Homework 4.

SUBMISSION INSTRUCTIONS. **Please pay attention to these as they will make grading your assignments less burdensome.**

1. Put all of your code files in a folder called <your first name>_<your last name>_homework_4. For example I would call mine

andrew_osborne_homework_4

2. Compress your folder using the zip tool of your choice.

3. Upload your HW to Canvas.

INTRODUCTION.

In this assignment you will develop a piece of code to compute groupwise cross sections using the Bondarenko method.

BACKGROUND.

In class, you learned that when you collapse continuous-energy cross-section data into multigroup cross-sections, you need to account for self-shielding effects. For a reaction of interest:

$$\sigma_i^g = \frac{\int_{E_g}^{E`_g} \frac{\sigma_i(E)dE}{\sigma_i^t(E)+\sigma_0^i(E)}}{\int_{E_g}^{E`_g} \frac{dE}{\sigma_i^t(E)+\sigma_0^i(E)}} \quad (1)$$

where

$\sigma_i^g \equiv$	Self-shielded microscopic cross section [barns] for material *i* and energy group *g*.
$\sigma_i(E) \equiv$	Microscopic cross section [barns] for material *i* at energy *E*.
$\sigma_i^t(E) \equiv$	Microscopic total cross section [barns] for material *i* at energy *E*.
$\sigma_0^i(E) \equiv$	Microscopic background cross section [barns] for material *i* at energy *E* (see below).
$E_g \equiv$	Energy of the lower bin edge of energy group *g*.
$E`_g \equiv$	Energy of the upper bin edge of energy group *g*.

The background cross section $\sigma_0(E)$ (without escape correction) is given by:

$$\sigma_0^i(E) = \frac{1}{N_i}\sum_{j\neq i} N_j \sigma_j^t(E) \qquad (2)$$

where N_i, N_j refer to the atomic densities [atoms/b-cm] of materials i and j. E.g. if one wanted to compute the self-shielded (n,g) cross section of ^{238}U in fresh LWR fuel, N_i would represent the atomic density of ^{238}U, and N_j would refer to the atomic densities of ^{235}U and ^{16}O in turn. When the background cross section $\sigma_0^i(E)$ is small, self-shielding effects are large. When it is large, this implies that the concentration of nuclide i is small, and in the limit where $\sigma_0^i(E)$ goes to infinity, the self-shielded cross groupwise cross section is then for the "infinitely dilute" case. The infinite dilute cross sections are the raw cross sections from e.g. ENDF data files.

The outcomes of this assignment are to:

1) Create a 100-group self-shielded absorption cross section for ^{238}U in a cylinder of 10% enriched uranium using the Bondarenko method.
2) Verify your answer to 1) by simulating the cylinder in Serpent, and configuring Serpent to print out the 100-group cross section.

To help you complete this assignment, the instructor has provided a data file data_10000g_15S0.mat, containing the variables:

Ef :	Array containing the lower and upper energy values of 10,000 energy groups [MeV].
reacNames :	Cell array containing the names of the reactions included in the data file.
reacU235:	10,000-group **microscopic capture and total** cross sections of ^{235}U **(infinite dilution)** at **300K**. This is a 3-D array, where the first index refers to the energy group, the second refers to the reaction (capture, total), and the third is the background cross section. In this assignment you will only use the first element of the 3rd index, reacU235(:,:,1). See Fig. 1.
reacU238:	10,000-group **microscopic capture and total** cross sections of ^{238}U **(infinite dilution)** at **300K**. This is a 3-D array, where the first index refers to the energy group, the second refers to the reaction (capture, total), and the third is the background cross section. In this assignment you will only use the first element of the 3rd index, reacU238(:,:,1). See Fig. 2.
S0:	Background cross section.

The instructor has also provided a data file data_100g_15S0.mat, which is identical to data_10000g_15S0.mat, except for 100 groups instead of 10,000.

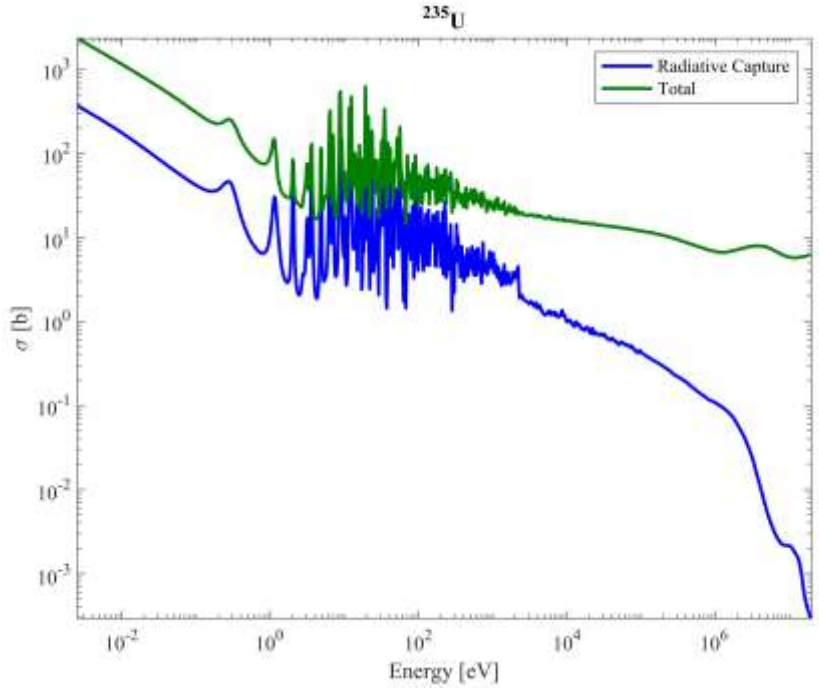

Figure 1. Cross-sections of ^{235}U.

