

Coastal Risk Information Service (C-RiSe)

Python Code Guidelines

1) Introduction

This note provides guidelines on the use of the python routines provided with the C-RiSe data

The code is written assuming the data and software are copied to your laptop in a given directory structure, i.e.

The software is under a directory named Shared\Software\Windows directly under the C: Drive, as follows

Wind and wave software - C:\Shared\Software\Windows\WindWave

Surface Current software - C:\Shared\Software\Windows\Currents

Sea Level software - C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation

The data files are in a directory named Shared\Data directly under the C: Drive, as follows

Wave data - C:\Shared\Data\WindWave\waves

Wind data - C:\Shared\Data\WindWave\winds

Surface Current data - C:\Shared\Data\Currents

Sea Level Tide Gauge validation data - C:\Shared\Data\TideGauge\C-RiSe_TG_validation

Sea Level satellite altimeter data - C:\Shared\Data\C-RiSe\AltimetryByPass

If you save the data and / or software in a different directory structure, you will have to edit the python code to match.

The USB drive distributed to C-RiSe partners did not include the latest Tide Gauge Data (updated to the end of 2020), or an essential file for the Sea Level Validation software (gshhs_full.mat).

For access to these, or if you have any questions about the software or access to the other data sets please send an email to d.cotton@satoc.eu

2) Python environment

- Install Anaconda python 3 www.anaconda.com/products/individual
- Open Anaconda-Navigator application
- Select "Environments"
- Give it a name, e.g. c-rise2020

- Under package select python 3.6 and then click “create”
- On the right is the installed package listing
- Select “Not installed” at the top
- In the “Search Packages” box enter “numpy”
- Check the box to the left of “numpy” and select “Apply”
- Select “Apply” in the “Install Packages” window
- Repeat the last 3 steps for “netcdf4”, “matplotlib”, “scipy”, “basemap” (use these specific named versions)

3) For Wind, Wave and Surface Current Data

a. Using Spyder to open, edit and run python code

- Select “Home”
- Ensure the subtitle is “Applications on c-rise2020” (or your chosen environment name)
- Select “Install” under the “Spyder” application
- Once installed select “Launch” under the “Spyder” application
- Open a python program, by selecting “file” at the top left of the window and searching through the directory structure to find the file needed: e.g. windstats_point_py3.py
- The code appears on the left of the Spyder window in an editor
- The first statements after the code head import the required python packages
- Under this is the location of the data directory:
- `datdir = '/Users/ellis/Data/L4winds'`
- Change this to the data directory on your computer
- Use the green arrow at the top of this window to save and run the code
- The code will run in the interactive python area (bottom right window)
 - Respond to the prompts for location or date information
- To see the output select “Plots” in the top right window
- You can save these plots by clicking on the disc icons at the top of the right hand window.

b. Analysis of Wind, Wave Climatology Data

Directory Structure

- The Wind Wave code is at:

C:\Shared\Software\Windows\WindWave

- The code requires the wind and wave climatology data in
/Shared/Data/WindWave/winds, and /Shared/Data/WindWave/waves

- Before running them check that the routines have the correct directory path to the source data files. If they have not then they should be edited so that they have the correct paths (e.g. in spyder or a text editor such as notepad)
- Some routines will save time series and plots. They will save these to /Shared/Results/.
- You may need to change this path, again it can be edited in spyder or on a text editor such as notepad.

Overview of Routines

- To run all the following routines,
 - Open Anaconda-Navigator application
 - Select “Environments”
 - Choose c-rise2020 and “Open Terminal”
 - Change directory to the location of the windwave code (e.g. C:\Shared\Software\Windows\WindWave)

```
cd C:\Shared\Software\Windows\WindWave
```

- and type python
- and from inside python window type

```
exec(open("programname.py").read())  
(replace programname with the name of the programme you want to run)  
..and answer the prompts
```

- It may be necessary to reinstall the correct version of matplotlib, as follows:
- In the python terminal window type:

```
conda install matplotlib=3.2
```

- *wavestats_CCI_point_py3.py* – Generates graphs of monthly mean wave heights and distribution at a location – saves figures to /Shared/Results/.
- *wavestats_CCI_point_timeseries_py3.py* – Generates time series of monthly mean wave heights and distribution at a location – saves figure to /Shared/Results/.
- *wavestats_CCI_region_py3.py* – Generates a maps of: monthly mean significant wave heights (SWH), standard deviation in SWH, and number of SWH measurements used.
- *windstats_point_py3.py* – Generates a graph of monthly mean wind speed at a location – saves figure to /Shared/Results/.
- *windstats_point_timeseries_py3.py* – Generates a time series of monthly mean wind speed at a location – saves data and figures to /Shared/Results/.
- *windstats_region_py3.py* – Generates a map of mean monthly wind speed (and direction if the region is small enough - 4° x 4° or less) for a selected area.

