UCF Physics PHZ 3150: Introduction to Numerical Computing
Homework 8
Due November 3 2022

Problem 1 (**5 points**). Make a new folder named hw8_<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 HW8 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. (**20 points**) In the early 17th century Kepler used observations of the motions of planets in our Solar System to derive his now famous laws of planetary motion. According to these laws, all planets in our Solar system move around in elliptical orbits with the Sun in one focus (1st law), their speed varies along the orbit with the further they are from the Sun the slower they move (2nd law) and their orbits are such that the square of their period is proportional to the cube of their (average) distance to the Sun (3rd law). Here you will test the validity of the 3rd law to planets of our Solar system and extrapolate it to constrain the orbital distance of three exoplanets from their parent stars.

In a file hw8_support_funct.py make a function called kepler_3rd(period) that gets as input the orbital period of a planet in years and returns the orbital distance of a planet to the Sun. This function should use the simple approximation for the 3rd law: $P^2 \propto \alpha^3$, since it will focus on planets of our Solar system. From Kepler's 3rd law P^2/α^3= constant, so you can deduce that $P_1{}^2/\mathrm{P}_2^2 = \alpha_1{}^3/a_2^3$. Use this equation in your function. Write an appropriate docstring.

Use the (P,α) properties of our Earth as the reference point (P1,α1). Remember that the period of our Earth is 1 year or 365.25 days and a_orb = 1 AU (1 Astronomical Unit ~ 150,000,000km or 92,967,000 miles).
Import the function in your main homework .py and calculate the distances of the other planets of our Solar system using the observed periods from the following table:

Planet	**Period [days]**
Mercury	87.96
Venus	224.7
Mars	686.97
Jupiter	4332.82
Saturn	10775.6
Uranus	30687.15
Neptune	60190.03

How do the values you get compare to the actual distances of the planets (0.4 AU ; 0.7 AU ; 1.524 AU ; 5.2 AU ; 9.6 AU ; 19.2 AU and 30.1AU) ? Make a plot that compares the real distances with the ones you retrieve from Kepler's law. Make sure you specify manually different symbols and colors for the two sets (predicted and real distances).

Bonus 2 points: Use the names of the planets as tick labels on the x axis.

Problem 3 (**20 points**) Do your homework(!) In Python.

a) (5 points) Choose any problem from a recent homework assignment or other assessment in another class that involves something you need to calculate or derive and that you can plot. State the class, section, assignment, and item number. State (or take a photo/ make a screenshot of) the problem. State (or take a photo/ make a screenshot of) your solution as given in that class (corrected by you).

b) (15 points) Now, code your solution and make and save one or more plots. Your code must use two or more of: control flow structures (conditional, loops, function calls), when possible, avoid loops and use NumPy array math (you'll get **bonus 5 points** for actively doing this, provided it is appropriate), broadcasting, string formatting, reading or writing data to/from a file, masks. However, everything you write should be natural and belong in the code.

Problem 4 (**15 points**) Graduation season is coming, and you need to buy presents for all your friends graduating! Surprising your friends with many little presents which you hide in one big box is more fun than getting them one big present. The logistics though, are hard. To prepare for that, you need to find out how much wrapping paper you will need for the boxes and how many presents you can fit in every box! Create a class `graduation_presents` that uses the dimensions `a,b,c` of a box to calculate the box's surface area and volume. Assume that your dimensions are in inches. The class should also know how much wrapping paper you have (in inches^2) and how many presents you want to fit in your box. Use a proper initiation function.

Your class should contain:

a) An initiation function
b) a function that returns the surface area of the box (named, e.g., `surface_area`),
c) a function that returns the volume of the box (e.g., `volume`),
d) a function that checks if you have enough wrapping paper to cover your box (`enough_paper`)
e) a function that checks if your box can fit all the presents (`fits;` assuming that the average volume of a gift is 25 inches^3) and

f) a function (`print_gift`) that combines all these information and lets you know (with print() statements) if you have enough wrapping paper to wrap your box, if your gifts fit in the box, or if nothing is right.

Test it for the following combinations:

a	b	c	`number_of_gifts`	`wrap_paper_available`
10	10	10	15	800
10	10	10	55	1000
100	20	10	55	300

Problem 5 (**10 points**). Prepare and submit your homework. Write what you did to make and submit the zip file into your log. Don't forget to commit your finalized notebook and push it to GitHub. When satisfied, close the log, copy it to your homework directory one last time, and make the zip file. Turn the file in on WebCourses.