Capstone Project Report

Introduction

The aim of this project was to effectively model a fission nuclear reactor using python. The general structure is as follows. Beginning with a set number of neutrons, a Monte Carlo simulation would be run to determine the number of fission, scattering, absorption, and escaped neutron events. Determining which events happen utilizes the cross-sections of the different materials in the reactor. A cross-section is a measure of area and can be used to determine the mean free path or the probability of an interaction happening, dependent on the particular elements involved and the energy of a given neutron. Using the cross-sections, cumulative density functions can be found which are used in the simulation.

With the number of different events, several interesting physical properties of the reactor can be calculated. Primarily, whether the reactor produces as many neutrons as it consumes; i.e. is the reactor critical. Other results include (but are not limited to) the number of antineutrinos the reactor produces, how much power the reactor produces, or how the criticality changes as the concentrations of materials are changed in the reactor.

Implementation

Our Process

Reaching our final desing took several iterations, as multiple hurdles had to be overcome before we reached a workable model. This process saw us first simulating a single neutron, turning this model into a crude Monte Carlo. As we worked towards a more polished model, we split our code into several interacting modules. These modules would first spawn a set number of neutrons into the simulation, process their interactions based on their cross-sections (probabilities), and produce counts of each event which occurred. These three modules, in order, may be referred to as the 'neutrons' modeule (initialize_neutrons), the 'density' module (density_functions), and the 'escape' module (check_escape). Finall a fourth 'master' file would be used to bring each of these pieces together by initializing the reactor size, number of neutrons, the particular elements involved, and then running each of the modules.

References

[1] "Delayed neutrons." [Online]. Available: https://ec.europa.eu/programmes/erasmus-plus/project-result-content/fbcfee8c-7b46-4e39-a5eb-129ea94d51c7/CTU1_Delayed_neutrons_part_2_Experiment_procedure_for_students.pdf

[2] J. A. E. Agency. [Online]. Available: https://wwwndc.jaea.go.jp/cgi-bin/Tab80WWW.cgi?/data/JENDL/JENDL-4-prc/intern/B010.intern

[3] OpenMC. [Online]. Available: https://docs.openmc.org/en/latest/methods/neutron_physics.html

[4] O. Andersen, "Uranium 238." [Online]. Available: https://www.sciencedirect.com/topics/pharmacology-toxicology-and-pharmaceutical-science/uranium-238#:~:

text=Uranium%20(U%2C%20atomic%20number%2092%2C%20in%20group%205%20of,is%20a%20naturally%20occurring%20actinide.

[5] D. Bogart, “Boron cross sections as - a source of discrepancy.” [Online]. Available: https://ntrs.nasa.gov/api/citations/19660018445/downloads/19660018445.pdf

[6] foxneSsfoxneSs, “Local (?) variable referenced before assignment.” [Online]. Available: https://stackoverflow.com/questions/11904981/local-variable-referenced-before-assignment

[7] H. Kim, “Neutron cross section evaluation of u-238.” [Online]. Available: https://t2.lanl.gov/fiesta2017/Talks/Kim.pdf

[8] W. McDowell. [Online]. Available: https://slideplayer.com/slide/8428135/

[9] username4567, “Calculate inverse of a function–library.” [Online]. Available: https://stackoverflow.com/questions/15200560/calculate-inverse-of-a-function-library

[10] Sep 2024. [Online]. Available: https://en.wikipedia.org/wiki/Neutron_transport