NPRE 247 Spring 2024
Computational Project 2 Due 2024.04.26

Note: This project is adapted for use in this class from a computer project used in previous iterations of this course, taught by Drs. Kathryn Huff and Madicken Munk.

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.

1 Problem: Neutron balance eigenvalue/eigenvector

Write a computer program to calculate the eigenvalue/eigenvector of a matrix using the power iteration method. The main program should do the following:

1. Read cross-section data from an input file.
2. Calculate Migration and Fission matrices.
3. Formulate neutron balance as eigenvalue/eigenvector problem: $A\phi = k\phi$
4. Calculate all eigenvalues/eigenvectors using computer software (e.g. python's `numpy.linalg` package).
5. Calculate the largest eigenvalue and the corresponding eigenvector using power iteration.
6. Write results to an output file that can later be used to plot the results. The output file should also contain the parameters read from the input file.

2 Report Content

Submit a brief report with your results. See later section on the report format. Introduce the theory behind the simulations. Include the following information about your solution mathematics and report the results of your calculations for the following three parts.

2.1 Part 1: Theory

1. Show full derivation of the two-group neutron balance in infinite space **[-10%]**.
2. Show two-group Migration and Fission matrices **[-10%]**.

2.2 Part 2: Two group eigenvalue/eigenvector solution

1. Show Migration and Fission matrices with numerical values **[-10%]**.
2. Calculate all eigenvalues/eigenvectors of the neutron balance analytically by "hand calculation" - show complete derivation and the results **[-10%]**.
3. Calculate all eigenvalues/eigenvectors of the neutron balance using computer software (e.g. python's `numpy.linalg` package). - show results **[-10%]**.
4. Calculate the largest eigenvalue and the corresponding eigenvector using power iteration - show results after 2 iterations **[-10%]**.

2.3 Part 3: Eight group eigenvalue/eigenvector solution (EXTRA CREDIT)

1. Show full derivation of the multi-group neutron balance (ie, for an arbitrary number of groups, N) in infinite space **[+3%]**.

2. Show Migration and Fission matrices with numerical values **[+3%]**.

3. Calculate all eigenvalues/eigenvectors of the neutron balance using computer software - show results **[+3%]**.

4. Calculate the largest eigenvalue and the corresponding eigenvector using power iteration - show results after 2 iterations **[+3%]**.

3 Data

Group	Σ_a	$\nu\Sigma_f$	χ
1	0.0092	0.0046	1.0000
2	0.0932	0.1139	0.0000

Table 1: Two-group data from NEACRP L336, assumes homogeneous UO_2 composition.

To (row) ↓	**From (col)** →	1	2
1		1.0000	0.0000
2		0.0202	2.0000

Table 2: Two group scattering cross sections $\Sigma_{col \to row}$.

Group	Σ_a	$\nu\Sigma_f$	χ
1	0.0056	0.0134	0.3507
2	0.0029	0.0056	0.4105
3	0.0025	0.0011	0.2388
4	0.0133	0.0067	0.0000
5	0.0473	0.0220	0.0000
6	0.0180	0.0222	0.0000
7	0.0558	0.0897	0.0000
8	0.1798	0.2141	0.0000

Table 3: Eight group data from VENUS-2, UO_2 3.3% composition, xs_pin2-8g-LF

To (row) ↓	**From (col)** →	1	2	3	4	5	6	7	8
1		0.1179	0 0	0	0	0	0	0	0
2		0.0530	0.1949	0	0	0	0	0	0
3		0.0301	0.1159	0.5868	0	0	0	0	0
4		0.0001	0.0005	0.0769	08234	0	0	0	0
5		0	0	0.0019	0.1961	0.8180	0	0	0
6		0	0	0	0.0050	0.1737	0.6902	0.0023	0
7		0	0	0	0.0007	0.0246	0.2707	0.8626	0.0275
8		0	0	0	0.0001	0.0073	0.0550	0.3589	1.9761

Table 4: Eight group scattering cross sections $\Sigma_{col \to row}$.

4 Power Iteration

Begin with conditions:

$$k_1 = 1$$

$$\phi_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Iterate with the following definitions:

$$\phi_{i+1} = \frac{B\phi_i}{\|B\phi_i\|_2}$$
$$k_{i+1} = \frac{\left(B\phi_{i+1}\right)^T \phi_{i+1}}{\left(\phi_{i+1}^T \phi_{i+1}\right)}$$

Note the definition of the euclidean norm (a.k.a the L^2 norm) for a vector $\vec{x} = (x_1, x_2, \cdots, x_n)$ is :

$$\|\vec{x}\|_2 \equiv \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

Additionally, Khan Academy has entire course on linear algebra, with sub-section on eigenvalue/eigenvector. For those who need a review of eigenvalue/eigenvector problems, we can recommend the intro to eigenvalues and eigenvectors, as well as the example for the 2x2 matrix `https://www.khanacademy.org/math/linear-algebra/alternate-bases#eigen-everything`.

