TP 3 – Graphics

Exercise 1: We will draw the curves of the sine and cosine functions between -π and + π. The objective is to obtain the result presented in the Figure 1.

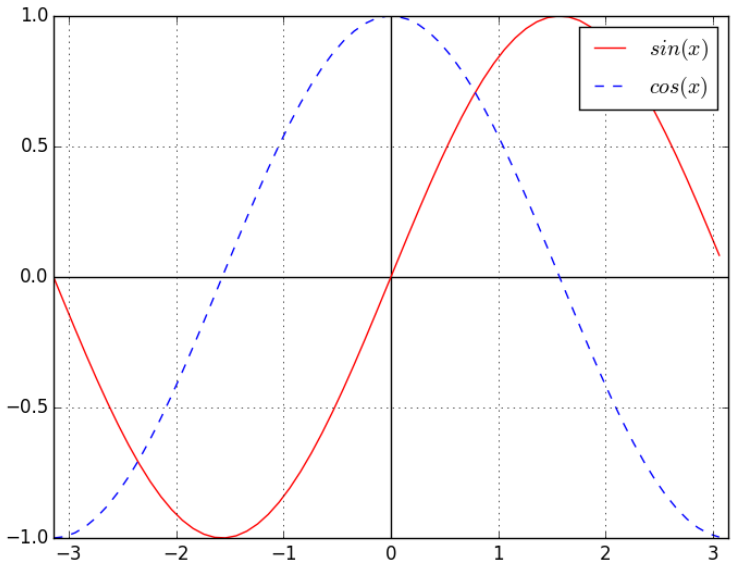


Figure 1

In a TP3-Ex1.py file:

1 - import the numpy and matplotlib.pyplot libraries as following:

```
import numpy as np

import matplotlib.pyplot as pp
```

2 - define the bounds in x of the figure (i.e. -π and + π) in the xstt and xend variables then in y in the ystt and yend variables. (example: xstt = -np.pi)

3 - define the precision of the figure in the form of a variable named step which will indicate the size of the interval separating two successive values of the values in x. (example: step = 0.1).

4 - generate the list of values of x in a variable named lx using the arange function of the numpy module. (Note: the variables xstt, xend and step previously defined to use in arange() function)

5 - generate the list of values of y in a variable ly using the sin () function of the numpy module. (Note: the sine function of the numpy library takes a value or a list of values as a parameter. If a list of values is given, a list of results is output from the call of the function np.sin (). It is possible to obtain this type of result with the standard libraries of python by using the function map (): lx = [0,0.1,0.2] ly = list (map (np.sin, lx))

6 - use the plot () method of matplotlib.pyplot to generate the curve we have just calculated.

7 - request the display of the curve using the show () function of the matplotlib.pyplot module.

8 - In a variable ly2 calculate the list of values obtained by numpy cos () function.

9 - add the curve defined by lx and ly2 using the plot function.

10 - add a legend (label / legend ()), display the grid (grid)

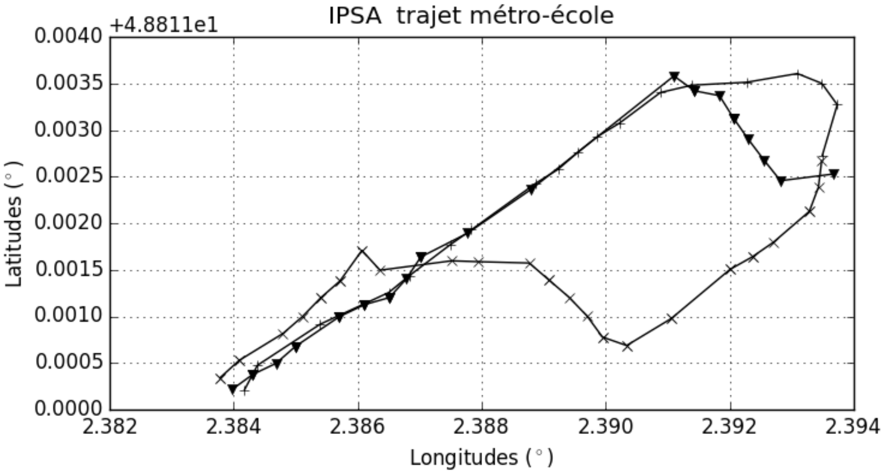
11 - use the axis () method of the matplotlib.pyplot module to define the limits of the rendering of the figure that we will generate. Use the previously defined xstt xend and ystt yend variables.

