Putting it all Together, Part 4.

Parts 1 and 2 instructed you how to use multiple features of Serpent to build a simple LWR core model, and in Part 3 you constructed a more realistic core with individual fuel assemblies. However in Part 3, only 3 burnup regions in the fuel were specified. In reactor design it is important to obtain measurements at much higher resolution.

GOAL: To increase the realism of the previous model of light water reactor (LWR). To understand how to use reflective boundary conditions to reduce the runtime and improve the fidelity of reactor simulations.

BACKGROUND: Real reactors are composed of hundreds of fuel assemblies. When these cores are simulated, analysts must make decisions in order for the simulations to be realistic, while remaining computationally tractable under hardware and software constraints. Computing hardware gets more powerful all the time, but no matter how good it gets, the question will always be "how realistic can I make this with the given hardware?".

One aspect of this is in the choice of granularity in reactor burnup simulations. We have many options, we could approximate the core as consisting of 3 individually homogeneous regions, or we could model burnup in individual assemblies, or pins. We could further segment pins into a number of homogeneous regions. With each increase in granularity, the computational burden increases correspondingly. Analysts must strive to achieve enough granularity to answer the question they are trying to study with a good degree of confidence.

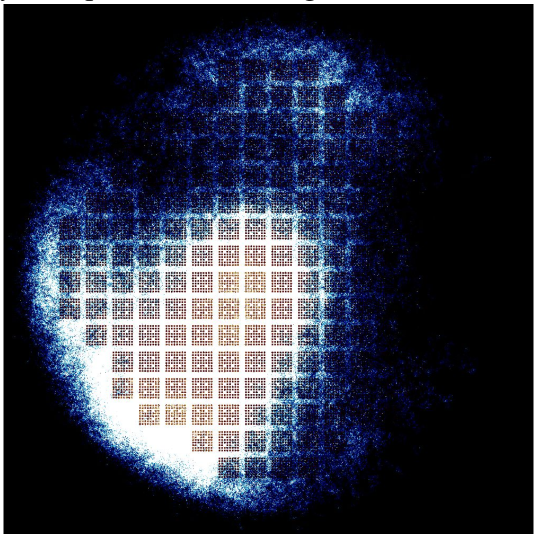
The question of how much granularity is enough is extremely problem dependent. However, Monte Carlo codes like Serpent provide tools to bring some insight to bear on this question, as well as to minimize the computational resources needed. In this lab session, we will use a common approach to increase the realism of our simulation, and reduce its overall RAM and CPU time requirements. We will simulate a core with 180 individual burnup regions. We will use a reflective boundary condition and make use of the 1/4 symmetry of the core to reduce the size of the problem. We will then identify additional symmetries to minimize the size of the problem.

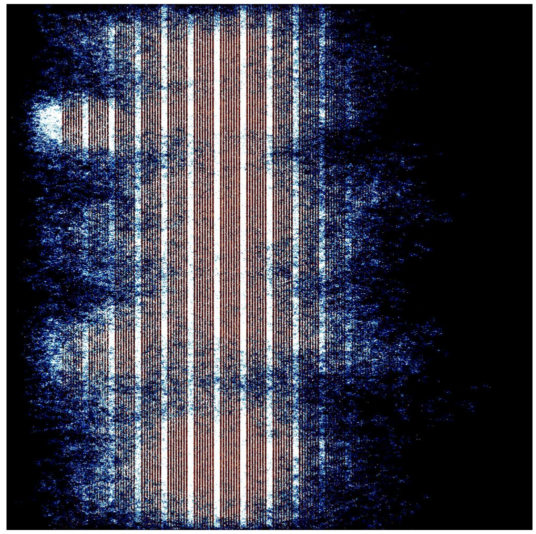
EXERCISES:

1) The template input deck ”180unique” packaged with this lab session contains definitions for 180 individual fuel materials, and their corresponding pin lattices already inserted into 180 assemblies. The fuel materials are from the 3 material definitions in the previous simulation. The first 60 assemblies go in the inner core, the next 60 assemblies go in the part of the core surrounding the inner core, and the last 60 assemblies go in the outer core region. The full core has been assembled in the text file lattice.txt. If you would like to see how these geometries were created check out the files in the folder “matlab code”.

