



# Modeling of the Advanced Test Reactor Using OpenMC, Cubit, and Griffin

August 2023

Khang H. N. Nguyen<sup>1</sup> and Mark D. DeHart<sup>2</sup>

<sup>1</sup>*Department of Nuclear Engineering, North Carolina State University*

<sup>2</sup>*Reactor Physics Methods Analysis Dept.*



*INL is a U.S. Department of Energy National Laboratory  
operated by Battelle Energy Alliance, LLC*

#### **DISCLAIMER**

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References 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 U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

# **Modeling of the Advanced Test Reactor Using OpenMC, Cubit, and Griffin**

**Khang H. N. Nguyen<sup>1</sup> and Mark D. DeHart<sup>2</sup>**

**<sup>1</sup>Department of Nuclear Engineering, North Carolina State University**

**<sup>2</sup>Reactor Physics Methods Analysis Dept.**

**August 2023**

**Idaho National Laboratory  
Reactor Physics Methods and Analysis Dept.  
Idaho Falls, Idaho 83415**

**<http://www.inl.gov>**

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

*Page intentionally left blank*

## SUMMARY

In the pursuit of the ability to perform multiphysics simulations of the Advanced Test Reactor, it is crucial to have a fast and highly accurate deterministic model. To achieve this, a contemporary two-step method is employed. The first step involves generating homogenized cross sections using OpenMC, a cutting-edge Monte Carlo neutron transport code. OpenMC offers excellent modular capabilities, allowing for easy component integration and flexibility in incorporating new designs into the model. The second step involves deterministic transport calculations, which are performed using Griffin, a reactor multiphysics application based on the Multiphysics Object-Oriented Simulation Environment. To ensure the accurate spatial resolution and assignment of material cross sections, a Cubit-generated mesh for the Advanced Test Reactor is utilized as an intermediate step between the OpenMC and Griffin models; Griffin utilizes the mesh for its finite element solution, while OpenMC material IDs are written to the mesh file to be used in Griffin material assignments. Additionally, a Python-based script converts the cross sections generated by OpenMC into the ISOXML format required by Griffin. Preliminary comparisons indicate good agreement between the neutron multiplication factors obtained from the standalone OpenMC model and the Griffin model, with differences of less than 50 pcm in the two-dimensional geometry configuration. However, in three-dimensional calculations, an unacceptably large error is found in the Griffin solution. Future work is planned to resolve this discrepancy.

*Page intentionally left blank*

# CONTENTS

SUMMARY .....	i
1 INTRODUCTION .....	1
2 COMPUTER CODES .....	2
2.1 OpenMC.....	2
2.2 Cubit .....	2
2.3 Griffin.....	2
3 APPROACH .....	3
3.1 OpenMC Model .....	3
3.2 Converting Cross Section to ISOXML Format .....	4
3.3 Mesh Generation Using Cubit and Extrude Mesh Utility in Griffin .....	5
3.4 ATR Model in Griffin.....	7
4 RESULTS.....	8
5 CONCLUSIONS AND FUTURE WORK .....	12
REFERENCES .....	14
Appendix : Input Files Summary .....	15

*Page intentionally left blank*



# 1 INTRODUCTION

The Advanced Test Reactor (ATR), located at Idaho National Laboratory (INL), is a 250 MW<sup>th</sup> high-flux test reactor, with a variety of missions including the accelerated testing of nuclear fuel and other materials in a very high neutron flux environment, medical and industrial isotope production. Construction of the ATR began in November 1961 and was completed in late 1965. Fuel loading commenced in 1967, core testing was completed in 1969, and finally, full power operation was achieved in August 1969.

Simulation and modeling play a crucial role in various aspects of ATR operations, including experiment design, fuel cycle management, and core experiment safety analysis. Experiment design and analysis for the ATR typically involve sophisticated three dimensional (3D) Monte Carlo analysis, utilizing the well-known continuous-energy MCNP5 code [1]. In this report, we present a contemporary two-step method for modeling the ATR, aiming to develop an appropriate computational mesh, a new cross-section generation approach and a 3D deterministic solution for neutron fluxes and reaction rates. This set of tools will provide a powerful multiphysics analysis capability with substantial flexibility in updating new designs or experiments. This involves utilizing the Monte Carlo code OpenMC [2], which serves as both a standalone analysis and verification tool, that can be used in conjunction with Griffin [3] to solve the neutron transport problem. To provide a spatial mesh for finite element analysis and to accurately assign cross-section data to the Griffin model, a computational mesh generated by Cubit [4] is employed. The overall process of verifying and validating the ATR model is depicted in Figure 1.