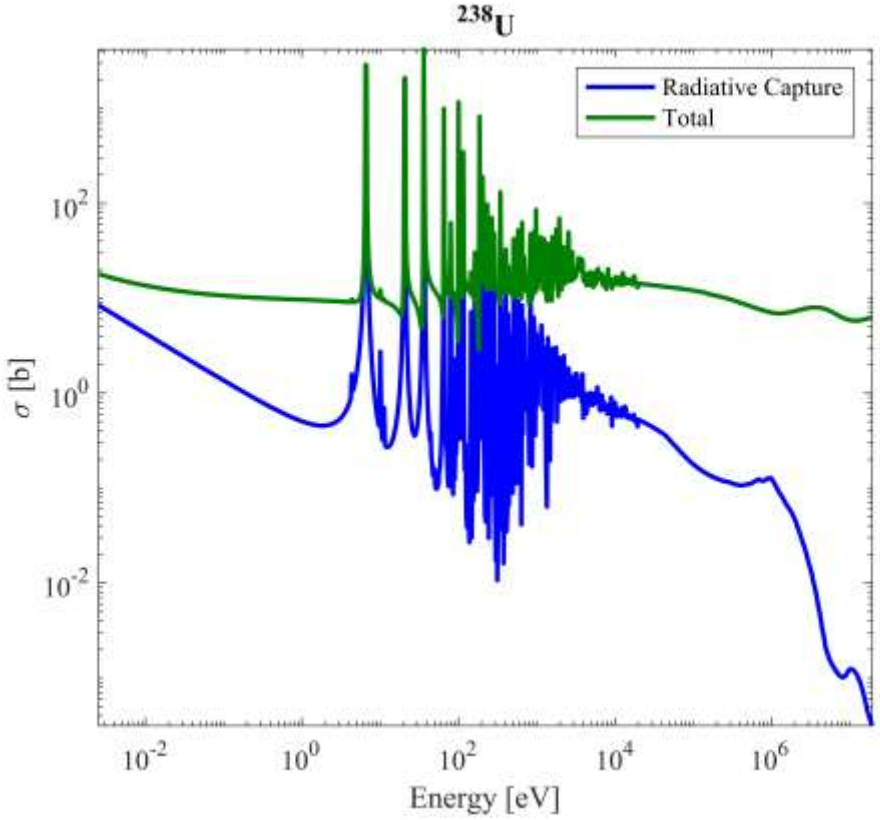

Figure 2. Cross-sections of ^{238}U.

Part 1. Write some code that will:

a) Compute the quantities of ^{238}U and ^{235}U, **in units of atoms per barn-centimeter,** in a cylinder of **pure** uranium enriched to 10 w/o. Assume the uranium has a density of 19 g/cm^3.
b) Compute the background cross section of ^{238}U for the 10,000 energy groups using Eq. (2), your answer to a) and the data provided.
c) Plot b) as a function of energy. You can use the mid-point of each bin as the ordinate. Your plot should be formatted in the same manner as Figs. 1 and 2.
d) Use Eq. (1) to collapse the 10,000-bin capture cross section of ^{238}U to a 100-group self-shielded cross section. Plot the original 10,000-bin cross section and 100-group self-shielded cross section vs. energy on the same plot. Your 100-group energy structure should correspond to the midpoint of each 100-bin fetch in the original 10,000-bin energy structure.

Part 2. Use interpolation on NJOY-generated self-shielded cross sections to cross-check your result.

BACKGROUND. Although Eq. (1) can be used to compute self-shielded cross sections, for reasons of performance it is more common to use a nuclear data processing code such as NJOY to do this. The approach most commonly used is as follows. Let us assume a 100-group cross section is desired:

- Pre-generate self-shielded 100-group cross sections for a range of background cross section values (typically 8 to 15 background cross section values from 10^{10} barns to 0.1 barns).
- Using the infinite dilution 100-group cross sections from 1), compute the 100-group background cross-sections for the isotope of choice.
- At each of the 100 energy groups, interpolate between the pre-calculated self-shielded cross sections, with the intent to find the self-shielded cross section in-between the pre-calculated values.

Tasks for Part 2:

a) Generate a 100-group background cross section for ^{238}U using the pre-generated 100-group cross section values data in data_100g_15S0.mat.
b) Generate a 100-group self-shielded microscopic capture cross section for ^{238}U using the pre-generated self-shielded 100-group cross section data in data_100g_15S0.mat and your answer to a). Use interpolation to do this.
c) Compare your answer to b) with the answer you got in Part 1d).

TIPS:
1) Work in groups.
2) Make good use of the TAs.
3) Use the interp1 function in Matlab.