Week 4 — *Introduction to Matplotlib*

The goal of the present exercises is to discover the main usage of the Python module Matplotlib. This learning necessarily goes through practicing.

Exercise 1: *Plotting data points*

- Create a new python script. In the header import the pyplot module of matplotlib

```
import matplotlib.pyplot as plt
```

- Create two lists x and y to contain each 6 elements

```
x = [0.,2.,4.,6.,8.,10.]
y = [0.,0.,0.,1.,1.,1. ]
```

- Plot the curve by using the plot function of the **Matplotlib** module
- Request to plot only data points with rounded symbols.
- Request to plot with crosses symbols and dashed lines.

Exercise 2: *Annotating the figure*

- Set the figure title

```
ax.set_title('my super cool plot')
```

- set the X and Y axis labels

```
ax.set_xlabel('my X')
ax.set_ylabel('my Y')
```

- Plot 2 functions each with a label (LateX is possible !)

```
ax.plot(x1,y1,label='$f_1$')
ax.plot(x2,y2,label='$f_2$')
ax.legend()
```

- Save the figure to a file. Open the generated file and check the content

Exercise 3: *Plotting analytic functions*

- For this, we want to evaluate a function at a bunch of x coordinates. Let us first generate these coordinates and store them in a numpy
- Then we can construct the function values to x. For instance, create another another numpy array which stores sinus value of x.
- Plot the sinus function.
- You can restrict the x and y plotting range

```
ax.set_xlim(0,1.)
ax.set_ylim(0,1.)
```

- Or by restricting the number of points (in a more Matlab way)

  ```
  ax.plot(x[:3],y[:3])
  ```

- One can use the algebra over numpy. Plot the $\sin^2$ function.

Exercise 4: ***Plotting data from a file***

- The file 'data.plot' can easily loaded in a numpy

  ```
  fdata = np.loadtxt('data.plot')
  ```

- This file contains three columns. The first column is the X axis values. The second column is an analytic prediction, and the third column is a measured data. Verify that it is correctly loaded by printing the shape of the vector
- Plot the analytic and measured curves on the same graph
- Plot directly the error the measure
- Plot the analytic prediction with error bars representing the shift of the measure

  ```
  ax.errorbar(fdata[:,0],fdata[:,1],np.sqrt((fdata[:,1]-fdata[:,2])**2))
  ```