MAT101 Programming – Homework 6

Deadline: Monday, 13.11.2023, 22:00

Login to `https://w3.math.uzh.ch/my` with your UZH credentials to submit your solved exercises for grading. You can find more information on how to upload/submit your exercises on `https://wiki.math.uzh.ch/public/studentUpload`.

For submission, please upload **at most 1** Python file **per exercise**. You could even just upload 1 Python file for the whole exercise sheet. You can use comments and/or print statements to answer non-programming tasks.

This exercise sheet is entirely on NumPy, so if you have not already, install it now using `pip` or any python package manager of your choice. It is common practice to do

```
import numpy as np
```

and we will use this convention in this exercise sheet.

Exercise 1. **15** P.

This exercise is intended to make you familiar with the basic functions of NumPy and its `np.ndarrays`.

a) create two `lists`: `[1, 2, 3]` and `[1.68, 2.71, 3.14]`, convert them to `np.ndarrays`, and name them `array_11` and `array_12`. **3** P.

Now using `array_11` and/or `array_12` and built-in operators or NumPy-functions, how can you obtain the `np.ndarrays`, which when `printed` give the following output (without modifying the individual elements of the vector):

b) `array([2.68, 4.71, 6.14])` **2** P.

c) `array([1.68, 5.42, 9.42])` **2** P.

d) `array([0.2, 0.4, 0.6])` **2** P.

e) `array([2.8224, 7.3441, 9.8596])` **2** P.

f) `array([1. , 2. , 3. , 1.68, 2.71, 3.14])` **2** P.

g) `array([2. , 3. , 1.68, 2.71])` **2** P.

Exercise 2. **10** P.

This exercise is intended to show you how you can use NumPy for linear algebra.

a) use `np.arange()` to get `array_21` like `np.array([0, 1, 2, 3])` **2** P.

b) use `np.linspace()` to get `array_22` like `np.array([0, 1/3, 2/3, 1])` **2** P.

c) compute the euclidean norm of `array_22` **2** P.

d) compute the inner product of `array_21` and `array_22` **2** P.

e) compute the outer product of `array_21` and `array_22` **2** P.

Note: The inner product is also called scalar product or dot product.

Exercise 3. **15** P.

💻 This exercise is about some special functions in `Numpy` that come handy from time to time. First, define and array via `array_31 = array([-10, 0, 30, 2, 20, 0])`.

a) Find the indices of the maximum, minimum and zeros of the array using `np.where()` in three separate lines. **3** P.

Define an array called `array_32` the elements of which are 1000 draws of random uniform distribution from 0 to 1 i.e, [0,1).

b) Sort the array using `Numpy` and call it `array_33`. **2** P.

c) Using `np.savetxt()` save the sorted array in a text file called `sorted_33.txt` in the directory of your choice. **3** P.

d) Reload the saved array via `np.loadtxt()` into a new variable called `array_34`. **3** P.

e) Append the elements of the original unsorted array (`array_32`) at the end of it. Check whether the length of the final array is correct. **2** P.

f) Again, write the final array with the correct length in a new text file, `sorted_and_original_arrays.txt`. **2** P.

Upload the text files along with your solutions.