Final Project

BME 3053 Spring 2023

Objective

To gain practical experience of computer programming and design using MATLAB.

Introduction

A research project investigates the effect of exercise on the function of the heart. Measurements have been made using magnetic resonance imaging of the flow of blood out of patients' left ventricles during exercise over a number of heart beats. Researchers now wish to analyze the data to learn about how diseased hearts respond to the increased load experienced during exercise.

Instructions

The file *flow.csv* (which is provided for you) contains exercise flow data for one patient. These flow data cover a number of heart beats of the patient so they are roughly cyclic.

The file contains two columns of data containing the times and flow measurements for the patient. The time and flow values in each row are separated by a comma, i.e. the file is in *comma separated values* format.

You are required to write code to read a file in this format and then to analyze the flow data as described below.

- First read in the time and flow data and store them in appropriate variable(s).
- Then *smooth* the flow data to reduce the noise in the measurements. (*hint: see MATLAB documentation for the command* `smooth`).
- The smoothed flow data should then be divided up into individual heart beats. This will be done by searching for *peaks* and *troughs* in the data (details of these are given later). The start of each heart beat can be found as follows (see Figure 1):
  1. Find data points that are *peaks*.
  2. Search backwards in time from each peak to find a *trough*.
  3. The troughs are considered to be the starting points of the cardiac cycles (heart beats).
  4. Use the troughs to divide up the flow data into separate cardiac cycles (your code should ignore the last heart beat as it may be incomplete).
- For each heart beat, estimate and store the area under the flow data points (*hint: see the MATLAB documentation for the* `trapz` *command*). Compute and display to the screen the mean and standard deviation of the areas of all heart beats. Display the results to two decimal places.
- Produce a figure contain the following visualisations:
  - A plot of the original flow data, plotted against measurement time.

  - A plot of the smoothed flow data, indicating the locations of the starting points of the heart beats (i.e. the troughs).
  - A plot of the flow data for all the individual cardiac cycles on the same graph, with each one starting at the left of the plot.
  - A histogram of the areas under the flow data points for all heart beats.

  All plots should be suitably annotated.

Details for finding peaks and troughs are now given below (also refer to Figure 1).

- A *peak* must have the following properties:
  - A flow value greater than a threshold of 300.
  - A flow value greater than both its predecessor's and successor's values.
  - The predecessor's flow is greater than the flow two time points before.
  - The successor's flow is greater than the flow two time points after.
- How to find a *trough*:
  - Find a peak.
  - Search backwards in time from the peak (the flow values should start off by descending)
  - After some backward steps, the values may stop descending. In this case, we have located a trough.

  Note that we may not always find a trough for every peak.

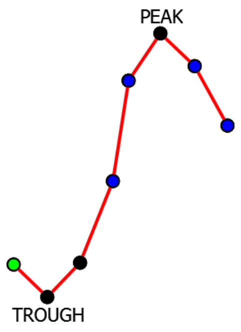

Figure 1: Finding peaks and troughs in flow data. Whether a measurement is at a peak depends on two data points before it and two data points after it (shown in blue). The trough is identified by searching backwards from the peak until we find a flow value that is higher than its successor (shown in green) - the trough is then at the value after this one (see text).

Expected output

The expected output for your program when run on the provided data file is shown in Figure 2. The mean and standard deviation output to the command window should be:

```
Mean/std dev of areas under curves = 97707.93 +/- 6438.12
```

Note that you should write your program in such a way that it will work for any input file with the same format as *flow.csv* file and with any number of lines.

Before starting any coding, you should produce a structured design for your program, using structure charts and/or pseudocode. You should then apply an incremental development approach to develop your code.

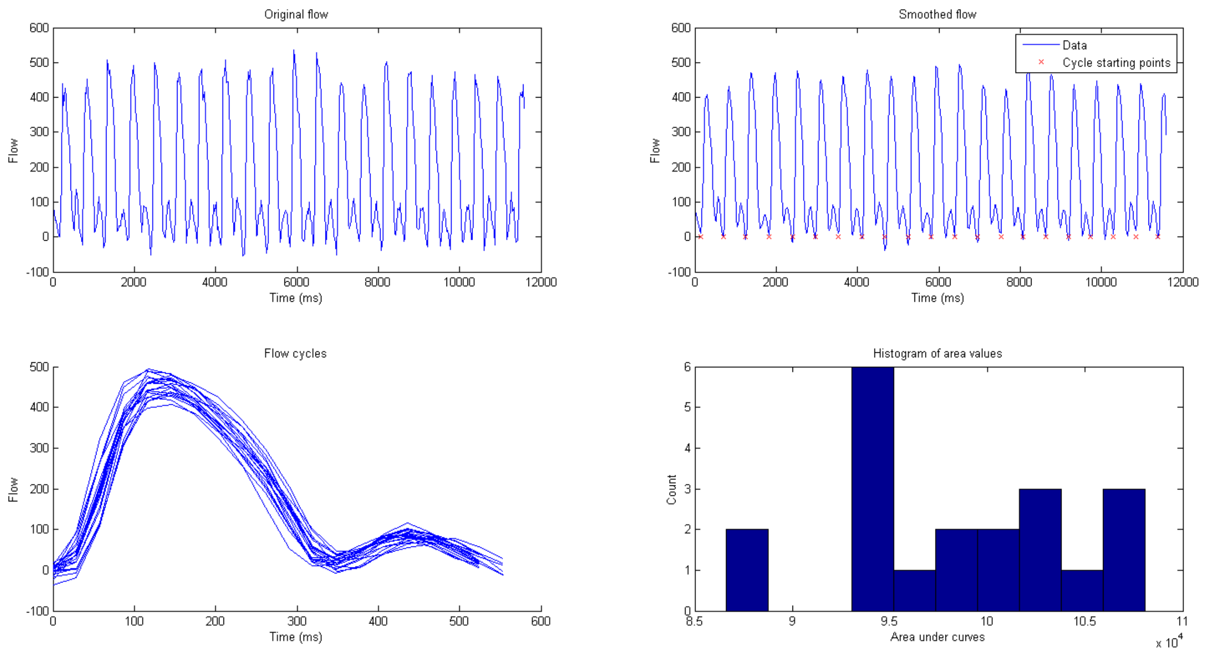

Figure 2: Expected output of the program when run on the provided data file.

There are many ways that this exercise can be tackled and a good approach is to break the tasks that you need to do down so that they can be implemented in multiple functions that are called by a single main function or script.