HW11-part2 (50 points)

These instructions are applicable for all scripts that you will write in python3.

```
Instructions for Python3 Homework submission

#class header

#Q1
var1=124

#Q2
var2=3+var1
```

```
The HW11-part2 is due Sunday April 21st 11:59pm

If you submit it by Saturday April 20th 11:59pm you will get 4 extra credit points.
```

Policy: Work on the HW problems on your own. It is not allowed to share code in any way. The use of chatGPT is not allowed. Using the internet to find out how to solve problems is not allowed.

General Grading rules we will follow:

- We do not give points for instructions but take points off for not following them.
- We take all points off for a question if your code has syntax errors (comment out the code for partial credit).
- We grade the code, not the output of the code. Even if the output is correct, hard-cording, redundant logic, code that has no purpose in the program will not earn full credits.
- We will take points off if you use out-of-class material.
- We will take points off if you do not follow directions when a question specifies to use a certain command or write the code in a certain way.

HW11-part2 (50 points)

1. In this exercise you will use NumPy, pandas, Matplotlib, and SciPy for fitting.

The data file **ExpData-Fluorescence.csv** contains experimental data on fluorescence lifetimes of fluorescent molecules, with two excited states. The first row is a header line.

The fields:

- 1st field reports time in ns
- 2nd field reports the fluorescence intensity.

The fluorescent molecules used in the experiment undergo a biexponential decay.
The equation for the biexponential decay is:

$$F = A\left((1-f)\,e^{\frac{-t}{\tau 1}} + f\,e^{\frac{-t}{\tau 2}}\right)$$

where F is the fluorescence intensity and t the time. A, f, τ1, and τ2 are all parameters.

In this exercise you will fit the data with this model, which is a non-linear model, plot the data and the model, and write results into a file.

In a script called **ex11-fluorescence.py** do the following:

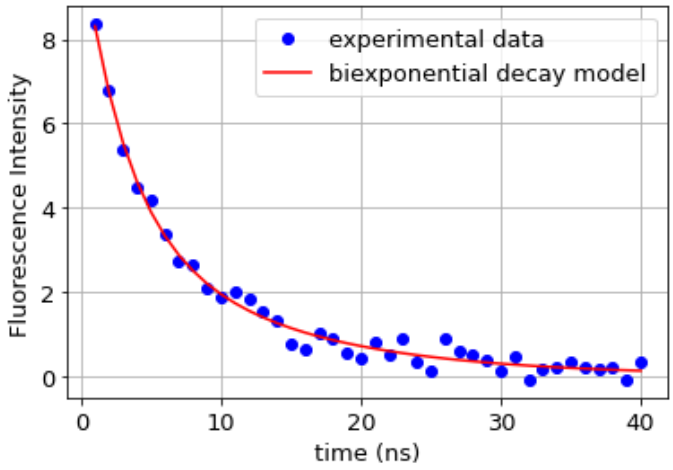
Q1 (5) Create the variables x and y, to store the data for the time and florescence, respectively, for the fit. Use NumPy loadtxt() only one time.

Q2 (12) Fit the data with the biexponential decay to estimate the fitting parameters:

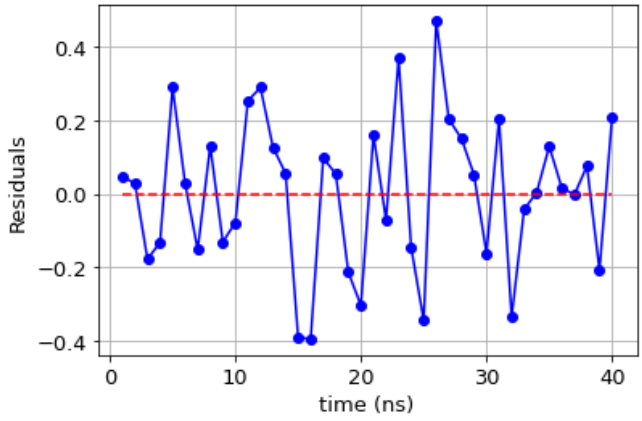
A - the amplitude
$\tau 1$ and $\tau 2$ the lifetimes
f – contribution to the decay

Use this initial guess for the parameters:
A = 10
f = 0.5
$\tau 1 = 2.5$
$\tau 2 = 10$

Q3 (10) Plot the data and the model. Label the axis, make the legend, set font size to 13, and add the grid to match the graph below.



Q4 (5) Evaluate the fit. Calculate and plot the residuals as a function of time.
Include the red dotted line.
Your plot should look like this:



Q5 (2) Write in a comment line – use triple double quotes - the reason why this model is a good model, based on the residuals.

Important Note: from Q6 to the end, the sequence of the numbers in the output depends on the sequence of the parameters you used in your function model (where you define the function). Here the sequence was the following: A, f, tau1, tau2. It is okay if your sequence is different, but the numbers should match.

Q6 (2) Print the fitted parameters to screen. Use a variable that has already been made.

```
[10.37783198  0.39222955  3.13039471 11.30544227]
```

Q7 (4) Calculate the standard errors of the fitted parameters and store them in a variable. Print it to screen.

```
[0.4490789  0.10756068 0.69556625 1.86769971]
```

Q8 (10) Now use pandas' **to_csv()** function and write in a file called **results.csv** the results of your fitting, in this format:

```
A,10.378,0.449
f,0.392,0.108
tau1,3.130,0.696
tau2,11.305,1.868
```

The first field is the name of the parameter, the second field the value of the fitted parameter, and the third field its standard error. Notice the numbers are formatted to 3 decimal places.

Use variables already made along with this provided variable.
`L=['A','f','tau1','tau2']`

If you use loops or comprehension, you will lose all points on this part.

Hint: Make a DataFrame from a 2D array and rename the rows.
The 2D array should contain the fitted parameters and their standard errors.
Below is the suggested DataFrame you should make:

```
              0         1
A     10.377832  0.449079
f      0.392230  0.107561
tau1   3.130395  0.695566
tau2  11.305442  1.867700
```

Upload to Gradescope HW11-part2:
ex11-fluorescence.py