for the detection threshold.

3.1 Dyke opening

Results relative to the first scenario (dyke opening) are presented in Figures 9 and 10. Figure 9 shows the detection delay of the CUSUM algorithm versus the opening rate of the dyke model. The detection performance rapidly degrades for dyke opening rates below 0.25 cm/sample. Hence, in this particular case, the gravity effect of slowly-evolving dikes is more difficult to detect. This mostly depends on the size of the chosen sliding window: the larger the window is, the more sensitive the algorithm will be to slow changes. Thus, the choice of the window size has to be tuned on the expected time scale of the anomaly.

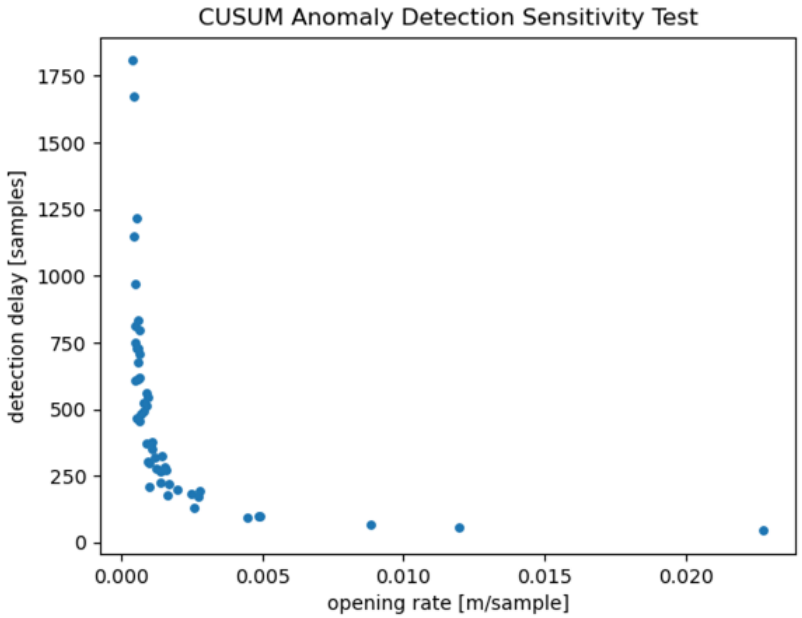

Figure 9 – Performance of the CUSUM algorithm versus the rate of opening of the dike-like gravity source. The algorithm performance is evaluated on the ground of the detection delay (i.e. number of samples from the start of the simulated anomaly to its detection).

Figure 10 presents a comparison between gravity and CUSUM time series. In the framework of the proposed scenario, while the CUSUM algorithms can be used to automatically detect the occurrence of the meaningful gravity change, the EOF analysis can provide insights into the spatial characteristics of the anomaly. In the case of the synthetic scenario discussed here, application of the EOF analysis allows to infer a spatial pattern typical of dike opening (Fig 11).

