**University of Oldenburg**
*Python Programming in Energy Science*, WiSe 2023/2024
Dr. Martin Dörenkämper, martin.doerenkaemper@iwes.fraunhofer.de

# Assignment 3

All tasks must be solved without numpy, scipy, pandas or other advanced libraries, except for loading the data as specified below. You are, however, allowed to (and should) make use of help from matplotlib examples from the internet. Please impress us and create professional and nice looking (in science we would say "paper ready") figures in this excercise. For incomplete figures (e.g. missing axes labels, units etc.) will result in less points in the grading.

**Data:** We will need some data for the visualisation in this exercise. The data can be loaded by executing the following commands, that load the data that are provided along with this exercise.

```
import numpy as np
ws100max=np.genfromtxt("NEWA-WSMAX-100m-30year.csv",delimiter=",")
lill2009newa=np.genfromtxt("VMM-WRF_D03_LILLGRUND_2009.csv",delimiter=",",skip_header=1)
ws100=lill2009newa[:,3]
ws050=lill2009newa[:,1]
ws150=lill2009newa[:,4]
```

This will provide the following one dimensional arrays `ws100`,`ws050`,`ws150` and the two-dimensional array: `ws100max`. FYI: The data are taken from the New European Wind Atlas that was created between 2015 and 2019 and can be accessed freely on the web: https://map.neweuropeanwindatlas.eu/

E4. **(10 points)** Histograms and Weibull Distribution.
Histograms are a very important concept in visualizing and also analyzing the statistics of wind time series. The statistical distribution of a wind time series can be approximated by the Weibull distribution, which is defined as:

$$f_W(U) = \frac{k}{A}\left(\frac{U}{A}\right)^{k-1} \exp\left(-\left(\frac{U}{A}\right)^{k}\right) \tag{0.4}$$

with $f_W(U)$ - the resulting Weibull frequency, $k$ - the shape parameter that describes the shape of the distribution and $A$ - the scale parameter that is related to the average wind speed. (Hint: Reasonable values for hub height offshore winds in the North Sea are around $A \approx 10$ and $k \approx 2$).

Create a python function that takes scale, shape and horizontal wind speed as input parameters and returns the Weibull frequency $f_W$. Map that function to a constant shape and scale parameters and a list of horizontal wind speeds between 0 and 25 $\mathrm{ms}^{-1}$ using a spacing of 1.0 $\mathrm{ms}^{-1}$.

Plot the resulting histogram from the wind time series specified in `ws100`. Add the Weibull distribution from the equation 0.15 as a line to that plot on top of the histogram. Play around with shape and scale so that the function approximates the histogram.

E5. **(10 points)** Matplotlib gallery.
The matplotlib gallery provides a large variety of plots and gives access to the python code to reproduce those: https://matplotlib.org/stable/gallery/index.html. Select an example (no plotting of fields, i.e. 2D data please) and plot the data from one of the lists above in the in the style you selected. If the example contains code from other libraries (e.g. numpy), you are allowed to use it here. Change some properties of your choice (line styles / colors / labels) of the plots and comment (can be in the code) what you intended to and have changed.

E6. **(10 points)** Wind Speed Profile.
Plot a vertical profile of the wind speed with the height at the y-axis and the wind speed of the x-axis for the three heights 50 m, 100 m and 150 m using the data from the arrays ws050, ws100 and ws150. Add the error bars for each of the heights. (Hints: You might want to use the numpy functions `np.mean` and `np.std` to calculate average and standard deviation. The matplotlib gallery might help you finding a good plotting function.)

E7. **(10 points)** Plotting of 2D data fields.
Write a function that plots the fields `ws100max` using at least three different matploltib plotting methods (e.g. `imshow, pcolormesh, contour, ....`). How do you need to transform the data in the array to let `imshow` produce the exact same result as pcolormesh? For simplicity you can assume equal axes in pcolormesh. Plot a figure that includes two subplots of imshow and pcolormesh that look the same.

E8. **(10 points)** Plot arrays / subplots.
Please combine from the above used examples at least three subfigures into one. You should use at least two subfigures that have a different width/height combination (This means 3 subplots of 2D data fields of each same width and height don't count). Combine the figures in a way that the resulting figure file looks nice. Add labels (e.g. A,B,C) to the subfigures. The figure should be saved as non-vector graphics (e.g. png) using a reasonable figure size and quality.