12 - use the savefig method to generate render files in svg, eps, png, pdf format.

Exercise 2: In this exercise we will perform the graphical representation of different GPS journeys using a projection equirectangular. In the TP3-Ex2.py file:

1 - Load data from CSV files into lists (you have the 3 routes in 3 csv files, import these 3 routes) and use these lists as sources of the curves of the graphic.

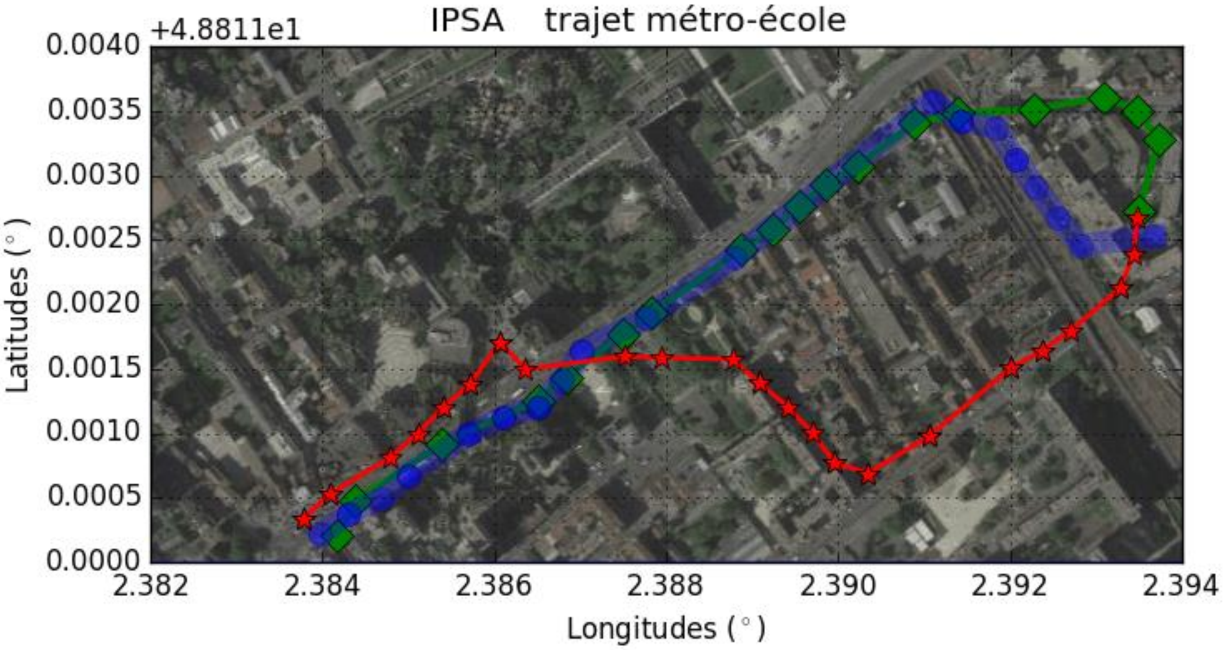
2 - Display the graph.



Exercise 3: The SatIvry.png image is a satellite photo in equirectangular projection on the zone defined between the coordinates 2.382,2.394, 48.811,48.815. Use the imread function to load this file into an image. Use this image and the extend parameter of the imshow function for add the image to the chart.

Please visit the given website for different parameter settings of matplotlib.pyplot.plot

https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html


IPSA trajet métro-école
+4.8811e1
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Latitudes (°)
2.382
2.384
2.386
2.388
2.390
2.392
2.394
Longitudes (°)