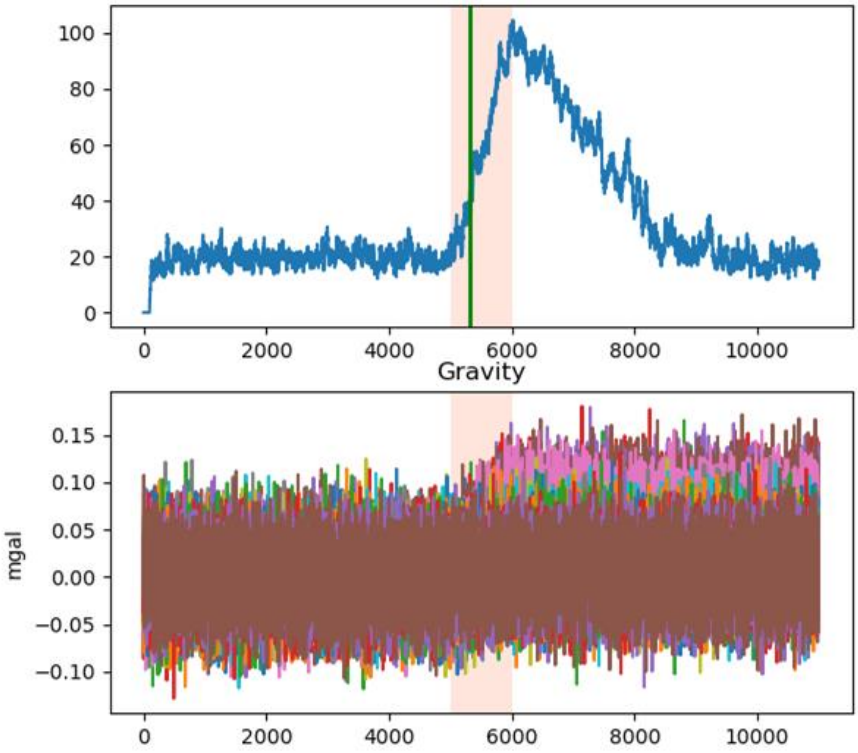

Figure 10 - Sensitivity test on the CUSUM algorithm, assuming a linear change (over the pink strip) in the opening of a dyke-like model (see text for details). The x-axis reports the number of samples in the time series. Top - CUSUM time series; the vertical green line marks the detection time (i.e., when the CUSUM threshold, set at 40, is reached). Bottom - Gravity time series simulated at 16 virtual stations on Etna (see Fig. 8).

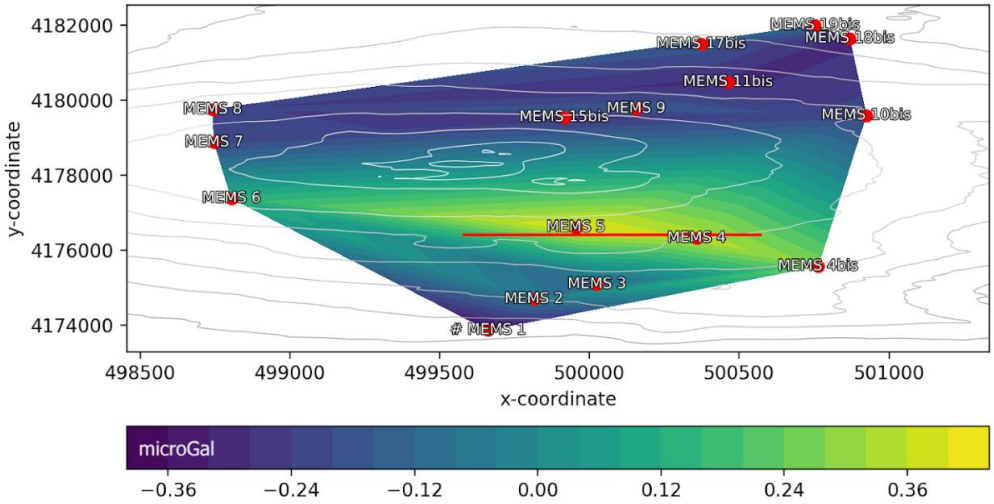

Figure 11 - EOF analysis (zeroth component) on synthetic data, obtained from the assumption of a dike opening scenario (see text). To better show the spatial (elongated) pattern (typical of a dike-related dynamics), we use different scales for the x and y axes (both in m). Visual inspection of the spatial pattern allows to also assess the orientation of the dike.

3.2 Density change within a spherical source

In the second scenario, we assume a spherical source at depth of 3 km, with a radius of 500 m. The density of the modelled source is decreased linearly, to simulate the substitution of resident, denser magma by lighter magma coming from below.

