INL/EXT-13-30903

Initial RattleSnake Calculations of the Hot Zero Power BEAVRS

Matthew Ellis
Javier Ortensi
Yaqi Wang
Kord Smith
Richard C. Martineau

January 2014

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-13-30903

Initial RattleSnake Calculations of the Hot Zero Power BEAVRS

Matthew Ellis[1]
Javier Ortensi
Yaqi Wang
Kord Smith[1]
Richard C. Martineau

[1]Massachusetts Institute of Technology

January 2014

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Issued by the Idaho National Laboratory, operated for the United States Department of Energy by Battelle Energy Alliance.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Abstract

The validation of the Idaho National Laboratory's next generation of reactor physics analysis codes is an essential and ongoing task. The validation process requires a large undertaking and includes detailed, realistic models that can accurately predict the behavior of an operational nuclear reactor. Over the past few years the INL has developed the RattleSnake application and supporting tools on the MOOSE framework to perform these reactor physics calculations. RattleSnake solves the linearized Boltzmann transport equation with a variety of solution methods. Various traditional reactor physics benchmarks have already been performed, but a more realistic light water reactor comparison was needed to solidify the status of the code and determine its fidelity. The INL team decided to use the Benchmark for Evaluation and Validation of Reactor Simulations, which was made available in early 2013. This benchmark is a one-of-a-kind document assembled by the Massachusetts Institute of Technology, which includes two cycles of detailed, measured PWR operational data. The results from this initial study of the hot zero power conditions show the current INL analysis procedure with DRAGON4 cross section preparation and using the low order diffusion solver in RattleSnake for the whole core calculations yield very encouraging results for PWR analysis. The radial assembly power distributions, radial detector measurements and control rod worths were computed with good accuracy. The computation of the isothermal temperature coefficients of reactivity require further study.

Acknowledgment

We would like to acknowledge Hans Rüdiger Hammer for all his help with the mesh generation and Cubit expertise. Without him we would not have been able to develop a model with the reactor barrel and reactor pressure vessel in such short notice. The Authors are also grateful for Alain Hebert's support with the DRAGON4 calculations. Additional thanks to the MIT students that provided some of their results for our comparisons.

We are especially indebted to Benoit Forget, Kord Smith and their MIT team for generating the BEAVRS, which will prove invaluable to begin the validation INL's reactor physics analysis capabilities.

Contents

Figures

1 Introduction

The validation of a reactor physics code system is an essential part of the code development process. It allows the determination of the code prediction capabilities of the complex physical processes that take place in a nuclear reactor under various operational conditions. The Idaho National Laboratory (INL) has developed a next generation transport solver, RattleSnake[1], and supporting reactor physics analysis tools within the MOOSE[2] framework. These tools have been developed with the necessary modeling flexibility to analyze a variety of experiments and reactor types, including the Advanced Test Reactor (ATR). Various standard reactor physics benchmarks have already been conducted with the code. Other studies are concurrently taking place with the primary goal of preparing RattleSnake for ATR analysis. In addition, it is desirable to determine the capabilities of the code in predicting typical light water reactor (LWR) behavior.

In early 2013, a Massachusetts Institute of Technology (MIT) team made the Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS) available on the public domain. This benchmark is unlike other reactor physics benchmark because it provides an unprecedented two cycles of detailed, measured PWR operational data. The dataset includes actual detector measurements, which would be used in typical Hot Zero Power (HZP) physics testing and day-to-day core follow activities.

This report details the analysis of the BEAVRS using the continuous Galerkin finite element neutron diffusion solver in RattleSnake. The goals of this study are to compare the following HZP values given in the BEAVRS:

1. Axially integrated fission chamber measurements from 58 in-core detectors
2. Control rod worth for control rod bank D
3. Core temperature coefficient

The cross sections used in this benchmark were generated using DRAGON4, and the process for generating these cross sections is described in Section 2 of this report. The mesh generation process is outlined in Section 3, and results are presented in Section 4. Suggestions for future improvements are given at the end of this report.

2 Cross Section Generation

Cross sections for the BEAVRS were generated using the lattice physics code DRAGON4[3], which was developed at École Polytechnique Montréal. In DRAGON4, there are many solution techniques available for a LWR lattice. The Collision Probability Method solver, SYBILT, was chosen because of its speed with regard to the number of different materials that would be present in a single lattice.

In the cross section generation process it is customary to finely discretize the fuel assembly and solve some variant of the neutron transport equation over the assembly assuming reflective boundary conditions. However, because an infinite lattice is assumed in this calculation, it is important to make some approximation of what the neutron radial and axial leakage should be. The leakage rate depends on the following factors [4]:

1. Scattering anisotropy
2. Streaming effects caused by strong heterogeneities and/or low optical density regions in the lattice.

The neutron leakage model in lattice physics codes can be used to correct for both of these factors and, in addition, it is used to apply the fundamental mode approximation and compute the diffusion coefficients for the low order diffusion operator. Since in a PWR calculation the effect of scattering anisotropy is very important, because of the hydrogen-based moderator, the scattering anisotropy cannot be ignored in the leakage model calculation. To make sure that the scattering anisotropy is taken into account properly, a critical bucking search with a B1 leakage model (TYPE B B1) was used in DRAGON4 [4].

In order to preserve the reaction rates from the higher order transport solution in the lower order diffusion model one must use an equivalence procedure. The equivalence procedure available in DRAGON4 is the Superhomogeneisation (SPH) [5]. This process was performed on the same homogenized geometry that is shown in Figure 1(a) and Figure 1(b). It is worth noting that DRAGON4 provides many different SPH iterative options. For the cross sections presented in this paper, an SPH iteration between the SYBILT solution and a first-order finite element solution with 7 pin-mesh refinements was performed in order to obtain equivalence with a 'converged' diffusion solution. This initial SPH iterative procedure was chosen because of challenges generating a pin-cell sized mesh with the mesh tools used with previous benchmarks. This issue is described in Section 3. In future work when the core mesh generation process is finalized, an exact SPH iteration will be performed.