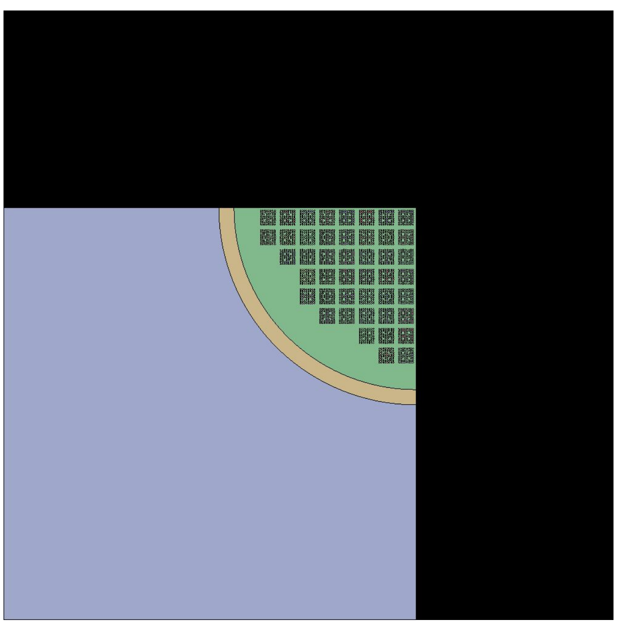
2) Run the input deck provided for a simulation with a single burnup step of 30 days at 3GW thermal power. “burn 1” was already added to your fuel materials, just add the power and burn

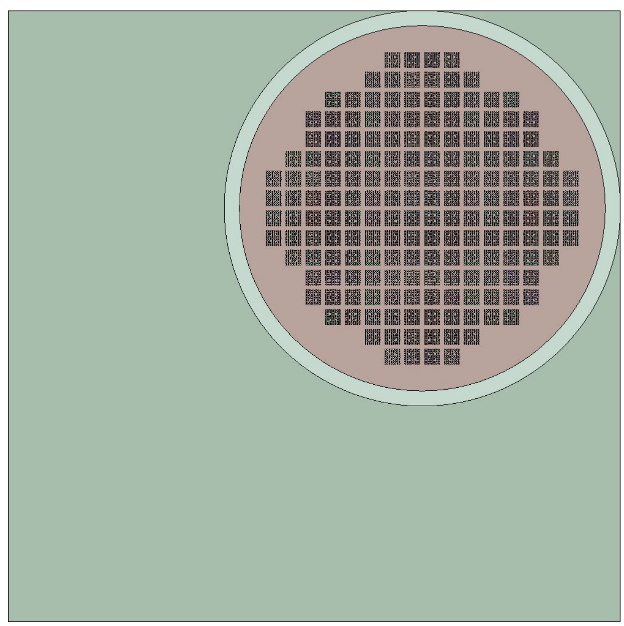
steps. Continue on to step 3) while waiting for outputs. Look at the flux distribution. Is it what you expected? How long did it take for the transport cycle to be completed?

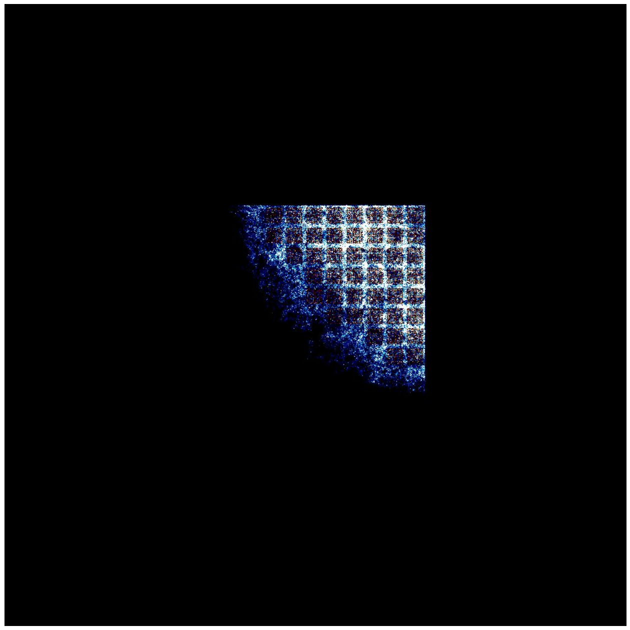




3) Modify the template input deck “45unique” of a 1/4 core. Place this core inside a large box. The box should be big enough to contain the 1/4 core. Don’t forget to recenter the cylinders and large box to fit around the lattice. Alternatively shift the lattice center. Fill the empty parts of the box (outside the containment vessel) with Boron-10 at 100 times natural density, you will need to define this in the file “B.txt”. NB. do not set the ^{10}B to be burnable. Run the deck just long enough to create a geometry plot and check them. Add reflective boundary conditions in the x and y directions. Re-run the deck and check your geometry plots again. Let the run finish this time.







Questions:

a) Why do we include ^{10}B?

b) What power do you now burn the core at?

c) Does the flux profile look any different to that in part 2)?

4) Your 1/4 core should have 45 assemblies, which means 45 fuel materials. When Serpent performs a burn simulation, it prints out the burnup value of each material, in MWd/kgIHM. Using the burnup values printed, can you find which assemblies you can set to have the same fuel material? I.e. which assemblies have the same burnup to ~5%?

5) Set all assemblies that have the same burnup to ~5% to contain the same material. Burn the core again, and comment on the results relative to previous. If you used more than 1 burnup step, could you use this approach to reduce your problem?