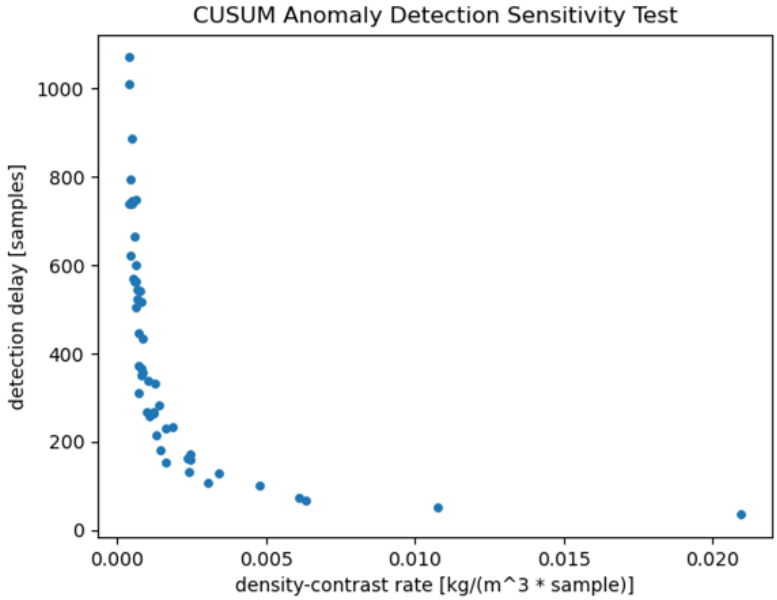

Figure 12 – Performance of the CUSUM algorithm versus the rate of density change within a spherical source. The algorithm performance is evaluated by the detection delay (i.e., number of samples from the start of the simulated anomaly to its detection).

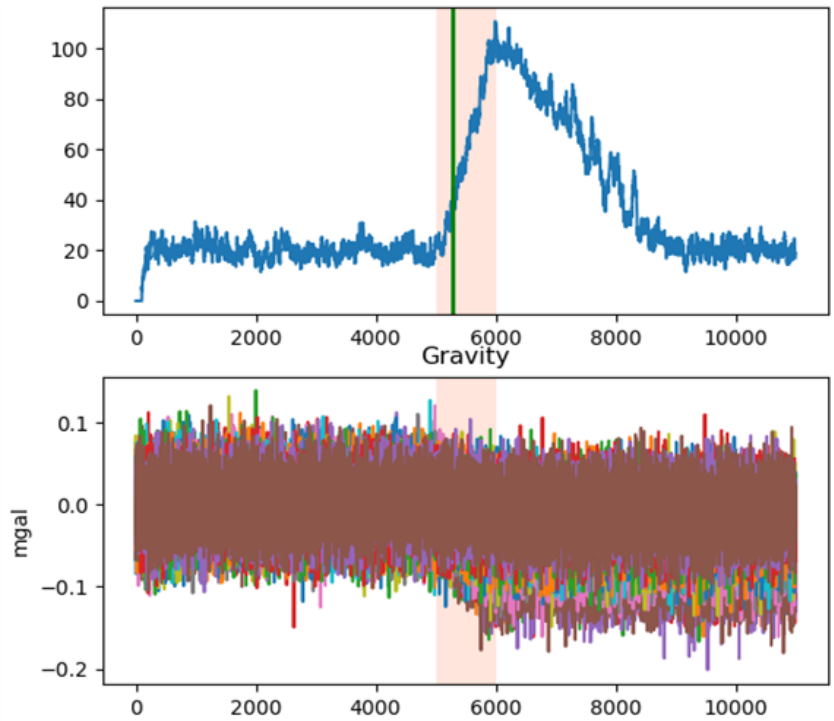

Figure 13 –Performance test on the CUSUM algorithm, assuming a linear change (over the pink strip) in the density of a spherical source (see text for details). The x-axis reports the number of samples in the time series. Top - The blue curve is the CUSUM time series; the vertical green line marks the detection time (i.e., when the threshold, set to 40, is reached). Bottom - Gravity time series simulated at 16 virtual stations on Etna.