- *windstats_region_seasonal_py3.py* - Generates a map of mean 3- monthly wind speed (and direction if the region is small enough - 4° x 4° or less) for a selected area. To save figures click on the disk icon

c. Analysis of Surface Current Data

Directory Structure

- The Surface Current code is at:

C:\Shared\Software\Windows\Currents

- The code requires the Surface Current climatology data in /Shared/Data/Currents/currents_L4
- Before running them check that the routines have the correct directory path to the source data files. If they have not then they should be edited so that they have the correct paths (e.g. in spyder or a text editor such as notepad)
- this path, again it can be edited in spyder or on a text editor such as notepad.
- We hope to provide updated surface current data later, and will provide new python code to analyse this data set.

Overview of Routines

- To run all the following routines,
 - Open Anaconda-Navigator application
 - Select “Environments”
 - Choose c-rise2020 and “Open Terminal”
 - Change directory to the location of the Currents code (e.g. C:\Shared\Software\Windows\Currents)

cd C:\Shared\Software\Windows\currents

- and type python
- and from inside python window type

exec(open("programname.py").read())
(replace programname with the name of the programme you want to run)
..and answer the prompts

- It may be necessary to reinstall the correct version of matplotlib, as follows:
- In the python terminal window type:

conda install matplotlib=3.2

- *curplot_point_month.py* – Generates a time series of current speed and direction for the selected year, and also for selected months. You will be prompted for new months, until you enter return. To save the files, click on the disc icon.

- *currents_region_day.py* – Generates a plot of daily mean current speed (and direction if the region is small enough - $4^\circ \times 4^\circ$ or less) for the selected region, year, month and day. To save the file, click on the disc icon.
- *currents_region_month.py* – Generates a plot of monthly mean current speed (and direction if the region is small enough - $4^\circ \times 4^\circ$ or less) and standard deviation for the selected region, year and month. To save the files, click on the disc icon.
- *currents_region_monthly.py* – Generates a plot of monthly mean (across all years) current speed (and direction if the region is small enough - $4^\circ \times 4^\circ$ or less) and standard deviation for the selected region and month. To save the files, click on the disc icon.

4) Analysis of Sea Level Data

Directory Structure

- The Sea Level code is at:

C:\Shared\Software\Windows\SeaLevel\SeaLevelValidation

- Folder *python_code/* contains the code for the validation
- All figures generated by the code are saved to *figures/*
- Statistics and processed data are saved to *data/*
- The code requires the along-track time series altimeter data (<mission>_stats_ales_P<cycle>.nc) in /Shared/Data/C-RiSe/AltimetryByPass
- There are two altimetry data sets one for the Jason missions (j1j2j3) and one for Envisat/Altika (n1sa). These need to be run separately for each tide gauge you validate

IMPORTANT – The USB Drive is missing a necessary file (*gshhs_full.mat*) which should be copied into the same directory as the rest of the sea level validation code:

C:\Shared\Software\Windows\SeaLevel\SeaLevelValidation\python

Performing Validation at a Selected Tide Gauge Station

- The USB drive was sent out before the updated Tide Gauge data (to the end of 2020) were available. We will provide a link to download the updated data on request.
- Open Anaconda-Navigator application
- Select “Environments”
- Choose *c-rise2020* and “Open Terminal”
- It may be necessary to reinstall the correct version of *matplotlib*, as follows:

- In the python terminal window type:
- `> conda install matplotlib=3.2`
- Change directory to the location of the sea level code (e.g.
`C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code`)

`cd C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code`

- Next, suppose that you want to validate data at Pemba using Jason data (the second parameter of the function is the altimetry dataset, which can take the values 'j1j2j3' (Jason missions) and 'n1sa' (Envisat/Altika)):

`python genStats.py Pemba j1j2j3`

- If you then wanted to validate data for Sainte Marie using Envisat/Altika data:

`python genStats.py Sainte_Marie_2 n1sa`

- The above command will produce figures showing the correlation and root mean square (RMS) difference as a function of distance from the coast for the closest altimetry passes. It will also produce a figure comparing the sea level time series from altimetry and the tide gauge for the “distance from coast” providing the maximum correlation. Finally, it will also generate a map showing the location of the tide gauges and the selected altimetry passes. All figures will be saved to folder figures/.
- Note that the performance of the altimetry data is assessed as a function of distance from the coast by assigning such data to distance bands of X km width starting from the 0- X km band. Only data that fall within Y km of the tide gauge are used. For each altimetry pass we obtain one altimetry value by averaging all the records falling within the selected distance band. Records with an absolute sea level anomaly larger than 2 m or three standard deviations above the mean are rejected prior to computing the average. The corresponding tide gauge matching value is obtained by linearly interpolating the tide gauge observations to the time of the altimetry pass.
- Some locations may not give results for one or other of the data sets, for instance if there are no tracks close enough to the Tide Gauge. If this happens we suggest to try the other data set.
- The width of the distance bands, X , and the maximum distance from the tide gauge, Y , can be adjusted by editing lines 12 and 13 of `genStats.py` (`python_code/`).
- Note: you can run `genStats.py` also from within the Python interpreter. To do that, start the Python interpreter from the shell and run the following commands:

```
from genStats import genStats
genStats ("Pemba", "j1j2j3")
```

Plotting sea-level maps

- Open Anaconda-Navigator application
- Select “Environments”
- Choose c-rise2020 and “Open Terminal”
- Change directory to the location of the sea level code (e.g.
C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code)

```
cd C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code
```

- The plotting routine needs the input file:
“\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\data\statsAltiTracks_mission”. Where “mission” is n1sa or j1j2j3.
- If this file is missing, then “plotStatsMap.py” will not run. This file is created by running “genStatsTracks.py” as follows (n.b. It can take a long time (~8 hrs) to run):

```
python genStatsTracks.py mission (mission is n1sa or j1j2j3)
```

- Function plotStatsMap.py allows us to plot maps of a preselected domain showing the amplitude and phase of the sea-level annual and semi-annual cycles, the spatial distribution of sea-level trends, and the standard deviation of detrended and deseasoned sea-level anomalies. Type:

```
python plotStatsMap.py lon_min lon_max lat_min lat_max stat mission (mission is n1sa or j1j2j3)
```

- where lon_min, lon_max, lat_min, lat_max denote the four vertices of a rectangle encompassing the domain you are interested in. They represent coordinates and have units of degree. The fifth input “stat” denotes the statistic you want to plot:
- amp_an : amplitude of the annual cycle
- phase_an : phase of the annual cycle
- amp_semian : amplitude of the semi-annual cycle
- phase_semian : phase of the semi-annual cycle
- trend : trend
- sdev : standard deviation of detrended and deseasoned sea-level anomalies
- The last input ‘mission’ identifies the altimetry dataset: ‘j1j2j3’ (Jason missions) and ‘n1sa’ (Envisat/Altika).
- As an example, the following will produce a map of regional trends in the domain (10°E-65°E,40°S-10°S):

```
python plotStatsMap.py 10 65 -40 -10 trend n1sa
```

- The figure will be produced in the directory
“\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\figures

Plotting time series of sea-level anomalies for selected points on map

- Open Anaconda-Navigator application
- Select “Environments”
- Choose c-rise2020 and “Open Terminal”
- Change directory to the location of the sea level code (e.g.
C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code)

cd C:\Shared\Software\Windows\SeaLevel\SeaLevel\Validation\python_code

- The function `plotSeriesSelect.py` allows us to select a point along a satellite track and plot the sea-level anomalies for that point.
- To run it, start the “ipython” interpreter:

```
ipython  
from plotSeriesSelect import plotSeriesSelect  
plotSeriesSelect('n1sa')
```

- This function takes one parameter denoting the altimetry dataset: ‘j1j2j3’ (Jason missions) and ‘n1sa’ (Envisat/Altika). The statement above will open a map showing all the altimetry tracks (maximize the window to see the tracks more clearly). Select a point on the tracks using the mouse. This will produce a figure of the sea level anomalies at the selected point (it might take several seconds while it extracts the data), and also a figure of the tracks with the selected location highlighted. Save the figures by clicking on the file icon.
- To exit the iPython interpreter

```
quit()
```

- The sea level anomaly time series, and times (in decimal years), will be saved in separate files created under `.data`. They can be imported to excel and combined to generate a time series for further analysis.