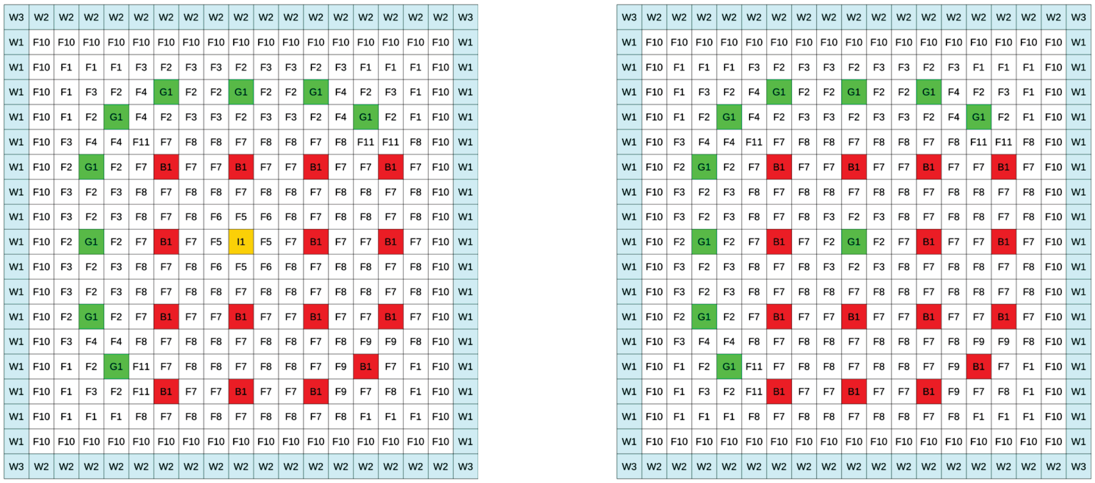
Three different iterations of cross sections were generated for use in the BEAVRS core. However, in the interest of brevity, only the last, most accurate method will be described in detail. The first two methods are briefly summarized below:

1. Only one assembly type for each enrichment was run in DRAGON4. Each assembly consisted of one collapsed cross section for fuel, guide tube, burnable poison, instrument tube,

and water gap between assemblies. The power distribution in the core was qualitatively poor and this method was abandoned.

2. Each assembly type in the BEAVRS benchmark was modeled in DRAGON4. Each assembly type consisted of only one collapsed cross section for fuel, guide tubes, and burnable poisons, and water gap between assemblies. The power distribution in the core was qualitatively acceptable, but pin power comparisons to published OpenMC values were significantly different, especially at the periphery of the core.

The third approach determined unique collapsed cross sections within an assembly based on relative locations to different material features in an assembly. For illustrative purposes the unique collapsed-cross section pin configuration for the 3.1 wt% enriched assembly with 15 burnable poisons is shown in Figure 1(a) and Figure 1(b).



W3	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W3
W1	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	W1
W1	F10	F1	F1	F1	F3	F2	F3	F3	F2	F3	F3	F2	F3	F1	F1	F1	F10	W1
W1	F10	F1	F3	F2	F4	G1	F2	F2	G1	F2	F2	G1	F4	F2	F3	F1	F10	W1
W1	F10	F1	F2	G1	F4	F2	F3	F3	F2	F3	F3	F2	F4	G1	F2	F1	F10	W1
W1	F10	F3	F4	F4	F11	F7	F8	F8	F7	F8	F8	F7	F8	F11	F11	F8	F10	W1
W1	F10	F2	G1	F2	F7	B1	F7	F7	B1	F7	F7	B1	F7	F7	B1	F7	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F8	F7	F8	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F6	F5	F6	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F2	G1	F2	F7	B1	F7	F5	I1	F5	F7	B1	F7	F7	B1	F7	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F6	F5	F6	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F8	F7	F8	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F2	G1	F2	F7	B1	F7	F7	B1	F7	F7	B1	F7	F7	B1	F7	F10	W1
W1	F10	F3	F4	F4	F8	F7	F8	F8	F7	F8	F8	F7	F8	F9	F9	F8	F10	W1
W1	F10	F1	F2	G1	F11	F7	F8	F8	F7	F8	F8	F7	F9	B1	F7	F1	F10	W1
W1	F10	F1	F3	F2	F11	B1	F7	F7	B1	F7	F7	B1	F9	F7	F8	F1	F10	W1
W1	F10	F1	F1	F1	F8	F7	F8	F8	F7	F8	F8	F7	F8	F1	F1	F1	F10	W1
W1	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	W1
W3	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W3

(a) Instrumented Assembly

W3	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W3
W1	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	W1
W1	F10	F1	F1	F1	F3	F2	F3	F3	F2	F3	F3	F2	F3	F1	F1	F1	F10	W1
W1	F10	F1	F3	F2	F4	G1	F2	F2	G1	F2	F2	G1	F4	F2	F3	F1	F10	W1
W1	F10	F1	F2	G1	F4	F2	F3	F3	F2	F3	F3	F2	F4	G1	F2	F1	F10	W1
W1	F10	F3	F4	F4	F11	F7	F8	F8	F7	F8	F8	F7	F8	F11	F11	F8	F10	W1
W1	F10	F2	G1	F2	F7	B1	F7	F7	B1	F7	F7	B1	F7	F7	B1	F7	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F8	F7	F8	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F3	F2	F3	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F2	G1	F2	F7	B1	F7	F2	G1	F2	F7	B1	F7	F7	B1	F7	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F3	F2	F3	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F3	F2	F3	F8	F7	F8	F8	F7	F8	F8	F7	F8	F8	F7	F8	F10	W1
W1	F10	F2	G1	F2	F7	B1	F7	F7	B1	F7	F7	B1	F7	F7	B1	F7	F10	W1
W1	F10	F3	F4	F4	F8	F7	F8	F8	F7	F8	F8	F7	F8	F9	F9	F8	F10	W1
W1	F10	F1	F2	G1	F11	F7	F8	F8	F7	F8	F8	F7	F9	B1	F7	F1	F10	W1
W1	F10	F1	F3	F2	F11	B1	F7	F7	B1	F7	F7	B1	F9	F7	F8	F1	F10	W1
W1	F10	F1	F1	F1	F8	F7	F8	F8	F7	F8	F8	F7	F8	F1	F1	F1	F10	W1
W1	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	F10	W1
W3	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W2	W3

(b) Non-instrumented Assembly

Figure 1: Collapsed Cross Section Pin Configuration for 3.1 wt% Enriched Assembly with 15 Burnable Poison Pins

Each assembly type has a similar set of pin definitions which are summarized in Table 1. Note that not all of the assemblies contain all of these pins; however, the cell labeling between assemblies in DRAGON4 input decks was consistent using the labels shown in Table 1.

Table 1: Homogenized Pin Descriptions

Pin Type	Desription
G1	Guide tube pin
I1	Instrument tube pin
B1	Burnable poison pin
F1	Fuel adjacent to fuel
F2	Fuel face-adjacent to (G1)
F3	Fuel diagonally-adjacent to (G1)
F4	Fuel diagonally-adjacent and face-adjacent to (G1)
F5	Fuel face-adjacent to (I1)
F6	Fuel diagonally-adjacent to (I1)
F7	Fuel face-adjacent to (B1)
F8	Fuel diagonally-adjacent to (B1)
F9	Fuel diagonnaly-adjacent and face-adjacent to (B1)
F10	Fuel adjacent to assembly water gap (W1, W2, W3)
F11	Fuel adjacent to (B1) and (I1)
W1, W3, W4	Water gap between assemblies

Because of the large number of homogenized pin types in Table 1, a Python script was created to help automate the generation of cross sections with DRAGON4. A script was written to automatically parse assembly diagrams like those shown in Figure 1(a) and Figure 1(b) and create the DRAGON4 input files for a given HZP temperature. An option was also created to automatically add spacer grids around each pin and around the entire lattice. This option has yet to be fully tested because a 3D model has not been run in RattleSnake due primarily to mesh generation challenges. Figure 2 shows the basic flow of the DRAGON4 input file automation. The following is a description of the function of each file in Figure 2:

- **assembly.py**: Assembly layouts are defined in this file. This file also contains functions for parsing assembly layouts and the class definition of Assembly().