In this case, the CUSUM algorithm maintains good performance for rates of density change higher than 0.0015 kg/m^3/sample (Fig. 12). The anomaly is detected by the CUSUM about 300 samples after it first appears in the gravity signal (Fig. 13).
It is worth noting that the performance of the detection algorithm can be improved by tweaking its setting parameters (i.e., model type, window size and *k*). The algorithm can thus be adapted to different dynamics. Hence, different versions of the algorithm, sensitive to different dynamics, can be run in parallel, to detect and classify different features of the same time series.

4. Assessing the performance of the data mining tools using quasi-real time series

In order to generate a set of "quasi-real" time series, we use the gravity data collected through a spring gravimeter (LaCoste & Romberg D-162) installed in the summit of Mt. Etna, during the summer of 2011, when several lava fountaining episodes occurred, all inducing a similar pattern of change in the gravity signal (Carbone et al., 2015). We consider the gravity signal during 7 eruptions and shift the time series along the time axis, in order to obtain the best possible superimposition of the gravity anomalies induced by the 7 fountaining episodes (Fig. 14a). To obtain a single averaged time series, most representative of the gravity signature of the 2011 fountaining episodes, we sum up the shifted time series, low-pass filter the resulting signal and divide it by the number of considered time series (black curve in Fig. 14a).