5 Infinite Medium, Two-Group Neutron Balance

Balance equation for group g:

$$\begin{gathered} \text{Loss} = \text{Gain} \\ \text{Absorption} + \text{Outscattering} = \text{Fission} + \text{Inscattering} \\ \Sigma_{ag}\phi_g + \Sigma_{g\to g'}\phi_g = \frac{\chi_g}{k}\left(\nu\Sigma_{fg}\phi_g + \nu\Sigma_{fg'}\phi_{g'}\right) + \Sigma_{g'\to g}\phi_2 \end{gathered}$$

Group 1 (fast):

$$\Sigma_{a1}\phi_1 + \Sigma_{1\to 2}\phi_1 = \frac{\chi_1}{k}\left(\nu\Sigma_{f1}\phi_1 + \nu\Sigma_{f2}\phi_2\right) + \Sigma_{2\to 1}\phi_2$$

Group 2 (thermal):

$$\Sigma_{a2}\phi_2 + \Sigma_{2\to 1}\phi_2 = \frac{\chi_2}{k}\left(\nu\Sigma_{f2}\phi_2 + \nu\Sigma_{f1}\phi_1\right) + \Sigma_{1\to 2}\phi_1$$

We can rewrite these equations in matrix form:

$$\begin{bmatrix} \Sigma_{a1} & 0 \\ 0 & \Sigma_{a2} \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} + \begin{bmatrix} \Sigma_{1\to 2} & 0 \\ 0 & \Sigma_{2\to 1} \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \frac{1}{k}\begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix}\begin{bmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} + \begin{bmatrix} 0 & \Sigma_{2\to 1} \\ \Sigma_{1\to 2} & 0 \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

Moving the inscattering to the left side of the equation, the equation becomes:

$$\begin{bmatrix} \Sigma_{a1} & 0 \\ 0 & \Sigma_{a2} \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} + \begin{bmatrix} \Sigma_{1\to 2} & 0 \\ 0 & \Sigma_{2\to 1} \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} - \begin{bmatrix} 0 & \Sigma_{2\to 1} \\ \Sigma_{1\to 2} & 0 \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \frac{1}{k}\begin{bmatrix} \chi_1\nu\Sigma_{f1} & \chi_1\nu\Sigma_{f2} \\ \chi_2\nu\Sigma_{f1} & \chi_2\nu\Sigma_{f2} \end{bmatrix}\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

Here, we can define the macroscopic absorption, inscattering, outscattering, and fission cross-section matrices:

$$\text{Absorption Matrix } (\mathbf{A}) := \begin{bmatrix} \Sigma_{a1} & 0 \\ 0 & \Sigma_{a2} \end{bmatrix}$$

$$\text{Inscattering Matrix } (\mathbf{S_{in}}) := \begin{bmatrix} 0 & \Sigma_{2\to1} \\ \Sigma_{1\to2} & 0 \end{bmatrix}$$

$$\text{Outscattering Matrix } (\mathbf{S_{out}}) := \begin{bmatrix} \Sigma_{1\to2} & 0 \\ 0 & \Sigma_{2\to1} \end{bmatrix}$$

$$\text{Fission Matrix } (\mathbf{F}) := \begin{bmatrix} \chi_1\nu\Sigma_{f1} & \chi_1\nu\Sigma_{f2} \\ \chi_2\nu\Sigma_{f1} & \chi_2\nu\Sigma_{f2} \end{bmatrix}$$

And now our equation is reduced to:

$$[A + S_{out} - S_{in}] \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \frac{1}{k} [F] \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

So, the final version is:

$$k \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = [A + S_{out} - S_{in}]^{-1} [F] \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

This gives us the final definition to note, the migration matrix:

$$\text{Migration Matrix:} = [A + S_{out} - S_{in}]$$

For the two group problem, let the eigenvector be $\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}$, let the eigenvalue be k, and calculate the eigenvalue and eigenvector of $[A + S_{out} - S_{in}]^{-1} [F]$.

6 Report Formatting

6.1 General

Please write a comprehensive, self-contained report.

- It must be computer generated, not hand written **[-5%]**.
- PDF must be submitted via Canvas by the due date **[-5%]**.
- Submit computer program via Canvas (if your code is in a Github repository, you may submit a link to that repository (make sure it is public)) **[-20%]** including sample input **[-20%]**, sample output **[-20%]**, and README **[-5%]** by the deadline.
- Input/output can be written in any format that is convenient for you. Input and output file should contain sufficient information such that 3rd person could understand what it contains **[-5%]**.

6.2 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 part 1 problem description, equations, derivation of solutions
  - 1-3 pages for part 2 problem description, results, discussion
  - 1-3 pages for part 3 problem description, results, discussion
- Obviously wrong solution [**-5%**].
- Include well-formatted references [**-5%**]

6.3 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 [**-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 part of the 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. [**-5%**].
- Do not use monospaced font for the report body [**-5%**].
- Glaring formatting errors, random looking formatting [**-5%**].

6.4 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%**].

7 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.

8 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!
- Use Python for programming and plotting.

9 Other

The purpose of the assignment is a comprehensive, self-contained, consistently formatted report and a demonstration that you understand neutron balance 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'm being picky because I want you to write good reports. The content and formatting rules are *almost* universal.