- **pin.py**: Unique pin definitions are included in this file. This file also contains functions for retrieving text for different pin types needed in the DRAGON4 input file. The Pin class is defined in this file.

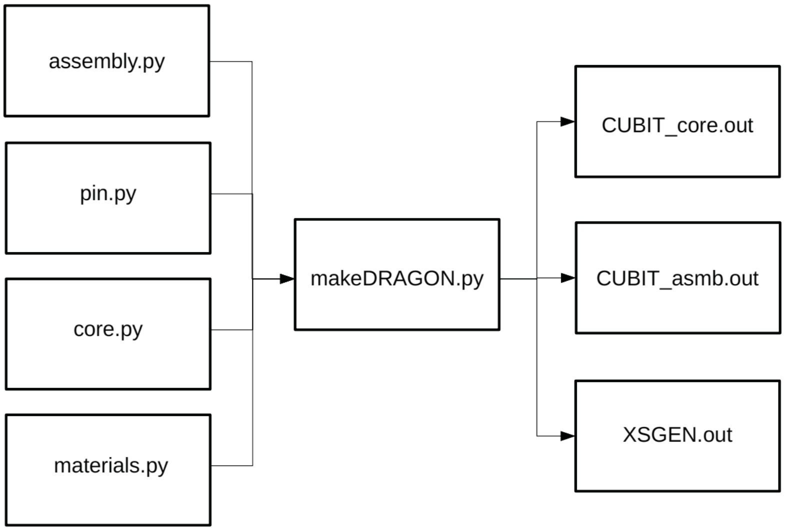
- **core.py**: This file contains the core layout for BEAVRS. It is used to generate the core layout needed to create the CUBIT mesh.

- **makeDRAGON.py**: This is the main driver script. This file is responsible for calling functions to create assemblies and pins and creates the output files. It also calls the DRAGON4 executable. There are file and directory dependencies that are hard coded into this file. When using this script pay attention to file and directory definition comments.

- **CUBIT_ core.out**: This file contains the core layout in Python list form that is generated by assigning correct assembly IDs used in the CUBIT assembly output.

- **CUBIT_ asmb.out**: This file contains each assembly in Python list form needed in the CUBIT mesh generator script. Note that the water gap around the assembly is removed. Also note that the pin IDs are not continuous because the water gap CUBIT material ID is set to the number of unique pins in the assembly plus one. This file also contains the "water vector" needed in the CUBIT mesh generator script.
- **XSGEN.out**: This output is used later by another Python script to automate the collection of cross section data from DRAGON4. It effectively contains a mapping of the input assembly layouts, the CUBIT mesh material/block IDs, and the DRAGON4 homogenized mixture numbers.

Figure 2: Python Script Flow for DRAGON4 input file generation.



The cross sections were retrieved from DRAGON4 using a Python script (xsgen.py) which executes XSGEN. The Python script also creates the RattleSnake material inputs which can be copied manually into a RattleSnake input file.

3 Mesh Generation

For the first two sets of cross sections discussed in the previous section a mesh was generated using INSTANT [6]. INSTANT has the ability to generate meshes for reactor analysis from an XML-based input file. The main disadvantage of INSTANT is that it currently does not have support for meshing curved structures like the core barrel and pressure vessel. Also, meshing the baffle and reflector causes uneven divisions within pin cells in order to maintain overall mesh structure. It was decided to use CUBIT to generate the final mesh after viewing initial results using INSTANT. The INSTANT mesh, however, was used with the second set of cross sections to calculate the rod worth of bank D. This was done only because time constraints did not allow for the recalculation of the control rod worth with the CUBIT mesh.

The BEAVRS mesh was generated using the Python interface in CUBIT. All of the radial features in the 2-D description of the BEAVRS core were included in the CUBIT model. This includes the baffle, barrel, reflector, shield pads, and pressure vessel. Portions of the full geometry are shown in quarter-core cuts in Figure 3 and Figure 4.

Figure 3: BEAVRS CUBIT Mesh with Different Fuel Assembly Types Colored By Object ID.

