Coding Exercise #3

OBJECTIVES

After completing this week's exercises, you will:

- Practice using functions
- Numerically solve ODEs
- Practice using for loops and lists
- Create plots using matplotlib

Tasks

1. Open Eclipse and find the **sample-code-solutions** repository under the Project Package Explorer window and **pull** from this repository. This should make the file "background_reading_3_2022.py" appear. Use this as you go through the Background Reading pdf, found on D2L under the Coding Exercise 3 folder, to review the basics that are relevant to this Coding Exercise.

2. Locate your **last name-first name** repository in Eclipse and **pull**. It is good practice to periodically pull from both the **sample-code-solutions** and **last name-first name** repos**,** especially before each commit, to avoid merge conflict errors.

3. Create a new .py file **in your last name-first name repository**. Name your file **coding_exercise_3.**

4. Review the Background Reading in **Coding Exercises > Code Exercise 3** on D2L.

5. Write a code that has the following functions (you may make additional functions as you please, but you **must** have the ones listed below to be eligible for full marks):

   - A function that sets up three coupled ODEs, which will be solved via the **solve_ivp** function in the **scipy** package (see Background Reading). The system that you will be solving the Yb-176/Lu-177 problem that was solved with the guess-and-checker in Assignment 2:

   $$^{176}\mathrm{Yb} \xrightarrow{^{1}n_0} {}^{177}\mathrm{Yb} \xrightarrow{\beta^-} {}^{177}\mathrm{Lu},$$

   where the initial amounts of the isotopes are: 100 g, 0 g, and 0 g, respectively.

   - A function that returns the initial number of atoms and the decay constant for an isotope, using the input of a generic string identifier as required by the **radioactivedecay** package. You can mirror (copy) the **initial_val_decay_const** function from the Background Reading, including the way that input is set up. Make any modifications you wish.

- A function that:

    - Returns the solution to the three ODEs using the **solve_ivp()** function and your previous two functions, i.e., returns $N_{Yb\text{-}176}$, $N_{Yb\text{-}177}$, and $N_{Lu\text{-}177}$ at t = input time (see the dense_output argument explanation in the Background Reading). Use 0 to 1000 hours as the integration range, or something similar as you see fit. Choose points to evaluate the ODEs at (e.g, every 10 h? 25 h? 200 h? as you see fit).

    - Shows and saves a png image of a plot of all the solutions of the three ODEs (plotted on the same graph) using the **matplotlib** package.

        - Name your plot image(s) **YbLu-177_Decay_YourName.png** and set the resolution to 300 dpi (see Background Reading example).

        - Ensure that your choice of the number of and interval between the t values at which you solved the system of ODEs results in a smooth shape to the scatter plots. If your plotted data have very abrupt increases or decreases with too few points in between, you need to decrease the intervals between the times at which the ODEs were solved.

        - You may find a log scale useful to plot all three isotopes together, or generating separate plots to show the behaviour of each species separately – it's up to you!

    - This entire third function will effectively take the free-form code (all the stuff that is not in functions) in the Background Reading #3 and clean it up into function form. This function will also call your previous two functions to use their output.
      Use lists and iteration to your advantage as much as possible.

      Add a print statement to the end of your script for the output of this function (the number of atoms of all three isotopes) at an input of t = 21 days.

      Hint: the solution for the number of atoms of Lu-177 at 21 days should correspond (roughly) to your solution for the atoms at 90% saturation of Lu-177 from Assignment #2.

- Use function docstrings to explain the basic purpose of your function and to provide some details on the input arguments, what is being returned, and if there are any side effects (see the Background Reading #2 pdf document for details on this).