Maria Pinilla
1207 Kearney St. Apt. 3
Manhattan, KS 66502

December 19, 2014

Dr. Jeremy Roberts
137D Ward Hall
Manhattan, KS 66506

Dear Dr. Roberts:

Please find enclosed my manuscript, “Final Burnup Calculator for Multibatch Core Loading,” which I would like to submit to you as part of the course requirements of ME 701.

This paper outlines an overview of the theory behind multibatch core loading and why it is important for nuclear reactor fuel management. I also describe the application I have created to calculate and compare the final burnup from a multibatch or a single-batch core loading cycle. The application models an infinite core with three user selected parameters: number of batches, fuel enrichment, and type of burnable poison. The user can select to view three types of plots and see the calculated fuel efficiency displayed under the plot. My hope is that this can be a useful tool to those working with or studying in-core fuel management.

I thank you for your consideration and look forward to your decision.

Sincerely,

Maria Pinilla

Maria Pinilla

Proceedings of ME 701 (Development of Computer Applications in Mechanical Engineering)
December 19, 2014, Manhattan, KS

FINAL BURNUP CALCULATOR FOR MULTIBATCH CORE LOADING

Maria Pinilla
Nuclear Engineering
Kansas State University
Manhattan, Kansas, USA

ABSTRACT

In order to increase fuel efficiency and reduce cost, nuclear power plants utilize a multiple-batch core refueling process. I created a graphical user interface to calculate and plot the increase in fuel efficiency provided by increasing the number of batches in a full refueling cycle. I used PyQt4 to create this simple and intuitive application which models an infinite core using nuclear data from Poropy software. Will add good/bad results when program is fully functional.

INTRODUCTION

In nuclear reactor operations, it has become incredibly important to optimize core fuel management for many reasons. These include the cost of the fuel, since it cannot be reprocessed or bred in reactors, and the limited on-site spent fuel storage capabilities of most plants.

Multibatch core loading is a technique in refueling, where a portion of the core is replaced with new fuel periodically, which greatly increases the total burnup of the core for the same amount of fuel. This technique reduces the enrichment requirements for criticality, lowering operation costs and helping to maintain good safety standards.

For this project I my intent was to create a graphical user interface (GUI) that computes and plots the fuel burnup for a multibatch core. I used PyQt4 in Python to create a GUI and a software package called Poropy to obtain the data used for the calculations.

THEORY

The infinite multiplication factor K, also denoted by K_{inf}, is the ratio of neutrons produced by fission in one generation to the number of neutrons lost through absorption in the previous generation. K_{inf} is a measure of the increase or decrease in neutron flux for an infinitely large reactor, where there is no neutron leakage. Criticality is defined as K = 1, that is the number of neutrons in the reactor maintains constant from one generation to the next. When K is less than 1, the reactor is said to be subcritical and each generation thereafter would have less neutrons than the previous one. When K is greater than 1, the reactor becomes supercritical, generating more and more neutrons with time.

Reactivity is an important measure that describes the behavior of the reactor when K deviates from criticality, it can be calculated using Equation 1. Just like the infinite multiplication factor, K, reactivity is proportional to the fuel enrichment, which diminishes almost linearly as a function of burnup, i.e. time.

Equation 1 → reactivity = (k-1)/k

When a reactor core is refueled, it is done so in batches. A batch is a group of fuel elements of the same characteristics that enters or leaves the core at the same time. By increasing the number of batches within a time period, the total energy output of the fuel can be greatly increased. While it is important to get the most fuel efficiency possible, it is critical to keep in mind the costs of refueling a nuclear reactor and attempt to balance the two.

To understand the how multiple batches affect the reactivity and total burnup we can look at a two-batch core. The final burnup is “1/3 more than that obtained in a one-batch core for the same initial reactivity,” or the same initial enrichment [1]. This can be explained by the negative reactivity the burned fuel has in relation to the fresh fuel inserted on the second batch. The burned fuel acts as a neutron sink, which removes excess neutrons from the fresh fuel. This decreases the necessity for poisons or other means of compensating for the excess reactivity that fresh fuel has.

The final burnup is equal to the average of the sum of the reactivities supplied by each batch. By combining multiple batches, the fresh fuel keeps older fuel, which would have already become subcritical, burning for a longer period of time, greatly increasing fuel efficiency.

(I intend to add content about burnable poisons.)

TOOLS AND IMPLEMENTATION

To create the graphical user interface I used PyQt4, which is a tool that brings together Python with Digia’s Qt C++ application framework. PyQt4 utilizes python’s object oriented programming to “employ a signal/slot mechanism for communicating between objects” [2]. It is an intuitive, yet powerful tool that is easy to implement. It contains all the basic objects needed to create a GUI, like textboxes, drop boxes, buttons, and more. It can also be coupled with matplotlib to create updatable graphs.

All of the nuclear data for this application was obtained from Poropy, which was provided by Dr. Roberts. Poropy is a software package that contains extensive data for nuclear reactor analysis and has a specific focus on in-core fuel management optimization. It is accessible through python by calling specific functions (with various parameters) from the software. For this project, I utilized the nuclear data package to obtain values for K_{inf} which depend on the fuel enrichment and type of burnable poison, as selected by the user.

For this project, I created a two column interface where the user can select the desired parameters on the left and select the type of plot they want to see on the right, as seen in Figure 1. The application begins with a sample K_{inf} vs. fuel burn up plot on the right. The user is then able to select the number of batches (1-6), the fuel enrichment (3%, 4%, and 5%), and the type of burnable poison (none, gadolinium, or IFBA).