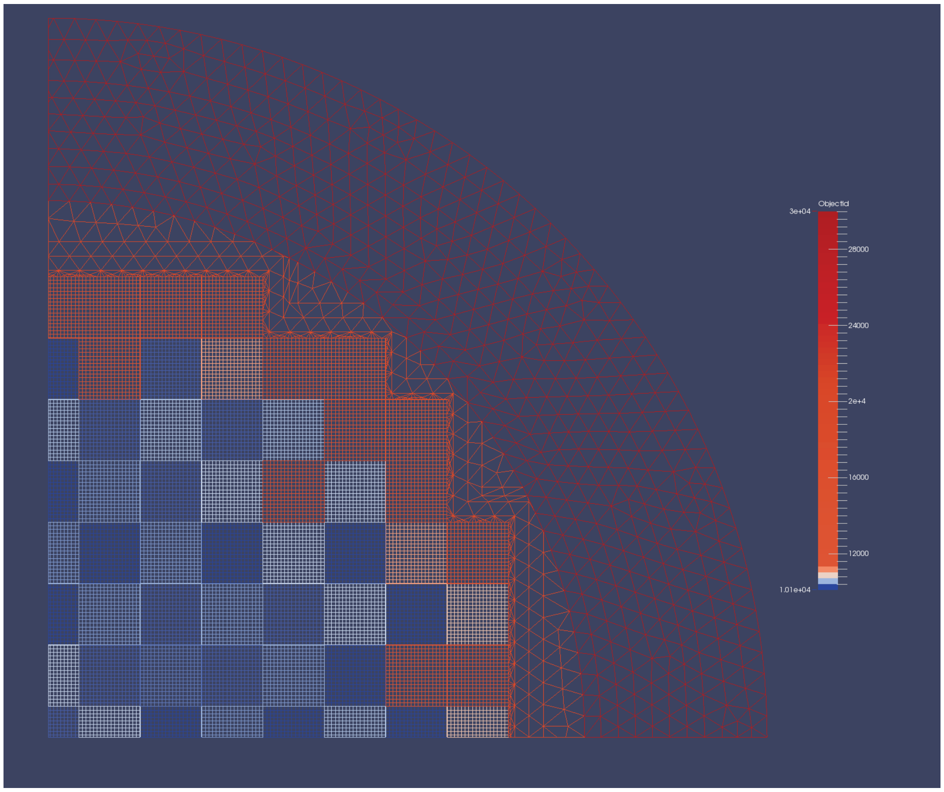
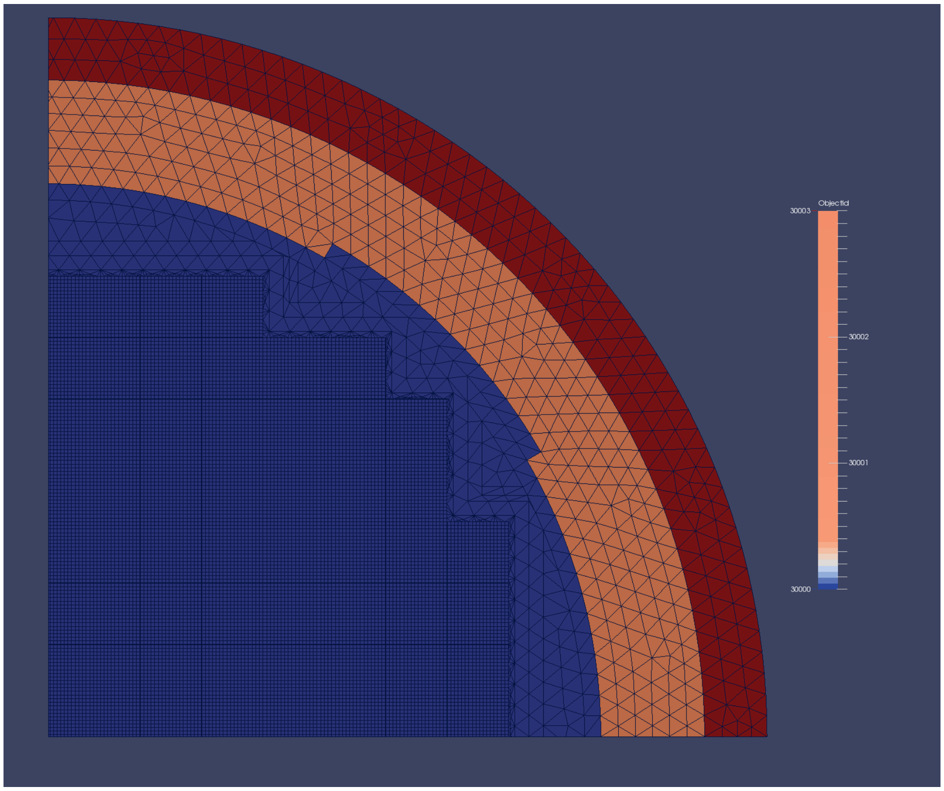
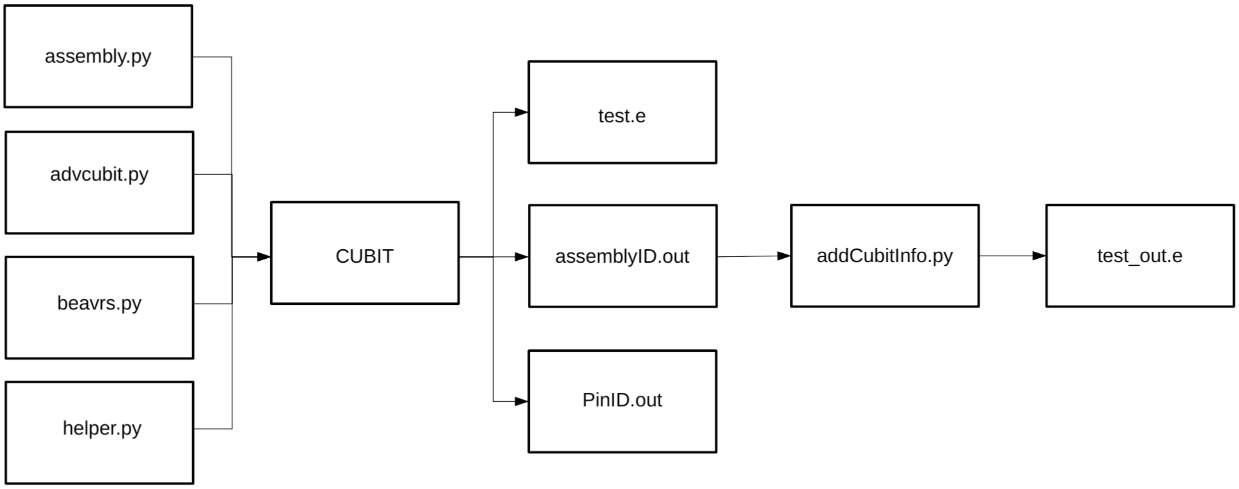


Figure 4: BEAVRS CUBIT Mesh with External Core Features Colored By Object ID.



Because the mesh generation process is quite complex and matching block IDs in CUBIT to material IDs from DRAGON4 is tedious for the number of unique materials present in the model, the mesh generating process was split into several independent Python scripts. The general flow of the mesh generation process is shown in Figure 5.

Figure 5: Python Script Flow for CUBIT Mesh Generation.



The function of the files shown in Figure 5 is summarized briefly as follows:

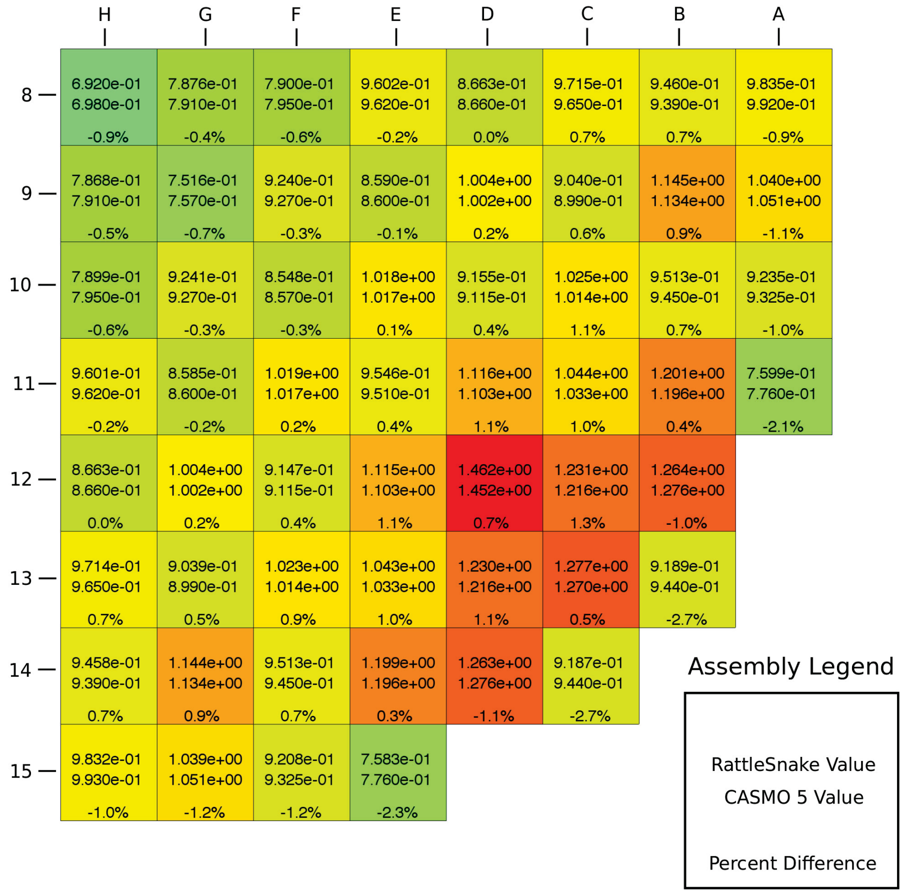
- **assembly.py**: This file is the main driver Python script that interacts with CUBIT. It controls the creation and movement of assemblies in the core. It also handles the imprinting functionality that overides the default imprinting functionality in CUBIT.
- **advcubit.py**: This file contains supporting function definitions used to interact with CUBIT.
- **beavrs.py**: This file contains the geometry description of the BEAVRS core. The contents of CUBIT_asmb.out and CUBIT_core.out from the makeDRAGON.py script are copied into this file.
- **helper.py**: This file contains function definitions needed by assembly.py.
- **test.e.py**: This is the mesh file created by CUBIT. It contains all of the material blocks defined by the assembly definitions in beavrs.py.
- **assemblyID.out**: This file contains assembly Cartesian coordinate numbering with assembly centroid locations relative to the center of the core.
- **pinID.out**: This file contains assembly pin Cartesian coordiante numbering with pin centroid locations relative to the center of the assembly.
- **addCubitInfo.py**: This file edits the original mesh exodus file (test.e) and adds the assembly ID and pin ID for every element based off of the information in assemblyID.out and pinID.out.
- **test_out.e**: This is the modified exodus file containing the assembly ID and pin ID information.