This report is structured as follows: first is an introduction to the ATR and the model used in this study. Next is a concise overview of the computer codes used. Preliminary results on ATR validation model completion are in the last section. This novel two-step method demonstrates its potential to enhance the efficiency and flexibility of the ATR model analysis by providing for the accuracy of a deterministic transport solution coupled with the multiphysics capabilities of Griffin.

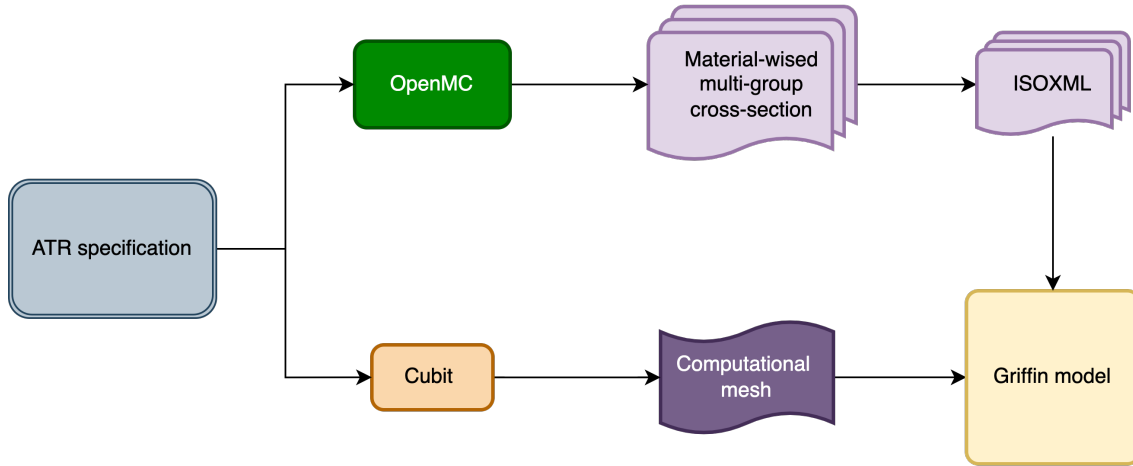


Figure 1: Workflow for ATR modeling with Griffin.

## 2 COMPUTER CODES

### 2.1 OpenMC

OpenMC is an open-source, community-developed Monte Carlo neutron and photon transport simulation code [2]. It offers a simulation and modeling capability for large-scale advanced nuclear systems that builds upon the capabilities of modern high-performance computational systems. The calculation can be performed on models built using either constructive solid geometry or computer-aided design representing a complex geometry. OpenMC supports both continuous-energy and multigroup transport. The continuous-energy particle interaction data is based on a native Hierarchical Data Format Version 5 format that can be generated from ACE-format files produced by NJOY [5]. Moreover, with the Python Application Programming Interface (API) language, users can have greater flexibility in generating input files, including handling materials, geometry, tallies, source definition, etc. During this work, we constructed an OpenMC model for ATR with detailed geometry from the ATR benchmark specification [6]. Consequently, a set of multigroup cross sections was generated for each material in the model, which are later used as input for the neutron transport code Griffin.

### 2.2 Cubit

Cubit [4] is a geometry and mesh generation software tool developed jointly by Coreform, LLC and Sandia National Laboratory. The main purpose of Cubit is to reduce the time required to generate meshes, especially large hexahedral meshes for complicated geometries. It generates 2D and 3D meshes for finite element analysis and computational fluid dynamics simulations. Cubit contains various meshing algorithms, such as quadrilateral and triangular paving, 2D and 3D mapping, hexahedral sweeping, and tetrahedral meshing. It also has algorithms to automate and control the meshing process, such as scheme selection, interval matching, sweep grouping, smoothing, and scaling. Furthermore, Cubit also developed a Python API to accelerate the mesh generation process. Within the scope of this work, this API is used with Cubit to create a finite element mesh and assign material cross sections generated from OpenMC model, and thus, to provide a link between cross-section spatial placement and the neutron transport calculation.

