In NPRE 247, a large part of your grade is earned with computational projects. These are intended to build your technical writing skills, tie together the lessons of the course, and hone your existing computational and problem solving skills.

This assignment is originally the work of Drs. Huff and Munk, and has been edited for this semester of NPRE 247

1 Problem: Three-component decay chain

Write a computer program to solve a three-component decay chain:

$$N_A \longrightarrow N_B \longrightarrow N_C$$

where

N_C is stable.

The program must solve this system **numerically** as well as **analytically**. The analytical solution should follow the discussion in your book. The numerical solution should use a forward difference approximation (explicit scheme). The main program should call functions/subroutines to do the following:

1. Read input file with the input data:

$$\begin{aligned}
\text{decay constants:}& \\
&\lambda_A \\
&\lambda_B \\
\text{initial conditions:}& \\
&N_A(t=0) = N_{A0} \\
&N_B(t=0) = N_{B0} \\
&N_C(t=0) = N_{C0} \\
\text{numerical parameters:}& \\
&\Delta t \\
&t_{final}
\end{aligned}$$

2. Perform the numerical solution of the equations for $0 < t < t_{final}$.
3. Perform the analytical solution of the equations for $0 < t < t_{final}$.
4. Write results to an output file. The output file should also contain the parameters read from the input file.
5. Read the output file and use it to plot the results.

2 Report Content

Submit a brief report with your results. See later section on the report format.

2.1 Part 1: Theory

Introduce the theory behind the simulations. Include the following information about your solution mathematics:

1. Show differential equations you are solving **[-10%]**.

2. Show analytic solution of the differential equations [-10%].

3. Show complete derivation of numerical solution [-10%].

4. Show complete derivation for the time of maximum N_B [-10%].

2.2 Part 2: Results

Report the results of your calculations with the following parameters.

$$
\begin{aligned}
t_{A,1/2} &= 2.53h \\
t_{B,1/2} &= 11.05h \\
t_{C,1/2} &= \text{stable} \\
N_{A0} &= 100 \\
N_{B0} &= 0 \\
N_{C0} &= 0 \\
t_{final} &= 60h
\end{aligned}
$$

For the numerical solution : Start with $\Delta t = 1h$ and keep reducing it by a factor of two until you get a reliable solution (solution does not change significantly with decreasing Δt).

Your results section should include the following results:

1. Plot the numerical solution for $N_B(t)$ vs. time for 3 different values of Δt (coarse, medium, fine), all of them on the same graph. Add the analytical solution on the same graph [-10%].

2. Plot numerical $N_A(t)$, $N_B(t)$, $N_C(t)$, and $N_A(t) + N_B(t) + N_C(t)$ as a function of time, all on the same graph, use a Δt that gives reliable solution [-10%].

3. Using the numerical solution, plot the time of maximum N_B vs. $\frac{1}{\Delta t}$ for several different Δt. Use the analytical solution to determine time of maximum NB, and add that value to the graph [-10%].

   Note: The numerical solution with large Δt might be unstable (oscillations between time-steps). If so, do not report such results. Continue reducing Δt until you get a physically realistic (smooth) solution [-10%].

3 Report Formatting

3.1 General

Please write a comprehensive, self-contained report.

- It must be computer generated,not hand written [-5%].

- Assignment must be submitted via Canvas by the due date, including the report [-10%], any code (scripts, IPython notebooks, etc) [-20%], a sample input [-20%] and sample output [-20%]. Additionally, include a README file (which may be plain text or a pdf) that explains how to use the code and sample inputs/outputs in sufficient detail that a third party (the grader) can reproduce your results [-5%].

- Input/output can be written in any format that is convenient for you (remember that your code must be able to read them, though). Input and output file should contain sufficient information such that 3rd person could understand what it contains [-5%].

