exercises

January 9, 2023

1 UNISACSI Exercises

This notebook contains a set of exercises to be used as alternative to the Matlab excercises from Jan Even. If you don't have any programming experience, I recommend you to stick to Matlab, as Jan Even spends more time on the basic concepts of programming in his lectures. This Python set of tasks is supposed to be an alternative for those of you who already have some experience with Python.

1.1 Importing packages, setting paths, ...

As in every Python script, first import all packages needed for the following code:

```
[3]: import unisacsi.Ocean as Oc
     import unisacsi.Meteo as Met
     from unisacsi.MET_model_download import download_MET_model_static_fields,
      ↪download_MET_model_data
     import matplotlib as mpl
     import matplotlib.pyplot as plt
     import pandas as pd
     import numpy as np
     import cartopy.crs as ccrs
     import gsw
     import cmocean as cmo
     import windrose
     import xarray as xr
     import seaborn as sns
```

Then, download the folder with the example and map data and save it locally on your computer. Then, specify this location here:

```
[4]: path_data = "/Users/lukasf/OneDrive - Universitetssenteret på Svalbard AS/
      ↪unisacsi_example_data/"
```

Now you are ready to dive into the exercises! You might find some help and hints in the example notebook also available from the Github repository ;)

1.1.1 1. CTD profiles

Some of the data you have gotten is CTD profiles from the AGF-214 cruise 2021. Read all profiles using the unisacsi-toolbox. Explore the dataset. In what type of variable are the profiles stored?

Which sub-levels are there? Compare the temperature and salinity profiles from stations 3 and 23!

1.1.2 2. CTD Section

Plot a CTD section containing the station numbers 1,3,6,10,12,18,23 for each of the variables (sometimes called CTD-transect; see the slide about CTD). Check the examples for already available functions for this task ;) Play around with the different argument options, e.g. for the interpolation! Can you put the two individual profiles from Exercise 1 into context?

1.1.3 3. Mapping with Cartopy

Wouldn't it be nice to know, where these stations actually are located? Make a map with the locations of the Billefjorden along-fjord CTD stations using the CTD map toolbox function. Can you make a nicer map using the provided map data and the toolbox?

1.1.4 4. T-S Diagrams

A TS-diagram is an axis system with salinity on the x-axis and temperature on the y-axis. Since density depends mainly on temperature and salinity, a TS-diagram also gives background contours showing the density of water at any given point in this coordinate system. Pressure also affects density but is a chosen constant for the TS-plot. T-S diagrams are very useful to investigate e.g. the origin and interaction of water masses. Plot a "normal" T-S diagram of all stations in the Billefjorden section (one color for one station, use the respective toolbox-function). What do the density lines tell you about the properties of dense and light water masses? Which property determines the density most in cold waters? Plot the same T-S diagram again, but now with the Oxygen measurements color-coded.

1.1.5 5. Wind Measurements

The Campbell AWS (Automatic Weather Station) measures wind speed and direction. You can find example data files in the respective folder. Read the data! Why are there two different files for the same period?

1.1.6 6. Polar vs. Cartesian Data

The wind direction is given in compass directions (0-360 deg, 0 deg = North). Transform the direction and speed data from sensor 1 into u (east) and v (north) components. Always use pen and paper to work out the trigonometry first! And what kind of angles do Python's trigonometric functions (e.g. sin and cos) accept, degrees or radians? Careful with the compass direction vs. the polar coordinate angle! Can you check your code by back-transforming u and v into speed and direction?!

1.1.7 7. Wind Rose

Make a wind rose plot of the time series. Why do we make such plots? What can they tell us? Make sure you get the degrees right and have north up in the plots.

1.1.8   8. Wind time series

Plot time series of the wind measurements, both for each component and with a stickplot. Maybe it is wise to plot a smaller part of the series. Try!

1.1.9   9. Temperature & Humidity

In addition to the wind measurements, read also the air temperature and humidity measurements from the Campbell AWS. Make a time series plot for humidity and air temperature (sensor 1), two different variables with different units, in the same graph. Also add ylabels to both coordinate systems (i.e., sides) of the graph.