The main concern with using CUBIT to generate the mesh is that the CUBIT runtimes are very long. It takes approximately 2.5 hours to generate the 2-D mesh for BEAVRS. This long runtime will only be an issue if many refinement studies are needed with regards to the number of unique pin types for each assembly. If the number of pin types are changed per assembly, the mesh must be regenerated. For a 2-D process this is time consuming. Approaches to improve the meshing capabilities are undergoing.

4 Results

To get an estimate of the accuracy of the assembly-wise results, a CASMO-5[7] model of the 2-D BEAVRS core without axial buckling was run. The CASMO-5 model was run with 35 energy groups. A comparison of the assembly average power calculated by CASMO-5 and RattleSnake is shown in Figure 6. The RMS percent difference between CASMO-5 and RattleSnake is 0.961%. The largest error in the assembly powers is at the two-baffle adjacent assemblies where cross section production may need refinement. However, the fission detector comparisons that will be shown later suggest that the RattleSnake results at the periphery are quite good, at least for fission detector response prediction.

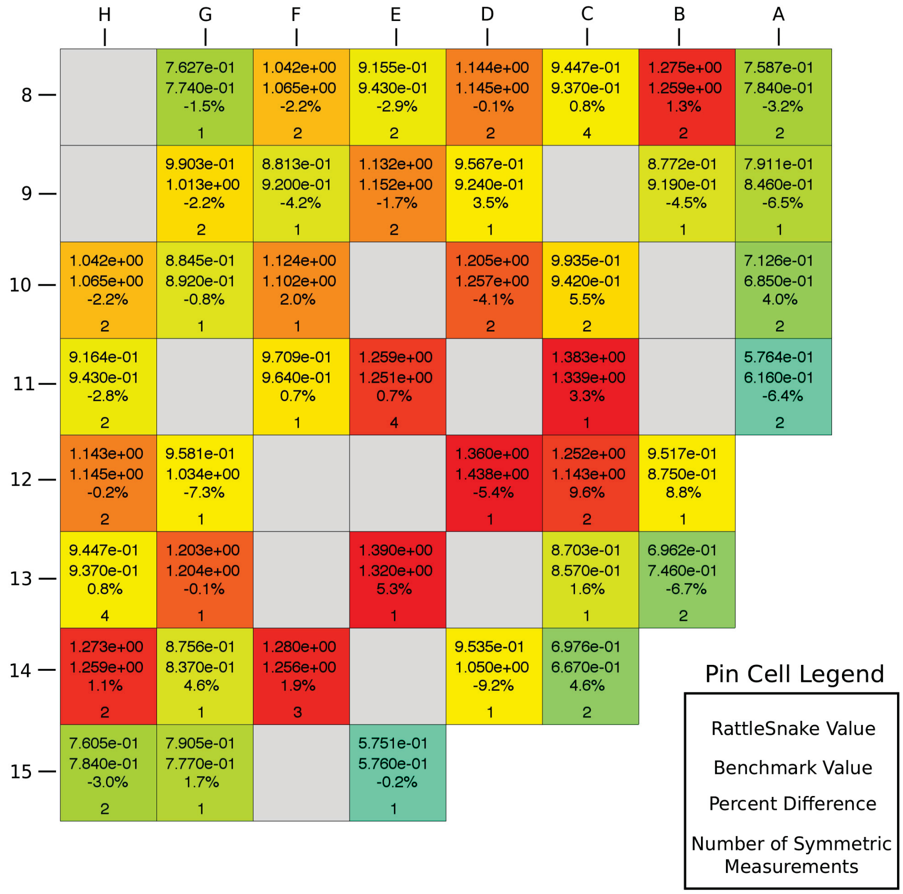
Figure 6: CASMO-5 and RattleSnake Normalized Assembly Power Comparison.



	H	G	F	E	D	C	B	A
8	6.920e-01 6.980e-01 -0.9%	7.876e-01 7.910e-01 -0.4%	7.900e-01 7.950e-01 -0.6%	9.602e-01 9.620e-01 -0.2%	8.663e-01 8.660e-01 0.0%	9.715e-01 9.650e-01 0.7%	9.460e-01 9.390e-01 0.7%	9.835e-01 9.920e-01 -0.9%
9	7.868e-01 7.910e-01 -0.5%	7.516e-01 7.570e-01 -0.7%	9.240e-01 9.270e-01 -0.3%	8.590e-01 8.600e-01 -0.1%	1.004e+00 1.002e+00 0.2%	9.040e-01 8.990e-01 0.6%	1.145e+00 1.134e+00 0.9%	1.040e+00 1.051e+00 -1.1%
10	7.899e-01 7.950e-01 -0.6%	9.241e-01 9.270e-01 -0.3%	8.548e-01 8.570e-01 -0.3%	1.018e+00 1.017e+00 0.1%	9.155e-01 9.115e-01 0.4%	1.025e+00 1.014e+00 1.1%	9.513e-01 9.450e-01 0.7%	9.235e-01 9.325e-01 -1.0%
11	9.601e-01 9.620e-01 -0.2%	8.585e-01 8.600e-01 -0.2%	1.019e+00 1.017e+00 0.2%	9.546e-01 9.510e-01 0.4%	1.116e+00 1.103e+00 1.1%	1.044e+00 1.033e+00 1.0%	1.201e+00 1.196e+00 0.4%	7.599e-01 7.760e-01 -2.1%
12	8.663e-01 8.660e-01 0.0%	1.004e+00 1.002e+00 0.2%	9.147e-01 9.115e-01 0.4%	1.115e+00 1.103e+00 1.1%	1.462e+00 1.452e+00 0.7%	1.231e+00 1.216e+00 1.3%	1.264e+00 1.276e+00 -1.0%	
13	9.714e-01 9.650e-01 0.7%	9.039e-01 8.990e-01 0.5%	1.023e+00 1.014e+00 0.9%	1.043e+00 1.033e+00 1.0%	1.230e+00 1.216e+00 1.1%	1.277e+00 1.270e+00 0.5%	9.189e-01 9.440e-01 -2.7%	
14	9.458e-01 9.390e-01 0.7%	1.144e+00 1.134e+00 0.9%	9.513e-01 9.450e-01 0.7%	1.199e+00 1.196e+00 0.3%	1.263e+00 1.276e+00 -1.1%	9.187e-01 9.440e-01 -2.7%		
15	9.832e-01 9.930e-01 -1.0%	1.039e+00 1.051e+00 -1.2%	9.208e-01 9.325e-01 -1.2%	7.583e-01 7.760e-01 -2.3%				