### 2.3 Griffin

In late 2019, INL and Argonne National Laboratory (ANL) agreed to jointly develop a new reactor physics code, named Griffin [3], which is based on integrating and combining the capabilities of two existing code suites, MAMMOTH/Rattlesnake (INL) and MC<sup>2</sup>-3/PROTEUS (ANL). Griffin is being developed based on the Multiphysics Object-Oriented Simulation Environment (MOOSE) framework and MOOSE quality assurance procedures. It leverages several multiphysics applications based on the MOOSE framework, including thermal-fluid applications like Pronghorn, RELAP-7, and Sockeye, as well as the fuel performance application BISON. Griffin has been utilized to model and analyze various reactor designs, such as pebble-bed reactors, prismatic reactors, molten-salt reactors, fast sodium-cooled reactors, nuclear thermal propulsion systems, and experimental reactor facilities. The multiphysics-coupling capabilities of Griffin allow it to integrate neutronic calculations with thermal-hydraulics simulations and fuel performance modeling. This enables Griffin to simulate the complex interactions between the neutron flux distribution, coolant flow conditions, and fuel behavior within a reactor system.

### 3 APPROACH

This work began through the development of a 3D OpenMC model of the core. This model was used to perform a check against the results provided in the benchmark specification.

#### 3.1 OpenMC Model

In this study, we modeled ATR using the geometry and materials specifications outlined in the ATR benchmark specification provided in Reference [6]. The objective is to ensure the model's accuracy and reliability, which is subsequently validated against reported results to ensure its fidelity.

The 3D benchmark ATR model is comprised of 43 distinct materials. We used OpenMC to generate a multigroup cross sections and establish a reference solution for comparison. OpenMC also computes and generates flux-weighted multigroup neutron cross sections for the various materials, which will be required for the deterministic neutron transport calculation.

Figure 2 provides a structured flowchart illustrating the process of generating the input for OpenMC for the ATR model. This diagram serves as a visual guide, aiding in the step-by-step creation of the input files required to run the simulation. The detailed description for each Python script is presented in Appendix 5.

The modular approach used in this process allows for better organizing and flexibility in handling various components of the model. Each component is separated into distinct modules, enabling users to conveniently input new designs, materials, or parameters to adapt the model for different scenarios. Therefore, it allows the easy exploration of different configurations and scenarios to gain valuable insights and optimize ATR modeling performance according to specific needs. An x-y cross-sectional cut of the geometry generated by the OpenMC model is illustrated

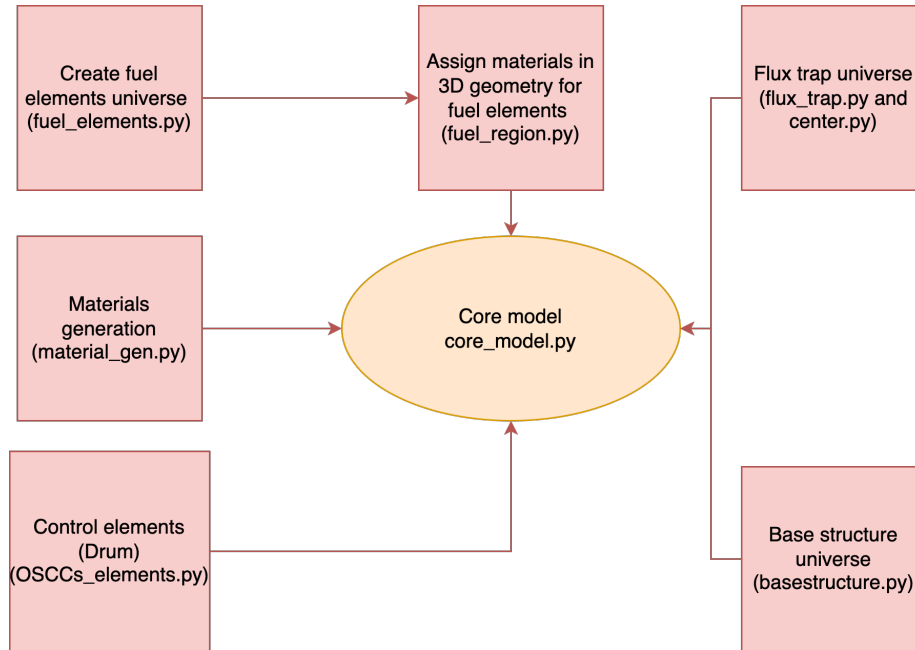


Figure 2: Input generation flow chart in OpenMC ATR model.

in Figure 3.