1.1.10   10. Wind speed and direction

Plot also a time series plot of wind speed and wind direction. What about the line for direction? Is it making sense? Change the 'linestyle' to 'none' and add dots as 'marker' for the line! What about the wind direction y-axis ticks? Can you change the labels into "N", "NE", "E", etc?! Play around a bit with changing properties of lines and axes.

1.1.11   11. Combine time series data

Even though the temperature/humidity and wind measurements are initially sampled at different frequencies and read separately, it might be useful to combine them into one Dataframe. Find the respective pandas function and create a new dataframe with all measurements from 2022-09-14 as individual columns. How does the function handle the gaps in the temperature time series with respect to the wind measurements? Interpolate the temperature measurements onto the high temporal resolution.

1.1.12   12. Modify dates and times

Tinytags are small and rugged temperature (and humidity) sensors we use in various settings to complement the full Campbell (or HOBO) weather stations. Read the example data file from TT6 using the toolbox. In what way are the time stamps in an inconvenient format? Fix the time series by interpolating it to a more convenient base! Furthermore, resample the time series to a 10-minute resolution! (Hint: pandas resample)

1.1.13   13. NetCDF files

NetCDF is an advanced form of data file that is widely used for large datasets. You can also choose to load smaller parts of the data arrays so that you do not fill up your workspace/memory unnecessarily. Here we are going to read some data from an atmosphere (weather) model, for the Svalbard region. NetCDF format lets you study the contents before loading. Use the xarray package to study the contents of all AROME-Arctic files. Study especially the dimensions and the different variables and coordinates. Load only the time period 2022-05-02 10:00:00 – 2022-05-04 08:00:00. What are the units of the temperature and humidity variables? Maybe convert them? Use the toolbox to plot an overview over whole of Svalbard to check which area is covered by the dataset (plot the first time index of the surface geopotential devided by the gravitational constant, which variable is that?). Consider using the longitude and latitude as your x/y-coordinates. Extract the Billefjorden area from the whole dataset and plot the temperature field from 2022-05-03 05:00:00.

1.1.14   14. Hovmöller Diagram

The example data folder contains one netCDF file with lidar wind profile measurements. Read the file and have a look at the different variables and their dimensions. Plot wind speed time series for the time period 2021-02-22 – 2021-02-24 from the instrument "Windcube" (wc) for all height levels. Plot also a Hovmöller diagram for the whole time period.

1.1.15   15. Averaging

Calculate average wind speed profiles for 2021-02-22 and 2021-02-23. Plot the individual wind profiles for the two days (two different colors). On top, add the average profiles for the two periods.

1.1.16   16. Download AROME-Arctic model data

You can download model data from AROME-Arctic using the toolbox. The configurations are set in an extra configuration file (also provided on github, will be introduced in class). Download the static fields from AROME-Arctic with a resolution of 2.5 km. Then download a time series (the last week) of temperature and wind data from the model grid point closest to Svalbard Airport. Compare against measurements (can be donwloaded manually from https://www.seklima.met.no and the resulting files can be read using the toolbox).

1.1.17   17. Read data from a textfile yourself

For most of the instruments you will use in the field, there are functions to read the data available in the unisacsi toolbox. However, it is essential that you are able to read and handle such data yourself. Practice developing a routine to read data from a text file by reading the radiation example data yourself (instead of using the respective function from the toolbox.)

1.1.18   18. Spectral Analysis

Read the mooring time series from the SeaGuard serial number 2306. First, plot a time series of pressure. What is the problem with the data? Fix it and plot the time series again. Can you see a prominent periodicity in the series? You might need to zoom into the plot or plot a shorter period. Using the scipy and spectrum packages (check the documentations for hints which functions to use!), plot a periodogram of the detrended (find the corresponding function from the scipy-package) time series! Can you recognize the frequency peak you identified in the time series? What are the other peaks? Use multitapering to smooth the periodogram and plot a more meaningful power spectrum.