The fission chamber measurements were then compared to the 2-D axially integrated values provided in the BEAVRS. To help reduce the affect of the radial tilt in the BEAVRS measurements, the available detector locations were folded into a quarter core map. The results are shown in Figure 7. The RMS value of the fission chamber measurements is 5.20%. It is worth noting that in the positions where there are four symmetric measurements, the detector comparisons are very good ($< 1.0\%$). It is important to note that this is a comparison between a simplified 2-D model and axially integrated values from detector measurements.

Figure 7: RattleSnake Simulated Fission Detector Response Compared to the BEAVRS Measurements.



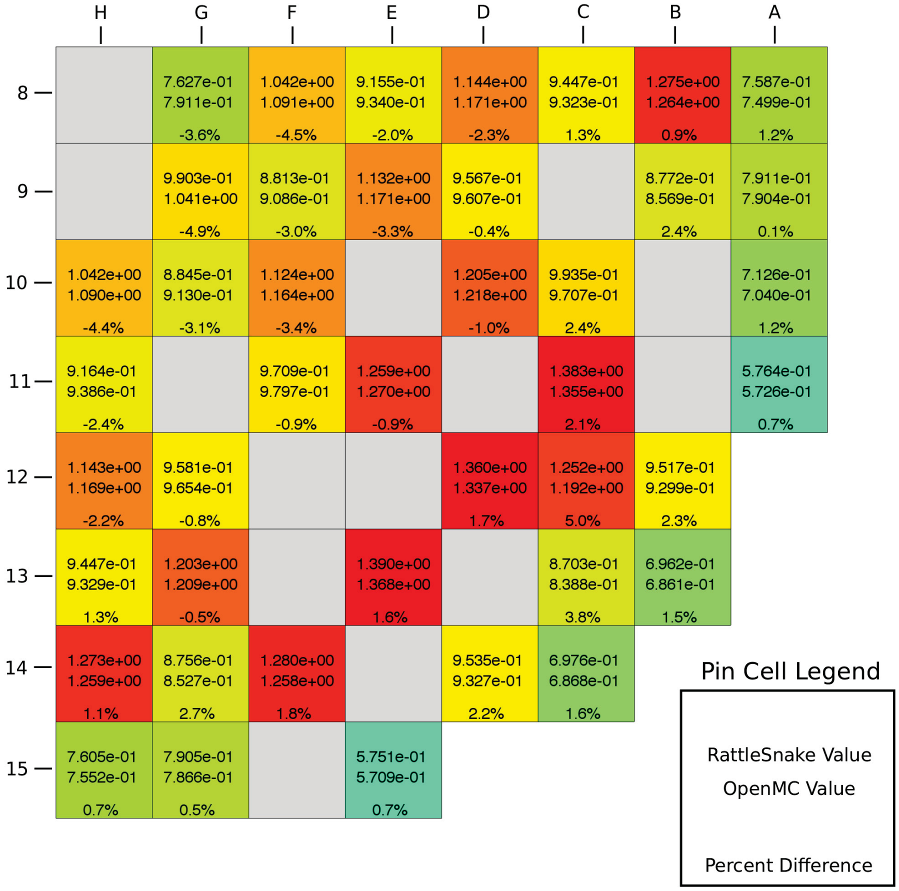
	H	G	F	E	D	C	B	A
8		7.627e-01 7.740e-01 -1.5% 1	1.042e+00 1.065e+00 -2.2% 2	9.155e-01 9.430e-01 -2.9% 2	1.144e+00 1.145e+00 -0.1% 2	9.447e-01 9.370e-01 0.8% 4	1.275e+00 1.259e+00 1.3% 2	7.587e-01 7.840e-01 -3.2% 2
9		9.903e-01 1.013e+00 -2.2% 2	8.813e-01 9.200e-01 -4.2% 1	1.132e+00 1.152e+00 -1.7% 2	9.567e-01 9.240e-01 3.5% 1		8.772e-01 9.190e-01 -4.5% 1	7.911e-01 8.460e-01 -6.5% 1
10	1.042e+00 1.065e+00 -2.2% 2	8.845e-01 8.920e-01 -0.8% 1	1.124e+00 1.102e+00 2.0% 1		1.205e+00 1.257e+00 -4.1% 2	9.935e-01 9.420e-01 5.5% 2		7.126e-01 6.850e-01 4.0% 2
11	9.164e-01 9.430e-01 -2.8% 2		9.709e-01 9.640e-01 0.7% 1	1.259e+00 1.251e+00 0.7% 4		1.383e+00 1.339e+00 3.3% 1		5.764e-01 6.160e-01 -6.4% 2
12	1.143e+00 1.145e+00 -0.2% 2	9.581e-01 1.034e+00 -7.3% 1			1.360e+00 1.438e+00 -5.4% 1	1.252e+00 1.143e+00 9.6% 2	9.517e-01 8.750e-01 8.8% 1	
13	9.447e-01 9.370e-01 0.8% 4	1.203e+00 1.204e+00 -0.1% 1		1.390e+00 1.320e+00 5.3% 1		8.703e-01 8.570e-01 1.6% 1	6.962e-01 7.460e-01 -6.7% 2	
14	1.273e+00 1.259e+00 1.1% 2	8.756e-01 8.370e-01 4.6% 1	1.280e+00 1.256e+00 1.9% 3		9.535e-01 1.050e+00 -9.2% 1	6.976e-01 6.670e-01 4.6% 2		
15	7.605e-01 7.840e-01 -3.0% 2	7.905e-01 7.770e-01 1.7% 1		5.751e-01 5.760e-01 -0.2% 1				

Pin Cell Legend

RattleSnake Value
Benchmark Value
Percent Difference
Number of Symmetric Measurements

Because of the large radial tilt, the detector predictions from a published OpenMC Monte Carlo model were compared to the RattleSnake values [8]. The comparison is included in Figure 8 and shows very good agreement between the two codes, even though the OpenMC model is in 3-D. A similar comparison was done with the CASMO-5 simulated detector responses and is shown in Figure 9. A summary of the comparisons is included in Table 2. The results show an improved solution of RattleSnake over CASMO-5, when compared agains the Monte Carlo solution.