An 18-group cross-section specification is also defined in the core model, based on the CASMO 18-group library data provided in Reference [7]. The energy group structure assumed for this work is given in Table 1.

In this study, we generated two ATR models for OpenMC, the 3D model discussed earlier, and developed an effective 2D axial-slice model based on the core configuration at the midplane fuel with a segment height of 1.0 cm. Reflective boundary conditions were applied to the top and bottom of this segment to simulate an isolated slice of the core. The 2D model was also used for cross-section generation and was linked to a 2D Griffin model to allow for testing and mesh refinement studies, as discussed in Section 4.

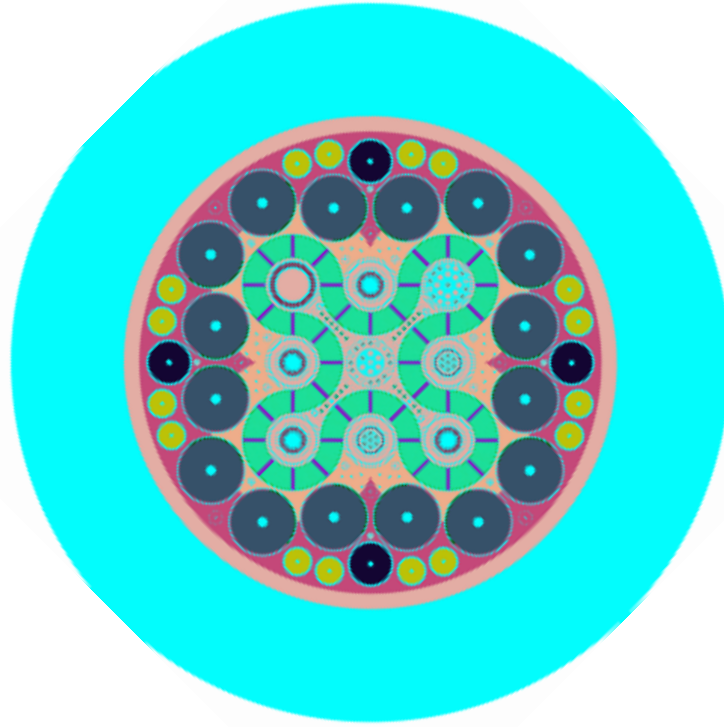


Figure 3: x-y cross-section view of ATR generated by OpenMC.

### 3.2 Converting Cross Section to ISOXML Format

We developed a Python-based converter script to translate the multigroup cross sections generated by OpenMC into the ISOXML format required by the Griffin reactor simulation code. The cross sections produced by OpenMC are not directly compatible with Griffin's solver algorithms, necessitating a conversion utility. This script converts the cross sections into either a tabulated or

Table 1: Multigroup structure.

Group Index	Energy Range (eV)	Group Index	Energy Range (eV)
1	0–5.8E-2	10	9.877–15.968
2	5.8E-2–0.14	11	15.968–1.48728E2
3	0.14–0.28	12	1.4728E2–5.53E3
4	0.28–0.625	13	5.53E3–9.118E3
5	0.625–0.972	14	9.118E3–1.11E5
6	0.972–1.15	15	1.11E5–5.0E5
7	1.15–1.855	16	5.0E5–8.21E5
8	1.855–4.0	17	8.21E5–2.2316E6
9	4.0–9.877	18	2.2316E6–2.0E7

untabulated macroscopic multigroup energy structure format. The conversion process is straightforward for all types of reaction cross sections except removal cross sections, which are calculated as in Equation 3.1. Currently, the converter is designed to produce cross sections for steady-state simulations in Griffin, though with modifications, it could be expanded to provide cross-section sets for transient or multiphysics simulations. The script is only compatible with scattering matrices that employ a Legendre polynomial expansion of order one or higher to approximate the angular distribution of scattered neutrons. The converter utility facilitates using OpenMC-generated cross sections as input data for a core calculation using Griffin.

$$\Sigma_r^g = \Sigma_{ra}^g + \Sigma_{ss}^g - \Sigma_{p0}^{g \rightarrow g} \quad (3.1)$$

where:

- $\Sigma_r^g$  is the removal cross section of group  $g$
- $\Sigma_{ra}^g$  is the reduced absorption cross section of group  $g$ , which is defined as the difference between the absorption and production of neutrons due to (n,xn) reactions
- $\Sigma_{ss}^g$  is the total scattering cross section of group  $g$
- $\Sigma_{p0}^{g \rightarrow g}$  is the diagonal elements of the  $P0$  scattering matrix.

