UCF Physics PHZ 3150: Introduction to Numerical Computing
Fall 2021 Homework 11
Due November 18 2021

Goals: practice coding, fitting data and making functions in Python.

Problem 1 (**5 points**). Make a new folder named `hw11_<yourname>` under you `handin` folder. As always, your log is part of your homework. After each problem number, give your answer and the names of any files you are handing in for each problem. If you made a HW11 entry in your log in a prior session and want to change it, just copy it to the current (last) session, and edit there. We will grade the last entry only. All text related to one assignment should be in one entry, with the problems done in order.

Problem 2 (**15 points**) In a lab you measured the dataset y(x) with:

```
x= np.linspace(1,100,40) and
y = np.array([  7.7,  15.82307692,  23.94615385,  32.06923077,
40.19230769, 48.31538462, 56.43846154,  64.56153846,
72.68461538, 80.80769231,  88.93076923,  97.05384615,
105.17692308, 113.3 , 121.42307692, 129.54615385, 137.66923077,
145.79230769, 153.91538462, 162.03846154, 170.16153846,
178.28461538, 186.40769231, 194.53076923, 202.65384615,
210.77692308, 218.9, 227.02307692, 235.14615385, 243.26923077,
251.39230769, 259.51538462, 267.63846154, 275.76153846,
283.88461538, 292.00769231, 300.13076923, 308.25384615,
316.37692308, 324.5]).
```

You know that the best-fit model for this dataset is of the form `y_mod = a*x_mod+b`, with `x_mod = np.linspace(1,100,100)`, `0.1<a<10` and `0.1<b<15`. Create a code that tests different models (so different combinations of `a` and `b`) to find the best-fit model for dataset `y(x)`. The code should print the best-fit model parameters `a` and `b` with an informative statement.

Problem 3. (**35 points total**) Let's practice some function building with Python.

a) (**7points**) The wind chill factor gives us the temperature feel on a windy day and is given by: $WC = 35.74 + 0.6215 * T - 35.75 * V^{0.16} + 0.4275 * T * V^{0.16}$ (with T in Fahrenheit and V in mph]. Create a function `wind_chill` that takes as input the actual temperature on a given day and the wind speed and returns the wind chill temperature. Run it for a day with T = 20F and V = 55mph and print the result with an informative statement.

b) (**8 points**) The parallax of a star is defined as the apparent displacement of a star on the sky in the course of a half year due to the Earth's motion around the Sun. Using simple trigonometry you can show that $\pi \sim 1/d$, where π is the parallax in arcseconds and d is the distance of the star to the Earth in pc. Create a function `parallax_to_distance` that takes as input a dictionary with information on stars observed (their name and their parallax) and returns a dictionary with the names of the stars and their distances from Earth. Call the function for a dictionary containing the following observations:

Name of star	**Parallax (in milliarcsec)**
Betelgeuse	7.63
Antares	5.89
Spica	13.06
Proxima Cen	768

c) (**10 points**) Make a function `lists_to_dict` that takes as input two lists and combines them in a dictionary. The lists could be either containing strings or numbers, but they will be homogeneous (i.e., if the first element is a string all elements of the list will be strings). Your function should check which of the input lists contains the keywords for the dictionary (i.e., which list has the strings; **don't** assume list1 or list2 will be one kind or another) and use that as keywords and use the other list as the values. The function should then return the dictionary. Call the function for the following pairs:

- List_1 = [1, 3, 5, 10, 12, 14] & List_2 = ['watermelon', 'melon', 'apple', 'strawberries', 'cherries', 'oranges']
- List_1 = ['cars', 'motorcycles', 'bikes', 'skateboards'] & List_2 = [200, 150, 10, 67]

d) (**10 points**) Make a function `matrix_mult` that takes as input two matrices and multiplies them, if possible. The function should first check if the matrices can be multiplied (i.e., are their dimensions compatible?). If not, the function should give an error message with information about the dimensions of the matrices. If the matrices can be multiplied the function should proceed to do the matrix multiplication (you need to this manually, don't use numpy functions like matmul!) and return the matrix product of the two matrices. Call `matrix_mult` for:
A1 = np.array([[1, 2, 3], [4, 5, 6]]) & A2 = np.array([[2, 3] , [2 , 2]]) ;
B1 = np.array([[1, 0], [0, 1]]) & B2 = np.array([[4, 1], [2,1]]) and
C1 = np.array([[1, 0, 3], [0, 1, 4], [2, 1 , 8]]) & C2 = np.array([4, 1 , 5])

Problem 4(**10 points**). Prepare and submit your homework. Write what you will do to make and submit the zip file into your log. Don’t forget to also commit your finalized log and push it to GitHub. When satisfied, close the log, copy it to your homework directory, and run the commands to make and submit the zip file. Turn the file in on WebCourses.