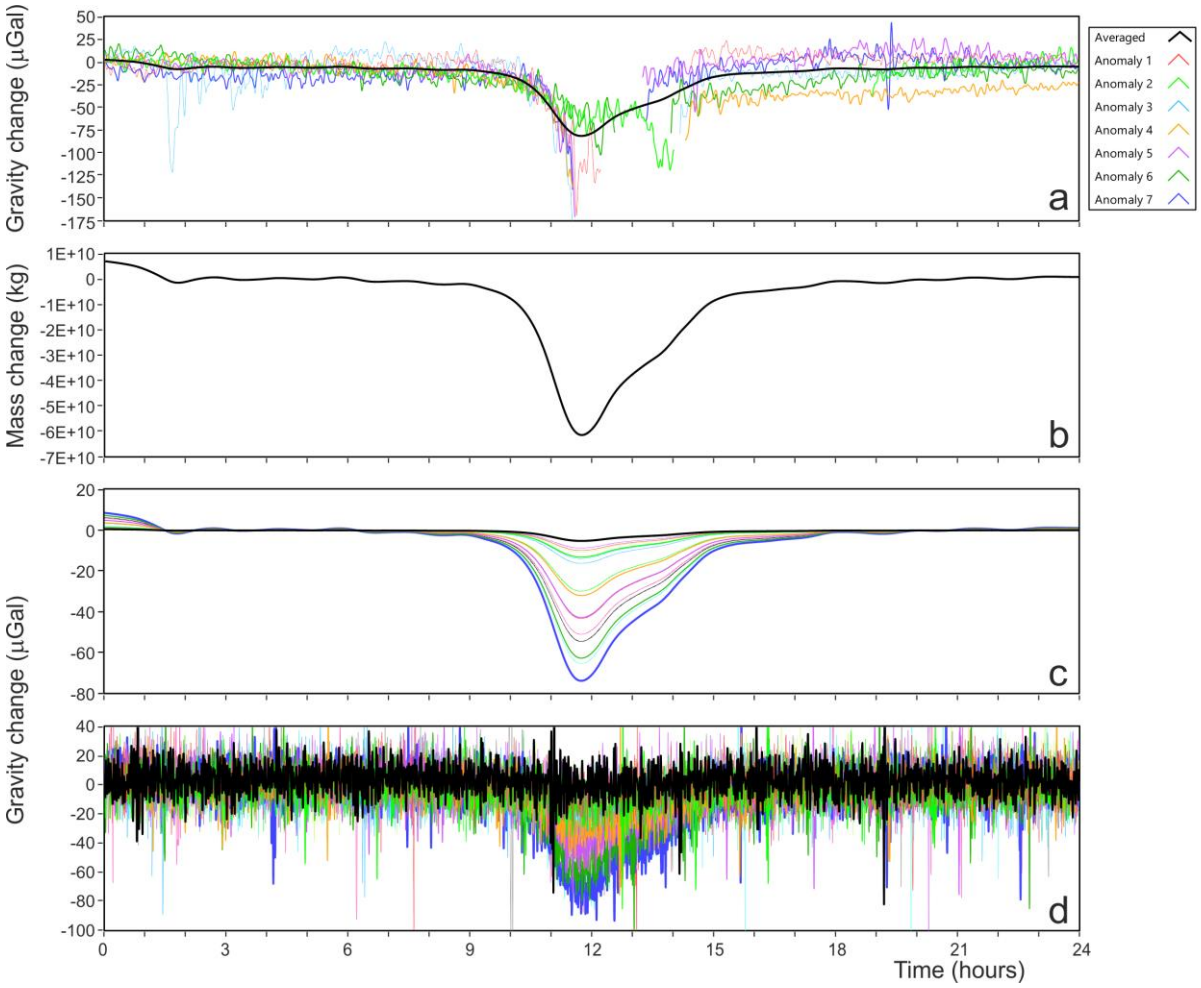

Figure 14 – (a) Time series of gravity changes acquired during the summer of 2011, each encompassing a lava fountaining episode. The black curve is the averaged and filtered time series (see text for details). (b) Time series of mass changes at the source inferred by Carbone et al. (2015) and needed to induce the averaged signal shown in panel (a), through the black curve. (c) Time series of gravity effects induced at the 16 inferred installation sites of the MEMS sensors by the mass changes of panel (b). (d) same time series as panel (c), but after addition of some background noise (see text for details).

As a further step, relying on the conceptual model proposed by Carbone et al. (2015) to explain the gravity anomalies produced by the 2011 fountaining activity, we assume mass changes within a static spherical volume as the driving process and calculate the time series of mass

change within the source volume (Fig. 14b), needed to induce the averaged time series at the installation site of the spring gravimeter in 2011. To perform this calculation, we assume a position of the mass source similar to that proposed by Carbone et al. (2015).
Using the time series of mass change at the source volume, we calculate the time series of gravity changes at the 16 points of an array in the summit of Mt. Etna (Fig. 14c). Also in this case, the coordinates of the points are those selected by the network optimization procedure described in D2.1.
Finally, we add to the gravity time series from the 16 observation points randomly chosen parts of the signal acquired through L&R D-162 during quiet periods in the summer of 2011. This strategy provides a reliable background noise model (Fig. 14d).
The obtained collection of "quasi-real" time series of gravity changes is used as an input to the developed data mining tools and allows to check their performance.
The CUSUM detection algorithm is applied to the "quasi-real" gravity data after defining the "in control" target value over a 100-sample window and setting the *k* value to 0.5. Results are shown in Figure 15. The detection delay depends on the chosen threshold; regardless, the ability of the algorithm to unambiguously detect the anomaly clearly appears in the CUSUM time series (bottom panel of Fig. 15), thus proving its effectiveness.