### 3.3 Mesh Generation Using Cubit and Extrude Mesh Utility in Griffin

The geometry of ATR in the benchmark specification is effectively variable in the x-y plane but uniform in z; material can change axially but the geometry does not change. Hence, we created a 2D mesh of ATR that could be extruded to three dimensions. This 2D mesh was generated using Cubit similarly to creating the OpenMC geometry model. The 2D mesh consists of 58 mesh blocks with triangular mesh elements and one block with quadrilateral mesh elements, which is illustrated in Figure 4. Here, Cubit uses blocks to group related sets of elements into a single entity. Each element in an element block must have the same element type and material assignment [4].

Using the MOOSE mesh extrusion tool, the 2D Cubit mesh extrudes a 3D geometry using the axial discretization scheme provided in Table 2. The scheme contains 13 axial blocks to account for

axial changes in materials, as illustrated in Figure 5. This resulted in the 3D mesh expanding from the original 59 blocks in the 2D mesh to over 700 blocks. The axial discretization introduces additional mesh blocks to more accurately represent the axial variation in geometry, materials, and cross sections within the ATR model, as depicted in Figure 6. The mesh serves as an intermediate step between the OpenMC and Griffin models. The material-wise multigroup cross sections in the ISOXML format, described in the previous section, must be mapped onto the mesh blocks in Cubit. This allows Griffin to perform deterministic transport calculations using the Cubit-generated mesh and OpenMC-derived cross sections. The Python scripts used to generate the mesh also provide this mapping. The use of a mesh-based intermediate step facilitates the integration of the OpenMC and Griffin models. The mesh provides a mapping between the continuous spatial distribution of materials and cross sections in the OpenMC model and the discrete spatial representation required by the Griffin transport solver.

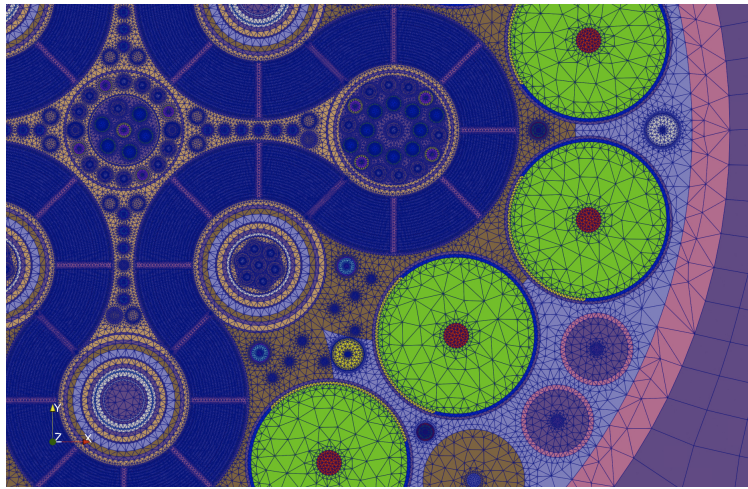


Figure 4: 2D mesh generated for ATR using Cubit.

