MAT101 – Programming with Python

Lorenzo Micalizzi

Institute of Mathematics, University of Zurich

NB: For all the matrix and vector structures use numpy.

Exercise 1

a) Code a function which takes as input a list of numbers l and a string s and returns as output another list v. If the string s is equal to "positive", it returns the list of the indices of the positive elements in the input list l; if the string s is equal to "negative", it returns the list of the indices of the negative elements in the input list l; for any other value of the string s the execution is stopped.

b) Code a function which takes as input a matrix M, a string s and a threshold t and returns as output another list v, which is a list of lists representing the indices of some elements of the matrix. If the string s is equal to "bigger", the function returns the list of the indices of the elements bigger than t in the input matrix M; if the string s is equal to "smaller", the function returns the list of the indices of the elements smaller than t in the input matrix M; for any other value of the string s the execution is stopped.

c) Code a function which takes as input a three-dimensional ndarray M and returns a three-dimensional ndarray Q with the same shape, whose general element with indices i, j, k is defined in the following way

$$Q_{i,j,k} = \begin{cases} \sqrt{M_{i,j,k}}, & \text{if} \quad M_{i,j,k} > 0.5 \\ M_{i,j,k}, & \text{if} \quad M_{i,j,k} \in [-0.5, 0.5] \\ M_{i,j,k}^2, & \text{if} \quad M_{i,j,k} < -0.5. \end{cases}$$

Exercise 2

a) Code a function with three inputs, a matrix M, a value x and a tolerance tol, which returns a new matrix Q with dimensions equal to the ones of the input matrix and the same elements up to the replacement of the elements $M_{i,j}$, which are close to x more than the given tolerance tol, with x.

b) Code a function which takes as input a user-defiend function f dependent on a real parameter x and a vector of abscissae a and returns as output the vector v of the evaluations of f in the elements of a.
NB: Not necessarily the user-defined function f can be used on vectors in the same way as the mathematical functions defined in numpy, therefore the evaluations of f must be performed one by one through a loop.

c) Given $a > 0$, the sequence $x_n = a^{\frac{1}{n}}$ with $n \in \mathbb{N} \setminus \{0\}$ converges to 1. Write a function which takes as input a and a tolerance ε and returns as output an integer k. If $a < 0$, then the function returns $k = -1$; if $a = 0$, then the function returns $k = 0$; if $a > 0$, then the function returns k equal to the first $n > 1$ such that $|x_n - x_{n-1}| \leq \varepsilon$.
Code the function using a while loop with a safety exit condition which stops the execution of the program based on the number of iterations.

Exercise 3

a) Write a script which, given $N = 5$, plots the functions x^j $j = 1, \ldots, N$ over $[0, 1]$ in the same plot with title, labels for the axes, a grid and a legend and saves the plot as "xj.png". Further, the script should produce a file named "my_columns.dat" in which the values of the plotted functions in $Q = 100$ equispaced abscissae in $[0, 1]$ are stored. The values of the functions should be saved in different columns, separated by commas and with a suitable header.

b) Write a script which opens the file produced in the previous point, reads the data in it and plots them (producing the same plot of the previous point).

   EXTRA: Try to rewrite the scripts for general $N, Q \geq 1$. This would be more complicated because if N is not fixed you cannot define a single vector of values for each function to be plotted and very likely you have to define a bidimensional array $Q \times N$. Also writing the header and the data in the file, as well as reading them, is a bit more cumbersome. Do not worry, in the exam you will have something in the spirit of the standard version. However, it is useful for you trying to solve this exercise :)