Figure 8: RattleSnake Detector Responses Compared to the OpenMC Calculated Responses



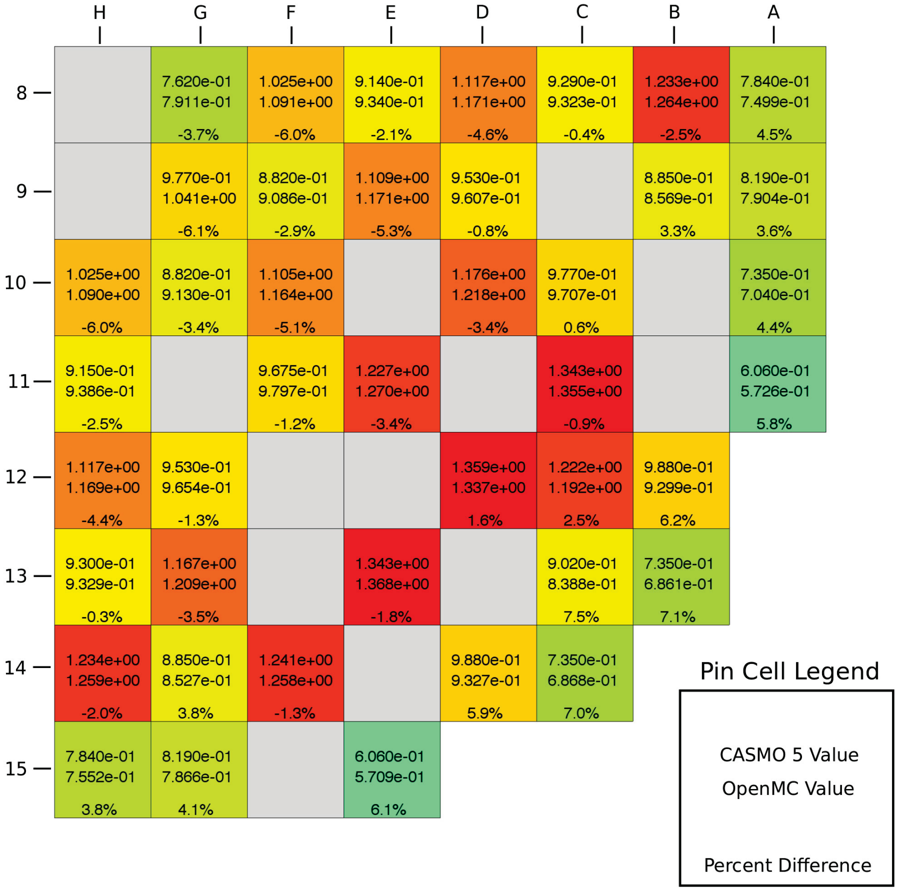
	H	G	F	E	D	C	B	A
8		7.627e-01 7.911e-01 -3.6%	1.042e+00 1.091e+00 -4.5%	9.155e-01 9.340e-01 -2.0%	1.144e+00 1.171e+00 -2.3%	9.447e-01 9.323e-01 1.3%	1.275e+00 1.264e+00 0.9%	7.587e-01 7.499e-01 1.2%
9		9.903e-01 1.041e+00 -4.9%	8.813e-01 9.086e-01 -3.0%	1.132e+00 1.171e+00 -3.3%	9.567e-01 9.607e-01 -0.4%		8.772e-01 8.569e-01 2.4%	7.911e-01 7.904e-01 0.1%
10	1.042e+00 1.090e+00 -4.4%	8.845e-01 9.130e-01 -3.1%	1.124e+00 1.164e+00 -3.4%		1.205e+00 1.218e+00 -1.0%	9.935e-01 9.707e-01 2.4%		7.126e-01 7.040e-01 1.2%
11	9.164e-01 9.386e-01 -2.4%		9.709e-01 9.797e-01 -0.9%	1.259e+00 1.270e+00 -0.9%		1.383e+00 1.355e+00 2.1%		5.764e-01 5.726e-01 0.7%
12	1.143e+00 1.169e+00 -2.2%	9.581e-01 9.654e-01 -0.8%			1.360e+00 1.337e+00 1.7%	1.252e+00 1.192e+00 5.0%	9.517e-01 9.299e-01 2.3%	
13	9.447e-01 9.329e-01 1.3%	1.203e+00 1.209e+00 -0.5%		1.390e+00 1.368e+00 1.6%		8.703e-01 8.388e-01 3.8%	6.962e-01 6.861e-01 1.5%	
14	1.273e+00 1.259e+00 1.1%	8.756e-01 8.527e-01 2.7%	1.280e+00 1.258e+00 1.8%		9.535e-01 9.327e-01 2.2%	6.976e-01 6.868e-01 1.6%		
15	7.605e-01 7.552e-01 0.7%	7.905e-01 7.866e-01 0.5%		5.751e-01 5.709e-01 0.7%				

Pin Cell Legend

RattleSnake Value
OpenMC Value

Percent Difference

Figure 9: CASMO-5 Detector Responses Compared to the OpenMC Calculated Responses



	H	G	F	E	D	C	B	A
8		7.620e-01 7.911e-01 -3.7%	1.025e+00 1.091e+00 -6.0%	9.140e-01 9.340e-01 -2.1%	1.117e+00 1.171e+00 -4.6%	9.290e-01 9.323e-01 -0.4%	1.233e+00 1.264e+00 -2.5%	7.840e-01 7.499e-01 4.5%
9		9.770e-01 1.041e+00 -6.1%	8.820e-01 9.086e-01 -2.9%	1.109e+00 1.171e+00 -5.3%	9.530e-01 9.607e-01 -0.8%		8.850e-01 8.569e-01 3.3%	8.190e-01 7.904e-01 3.6%
10	1.025e+00 1.090e+00 -6.0%	8.820e-01 9.130e-01 -3.4%	1.105e+00 1.164e+00 -5.1%		1.176e+00 1.218e+00 -3.4%	9.770e-01 9.707e-01 0.6%		7.350e-01 7.040e-01 4.4%
11	9.150e-01 9.386e-01 -2.5%		9.675e-01 9.797e-01 -1.2%	1.227e+00 1.270e+00 -3.4%		1.343e+00 1.355e+00 -0.9%		6.060e-01 5.726e-01 5.8%
12	1.117e+00 1.169e+00 -4.4%	9.530e-01 9.654e-01 -1.3%			1.359e+00 1.337e+00 1.6%	1.222e+00 1.192e+00 2.5%	9.880e-01 9.299e-01 6.2%	
13	9.300e-01 9.329e-01 -0.3%	1.167e+00 1.209e+00 -3.5%		1.343e+00 1.368e+00 -1.8%		9.020e-01 8.388e-01 7.5%	7.350e-01 6.861e-01 7.1%	
14	1.234e+00 1.259e+00 -2.0%	8.850e-01 8.527e-01 3.8%	1.241e+00 1.258e+00 -1.3%		9.880e-01 9.327e-01 5.9%	7.350e-01 6.868e-01 7.0%		
15	7.840e-01 7.552e-01 3.8%	8.190e-01 7.866e-01 4.1%		6.060e-01 5.709e-01 6.1%				