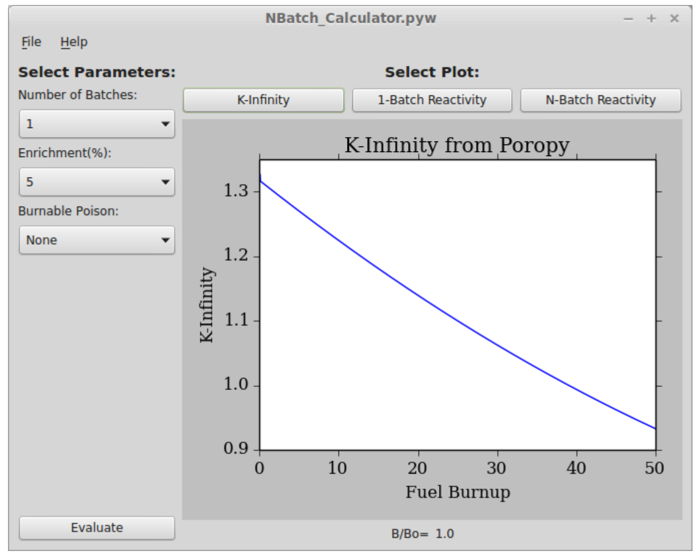

Figure 1: K-inf vs. fuel burn up, data acquired from Poropy

These parameters are entered into the K_{inf} function from Poropy to obtain K_{inf} as a function of burnup. Then the user can click the “evaluate” button to update the graph on the right to display N-batch reactivity vs. fuel burn up. Alternatively, either one of the three buttons above the plot can be clicked to update the plot depending on the parameters entered on the left. An example of the reactivity vs. fuel burn up using IFBA as a poison can be seen in Figure 2, and Gadolinium in Figure 3. I thought it would be important to add the ability to see K_{inf} vs. fuel burnup as well as one-batch reactivity vs. fuel burn up so the user can utilize it as a baseline.

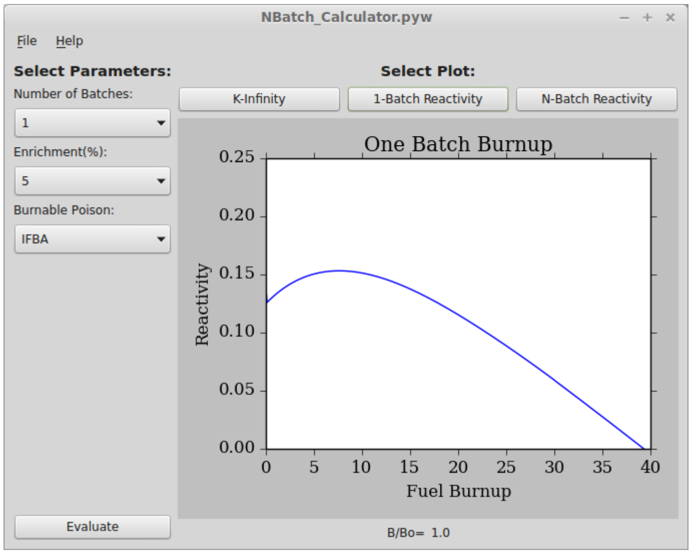

Figure 2: Calculated reactivity vs. fuel burn up for one batch using IFBA as a burnable poison

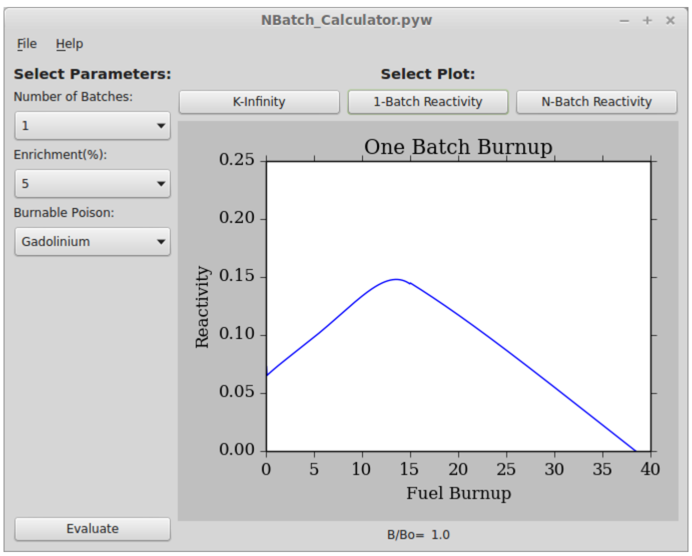

Figure 3: Calculated reactivity vs. fuel burn up for one batch using Gadolinium as a burnable poison

RESULTS

The main output of this application is the plot displayed on the right, which can be set to show K_{inf} or reactivity vs. fuel burn up for one or multiple batches. Below is a textbox that displays the ratio of the n-batch to the one-batch burn up so the user can see the increase in fuel efficiency by increasing the number of batches.

Right now I do not have the n-batch reactivity working because I cannot get the math to work. This is something I will be sure to accomplish before the end of the project so I can add some real results to this section.

REFERENCES

[1] Cochran and Tsoulfanidis, "The Nuclear Fuel Cycle." ←Get real citation from Dr. Robert's book

[2] Riverbank Computing Limited, "What is PyQt?"

-Fix citations and formatting errors (add equations, etc.)
-Finish Results section
-Add a discussion section
-future work? Improvements?