Assignment 1 – ”Pulses”

Learning Outcomes Assessed

1. Write code using for loops, to work with sequences of values.
2. Refactor code into functions to allow re-use; factoring out constants.
3. Write code for reading text and CSV data from files.
4. Write code to visualize data with matplotlib.

Introduction

The data you will be working with has been taken with the neutron time-of-flight spectrometer TOFOR at the JET fusion reactor (picture to the left). Detector pulses are recorded with a multi-channel digitizer device (right).

Download the CSV file pulses.csv – it contains sensor readings for pulses from an experiment.

- Each row represents a single pulse, recorded with a detector.
- The first column is the timestamp of the pulse.
- The remaining columns are detector readings taken at 0.5 nanosecond intervals.
  The units are from an Analog to Digital Converter (https://en.wikipedia.org/wiki/Analog-to-digital_converter).

In this assignment, we will analyse this dataset.

Submission

You need to submit your solution as a Jupyter Notebook (file ending: .ipynb). Other formats will not be accepted.

The notebook should contain your code, the figures requested in the various questions, and short explanations of the procedures you used to get them.

Grading

Grade	Criteria
3 (Pass)	Substantially correct attempt at mandatory parts. Code quality satisfactory. Report including the required figures.
4	As 3, but good attempt, including some extended parts. Code quality is good, code commented. Report including the required figures and comments about how you created them.
5	As 4, but excellent attempt, including several extended parts. Code quality is excellent, code well commented. Report including the required figures and comments about how you created them and what they show.

We will make a judgement on the code quality. Good quality code (for this assignment):

- Uses correct naming conventions for variables, functions, constants, etc.
- Sensible names for variables, functions, constants, etc.
- Terse, well-written comments, where appropriate; and to indicate code relating to different sections. Correct spelling.
- Readable; avoids substantial, unnecessary complexity and repetition.
- No redundant or unused code.
- Comments for both individual lines and sections of code.

Mandatory Questions

Question 1

In Lab 1, we read data from a CSV file and "manually" handled the rows and columns – to practice working with arrays directly. For this assignment, begin by loading the CSV file as a numpy array using the `loadtxt(...)` function, described here:
https://numpy.org/doc/stable/reference/generated/numpy.loadtxt.html

Load the file and print the content.

Question 2

Convert the readings to voltages according to this formula (the ** means "raised to the power of" in Python):

```
Voltage (Volts) = (ADC Reading) / (2**10 - 1) * 0.6
```

Question 3

1. Plot a single pulse (i.e. row of the CSV file).
2. Add axis labels and a title.
3. Display your figure in your notebook.

Question 4 – *Baseline Correction*

The pulses typically have a flat section at the beginning, which then drops and later returns to the original level – which we call the *baseline*.

The baseline typically represents some noise level, so we will process our data to remove the baseline, in order to help us understand and visualize our data.

1. For each row/pulse:
    a. Find the mean (average) of the first 10 elements.
    b. Subtract this value from all values for that pulse.
2. Use a variable for the number of elements used for the mean.

Extended Questions

Question 5

Repeat Question 4, showing the values for a pulse with the baseline correction. Include this figure in your report.

Question 6

The amplitude of the pulse is related to the maximum value of the detected signal voltage, and the number of values different from the baseline to the length of the signal. The total area enclosed by the pulse (after baseline correction) corresponds to the energy deposited in the detector.

For each pulse, find the (baseline corrected) sensor reading maximum value, using `numpy.min()`.

For each pulse, add up the (baseline corrected) sensor reading values, using `numpy.sum()`.

Plot histogram of those values for all pulses, showing the distribution.

Use `matplotlib.pyplot.hist(...)`.
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

Compare the two distributions. Discuss similarities or differences in terms of physical properties as described above.