3.2 Content Formatting

- Report should be self-contained, do not repeat assignment text, do not copy/paste the assignment itself **[-5%]**.
- Do not submit results as raw "column of numbers" data **[-5%]**.
- Do not include your source code in the report **[-5%]**.
- Snippets (small parts) of the source code are OK, if relevant. Consider using the LaTeX`minted` package for syntax highlighting, if you're using LaTeX.
- Do not include commands typed in the prompt (Python, shell, compiler, etc) **[-5%]**.
- Do not include extra plots/figures **[-5%]**.
- Additional figures that support the requested results are OK.
- Report length should be less than 10 pages, if you exceed 10 pages, you are probably doing something wrong, for example:
  - 1-3 pages for problem description, equations, analytic solution, derivations
  - 1-3 pages for problem 1 (problem description, results, discussion)
  - 1-3 pages for problem 2 (problem description, results, discussion)
- Obviously wrong solution **[-5%]**.
- Include well-formatted references and in-text citations **[-5%]**.

3.3 Formatting

- Cover page with your name, assignment title/number, course number, date **[-5%]**.
- Include page numbers, except on the title/cover page **[-5%]**.
- Report body has to start on page 1 (i.e., the title page is NOT numbered) **[-5%]**.
- Use portrait orientation **[-5%]**.
- Landscape for a single page with large table/figure is OK.
- Separate and clearly label each problem/exercise, so that it is easy to see where one ends and the other begins. For multiple parts of an exercise, separate each sub-exercise so it is easy to find **[-5%]**.
- Plots, figures, and their labels must be formatted to be visible, readable and differentiable on the printout **[-5%]**.
- Use only one font type and size for the main body of the report **[-5%]**.
- Use monospaced font (e.g. `Courier`) for computer programs, functions, scripts, etc printed in the report body. **[-5%]**.
- Do not use monospaced font for the report body **[-5%]**.
- Glaring formatting errors, random looking formatting **[-5%]**.

3.4 Equations

- Number each equation in a consistent way [-5%].
- Equations should be numbered to the right of the equation [-5%].
- Use notation consistent with the class lectures or textbook [-5%].
- Typeset equations properly (e.g. Equation Editor, LaTeX, MathType, etc.), do not type them as unformatted text or inject them as grainy images. [-5%].

4 Tables and Figures

- Number and label each table and figure in a consistent way [-5%].
- All figures should be captioned and should be referenced in the text.
- Use proper labels for plots, figures, tables – title, axis, legend, units, etc. [-5%].
- Table title should be above the table, figure title should be below the figure [-5%].
- Titles, legends, labels must be of sufficient size and quality to be easily readable [-5%].
- Make units (e.g. time) on plots/figures understandable to humans [-5%], for example:
  - if scale exceeds 100s of sec, change to min
  - if scale exceeds 100s of min, change to hours
  - if scale exceeds 100s of hours, change to days, etc...
  - If solution behavior is not visible on the plot because of the scale, make another plot with a different scale (or log scale) that clearly shows the solution behavior [-5%].
- Confusing y vs. x and x vs. y [-5%].
- Clearly label each numerical solution inside each figure and inside each tables with the value of Δt used for the numerical solution [-5%].
- Use sufficiently high quality figures such that they look smooth and sharp [-5%].
  - Screen shots are probably too low quality.
  - `jpeg` and other lossy compression types are probably too low quality.
  - High resolution and lossless vectorized image types are recommended.

5 Programming

- Your program should be clear and readable [-5%].
- Include enough files for your program to run succesfully [-5%].
- Test your programs/functions thoroughly, after you finished testing, test it some more!
- Your program should be written using Python

6 Other

The purpose of the assignment is a comprehensive, self-contained, consistently formatted report and a demonstration that you understand decay calculations. The emphasis is not the programming itself. If you are not sure about what and how much to include in the report, imagine that you have to grade it – make it concise and easy to follow. I am being picky because I want you to write good reports. The content and formatting rules are *almost* universal.