Pin Cell Legend

CASMO 5 Value
OpenMC Value

Percent Difference

Table 2: Summary of Quarter Core Assembly and Detector Calculation Comparisons

Calculation One	Calculation Two	RMS % Difference	Max. % Difference
RattleSnake Assembly Powers	CASMO-5 Assembly Powers	0.961%	2.68%
RattleSnake Detectors	BEAVRS Detectors	4.18%	7.53%
RattleSnake Detectors	OpenMC Detectors	2.37%	5.03%
CASMO5 Detectors	OpenMC Detectors	4.16%	7.52%

A summary of the eigenvalues produced by RattleSnake, CASMO-5, OpenMC, and SERPENT[9] are summarized in Table 3. Note that the SERPENT run was used to verify the eigenvalue results from RattleSnake because the published OpenMC model was three-dimensional and the RattleSnake model was only two-dimensional.

Table 3: Summary of Eigenvalue Results for the BEAVRS

Analysis Method	Eigenvalue Results
CASMO-5	1.00271
RattleSnake	1.00690
OpenMC (3-D)	0.99920 ± 0.00004
SERPENT	1.00530 ± 0.00001

As previously mentioned the control rod worth was measured using the second set of cross sections and the mesh generated by INSTANT. The calculated bank D rod worth was 761 pcm compared to a reference value of 788 pcm.

Finally, an attempt was made to calculate the core temperature coefficient. The cross sections were generated at 5 K above and below the reference HZP temperature of 566.483 K. A reference calculation was also run with SERPENT. The eigenvalue results are shown in Table 4.

Table 4: Summary of Eigenvalue Results for the Temperature Coefficient Calculation

Temperature (K)	RattleSnake Eigenvalue	SERPENT Eigenvalue
561.483	1.00674	1.00475
566.483	1.00690	1.00472
571.483	1.00680	1.00449

Table 5 summarizes the temperature coefficient calculated using the three different combinations of the values shown in Table 4. The results obtained with RattleSnake show a discrepancy with both the BEVRS and the SERPENT values, pointing to potential problems in the cross section preparation.

Table 5: Summary of Eigenvalue Results for the Temperature Coefficient Calculation

Temp. One (K)	Temp. Two (K)	RattleSnake Temp. Coef. (pcm/F)	SERPENT Temp. Coef.(pcm/F)
566.483	561.483	1.80	-0.330
566.483	571.483	-1.11	-2.53
561.483	571.483	-0.621	-1.43

The reference temperature coefficient given in the BEAVRS is -1.75 pcm/F.
At the time of this publication the parallel mesh capability in MOOSE was not functioning with the mesh generated in CUBIT. As a result, only two or three processors per node could be used while running this benchmark. With 2 processors per node and 15 nodes, the alive time of MOOSE was reported to be 540.45 seconds. The active time was 614.788 seconds. As reported in the MOOSE summary output, the total number of degrees of freedom for the results shown above is 2.3 million.

5 Conclusion

The results presented in the previous section are very encouraging with regards to the applicability of RattleSnake to full core reactor analysis. The RattleSnake model with cross sections from DRAGON4 produced assembly power values similar to those from CASMO-5. Additionally, fission chamber detector predictions by RattleSnake compared to those given in the BEAVRS were reasonable given the amount of radial tilt in the benchmark measurements. The fission chamber detector predictions by RattleSnake compared to published OpenMC results were closer than the comparison to the benchmark values. Additionally, the control rod worth was accurately predicted within 27 pcm.

The major deficiency in this benchmark work is the prediction of the core temperature coefficient. With the cross sections and CUBIT model shown in this report, the temperature coefficient could not be correctly predicted. Unfortunately, time constraints did not allow for a detailed investigation into the temperature coefficient error. However, it must be said that predicting a small change in core reactivity ($<$ 10 pcm) is very difficult, and a significant amount of time will need to be devoted to resolving the error in the temperature coefficient.

Future work should first focus on resolving the temperature coefficient calculation error. Additionally, a full three-dimensional model should be produced to see if the correct eigenvalue, as reported in the OpenMC calculation, can be reproduced using RattleSnake. Finally, hot full power (HFP) should be simulated using DRAGON4 and RattleSnake. The results in this document show that the continuous Galerkin Finite Element neutron diffusion solver in RattleSnake produces very good results for the BEAVRS. The cross section generation technique outlined in this document was sufficient to produce fission detector responses that agree within 5% of a reference OpenMC calculation in 3-D. Additionally, the eigenvalue for the 2-D BEAVRS model in RattleSnake produced an eigenvalue that was only 159 pcm different from a reference Monte Carlo calculation performed in SERPENT. Future work should be focused on developing a full three dimensional model in RattleSnake of the BEAVRS core.

References

[1] Yaqi Wang. Nonlinear diffusion acceleration for the multigroup transport equation discretized with sn and continuous fem with rattlesnake. In *Proceedings to the International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2013)*, Sun Valley, Idaho, USA, May 5-9 2013.

[2] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandie. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.

[3] G. Marleau, A. Hébert, and R. Roy. A user guide for dragon version4. IGE-294, École Polytechnique de Montréal, July 2013.

[4] Alain Hebert. *Applied Reactor Phyics*. Presses Internationales Polytechnique, 2009.

[5] A. Hebert and G. Mathonniere. Development of a third-generation superhomogenization method for the homogenization of a pressurized water reactor assembly. *Nuc. Sci. Eng.*, 2:115, 1993.

[6] Yaqi Wang, Cristian Rabiti, and Giuseppe Palmiotti. Krylov solvers preconditioned with the low-order red-black algorithm for the PN hybrid FEM for the INSTANT code. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*, Rio de Janeiro, RJ, Brazil, May 8–12 2011. Latin American Section (LAS) / American Nuclear Society (ANS).

[7] Joel Rhodes, Kord Smith, and Deokjung Lee. CASMO-5 development and applications. In *PHYSOR-2006, ANS Topical Meeting on Reactor Physics*. American Nuclear Society, September 10-14 2006.

[8] Nicholas Horelik, Bryan Herman, Beoit Forget, and Kord Smith. Benchmark for evaluation and validation of reactor simulations (BEAVRS). *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*, 2013.

[9] J. Leppänen. Development of a new monte carlo reactor physics code. D.Sc. Thesis VTT Publications 640, Helsinki University of Technology, 2007.

Idaho National Laboratory