

## Information Processing: First Laboratory Exercise

Rivers are of great economic importance, from thermal power plants and nuclear power plants that need rivers to operate, public water supply to agriculture and transport. Although people today largely manage watercourses, weather conditions can lead to flooding, soil erosion and similar natural disasters. In order to be able to prepare for potential problems, it is necessary to investigate the behavior of the river based on multi-year monitoring of water levels. In this laboratory exercise, you will analyze the water levels of the Sava River over a period of 25 years.

Within the compressed directory with this notebook, a .mat file with data on the water level of the Sava River in the period from January 1st, 1982, to December 31st, 2007, is located. Each data row indicates one day of water level measurement.

The exercise is conducted in Python/Google Colab, and all the commands needed to conduct the exercise are given in the lecture. This laboratory exercise brings 3 points.

When you are done with the exercise, place a .pdf report (you can generate a .pdf report with the command File-Print-Location: PDF) on your *Moodle*.

1. Import libraries: NumPy, matplotlib.pyplot, scipy.fft, signal from scipy, and pywt.

```
# Place for your code.
```

2. Load water level information from the .mat file from the compressed directory. Loaded data are saved in dictionary. Extract the key and the corresponding water level data and reduce the dimension if necessary.

Tip: Import the .mat file from your Google Drive (place the .mat file within the Colab Notebooks folder on your Google Drive). Other tips can be found in the following code section.

```
# Place for your code.
```

```
from google.colab import drive
drive.mount('/content/drive')

#After running this code section, a link will appear in the window below. Click on the link, sign in with your Google account, and you will
#get the code. You need to copy the code and paste it into the output below this window (in the "Enter your authorization code: " window).
#After entering the received code, the message "Mounted at
#/content/drive" will appear.
#If you have the file "OIkobas.mat" saved in such a Colab Notebook
#directory, you can access it with the following lines of code:
#from scipy import io
#kobas_iz_matlaba=io.loadmat("drive/My Drive/Colab
#Notebooks/OIkobas.mat")
#print(kobas_iz_matlaba)
```

**3.** Print: number of days for which water level data are available, mean water level, standard deviation, minimum value, maximum value, and median.

# Place for your code.

**4.** Plot the water level. Mark the x and y axes and the title of the figure.

# Place for your code.

**5.** Calculate the discrete Fourier transform of a given water level. Print the obtained values. Plot the absolute value of the obtained coefficients.

# Place for your code.

**6.** Calculate the Short time Fourier transform of a given water level using a rectangular window, 2 years wide. Show the result using pcolormesh. Answer: what frequencies are visible in the water level? What years do they occur?

# Place for your code.

ANSWER: *Answer here to questions from Task 6.*

**7.** Use a different window (of your choice) and repeat the previous task. Print the width of the used window. Explain the differences in the obtained figures.

# Place for your code.

ANSWER: *Answer here to question from Task 7.*

**8.** List all continuous wavelet families in the PyWavelets library. In the same figure, using the subplot, plot two wavelet functions of your choice. Write which wavelet is shown in the title of each subplot.

# Place for your code.

**9.** Using a Morlet wavelet, determine the continuous wavelet transformation of a given water level. Plot the absolute value of the obtained coefficients. Answer: what frequencies are visible in the water level? What years do they occur?

# Place for your code.

ANSWER: *Answer here to questions from Task 9.*

**10.** Answer: explain the difference in the obtained STFT and CWT results.

ANSWER: *Answer here to questions from Task 10.*