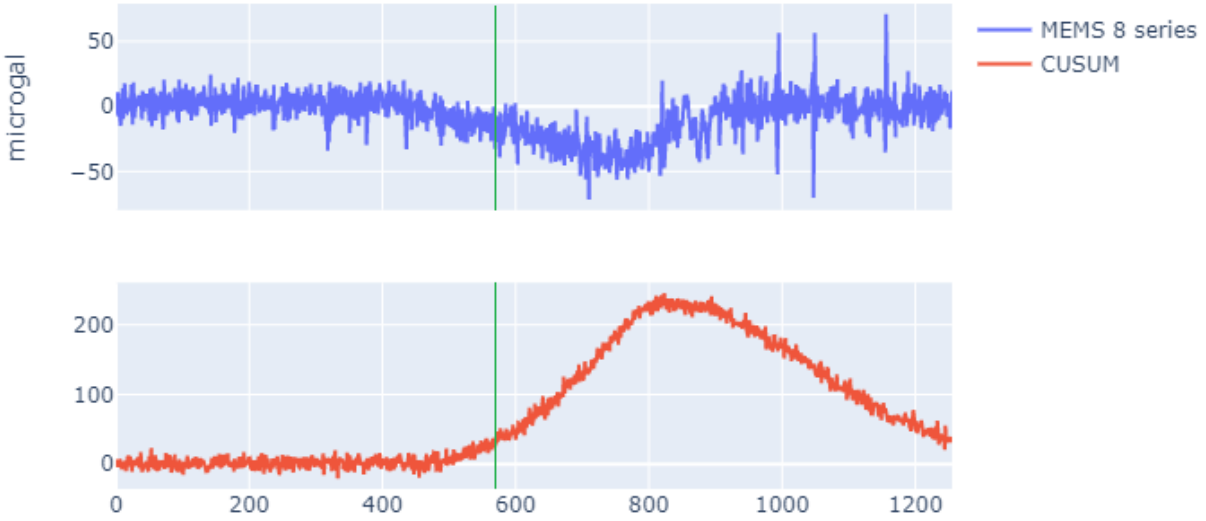

Figure 15 – CUSUM anomaly detection on quasi-real data. The x-axis reports the number of samples in the time series. Top panel - quasi-real time series obtained from gravity data recorded during the summer of 2011 (resampled at 50s). Bottom panel - CUSUM time series of anomaly detection. The threshold of 40 is reached at sample 580, about 1h after the start of the anomaly.

5. Concluding remarks and GitHub repository

In this document we present the data mining strategy that has been developed to automatically detect and pre-analyze meaningful anomalies occurring in the time series from the NEWTON-g's gravity imager. This strategy relies on two complementary methods: the CUSUM change detection algorithm and the Empirical Orthogonal Function (EOF). The former allows to highlight transient deviations from a reference model of the target time series and its detection capability depends on the input parameters (e.g., the size of the window sliding along the signal) that must be tuned on the characteristics (e.g., time scale) of the anomalies of interest. The EOF analysis can be applied downstream of the CUSUM algorithm and allows to expose possible spatio-temporal patterns of variability in the signal, thus providing insight into the underlying physical processes.
Because of delays induced by the COVID-19 pandemic, data from an extended array of MEMS gravimeters will not be available until the summer of 2021. Hence, to test the performance of the already developed data mining tools and those that are still under development, we designed a forward simulation software, able to generate realistic synthetic time series of gravity changes.

In the near future, we plan to:

- check how efficient a linearly increasing (or decreasing) reference model (instead of, e.g., the average) may be within the CUSUM detection analysis, to simulate the instrumental drift of the MEMS devices;
- perform sensitivity analyses to understand which signal-to-noise ratio needs to be met, under different condition, for a given anomaly to be detectable by the developed mining tools.

The source codes of the data mining software tools are included in the collective NEWTON-g GitHub repository (https://github.com/NEWTON-g).

References

Bjornsson, H., Venegas, S. A., 1997. A manual for EOF and SVD analyses of climate data. McGill University, CCGCR Report No. 97-1, Montreal, Quebec, 52pp

Carbone, D., Zuccarello, L., Messina, A., Scollo, S., Rymer, H., 2015. Balancing bulk gas accumulation and gas output before and during lava fountaining episodes at Mt. Etna. Sci. Rep. 5, 18049.

Mogi, K., 1958. Relations between the eruptions of various volcanoes and the deformations of the ground surfaces around them. Bull. Earthq. Res. Inst. Tokyo, 36, 99–134.

Okubo, S., 1992. Gravity and potential changes due to shear and tensile faults in a half-space. Journal of Geophysical Research: Solid Earth, 97(B5), 7137-7144.