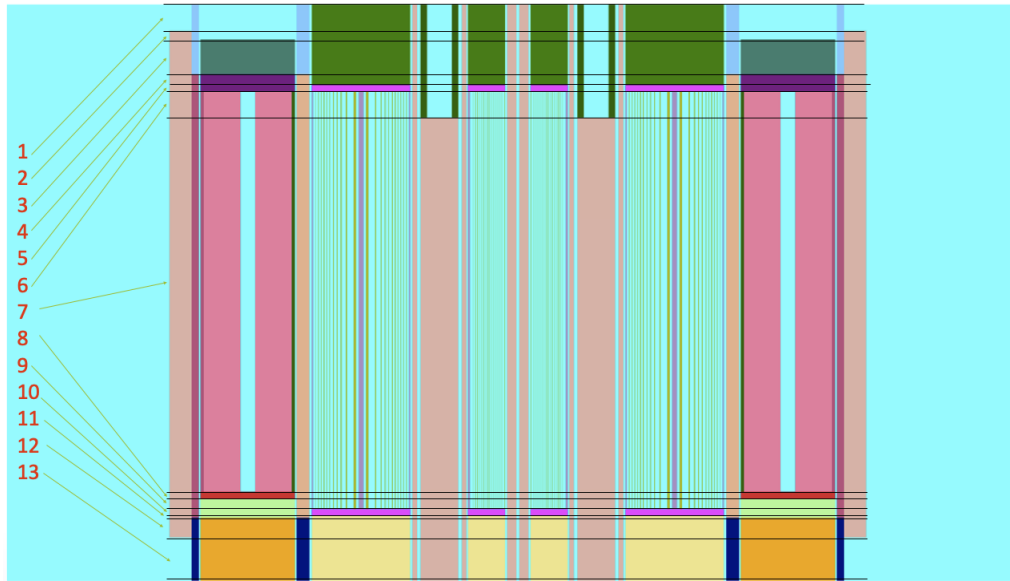


Figure 5: Axial node indexing scheme for ATR.



Table 2: Axial discretization for 3D ATR model.

Axial Node	Height (cm)	Plane Begin (cm)	Plane End (cm)	# Subnode
1	7.62	147.32	139.7	1
2	2.54	139.7	137.16	1
3	10.16	137.16	127	1
4	3.175	127	123.825	1
5	1.905	123.825	121.92	1
6	7.62	121.92	114.3	1
7	109.22	114.3	5.08	11
8	2.0637	5.08	3.0163	1
9	3.0163	3.0163	0	1
10	1.9505	0	-1.9505	1
11	0.5895	-1.9505	-2.54	1
12	5.715	-2.54	-8.255	1
13	12.74	-8.255	-20.995	1

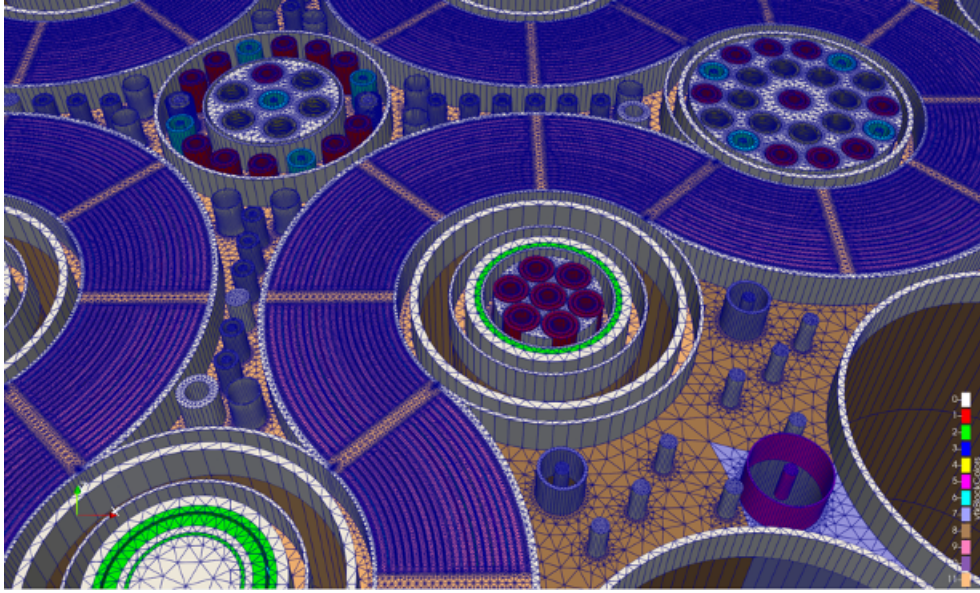


Figure 6: 3D mesh generated for ATR using the extruded mesh generator in Griffin.

### 3.4 ATR Model in Griffin

The Griffin input was prepared with material assignments corresponding to the material names provided by OpenMC and ported to the ISOXML cross-section library and assigned to the appropriate blocks in the mesh file. Paths were provided to the Cubit-generated mesh and ISOXML

cross-section file. The eigenvalue-type executioner was specified, with CFEM-diffusion option was selected as the transport system solution scheme; this applies the continuous finite element to discretize the Griffin diffusion solver, base on the weak form of the multigroup diffusion equation [8].

As noted earlier, two Griffin models were created; the first was a 2D model with 58 mesh blocks containing a subset of assigned materials (top and bottom materials not present in this model). The second model, based on the first, included the mesh extrusion input and assigned materials to the mesh for each axial and radial block. The input was expanded to read the full set of materials from the appropriate ISOXML library and assign materials to the correct blocks from the 3D mesh.

## 4 RESULTS

Once both the 2D and 3D OpenMC models had been developed, we conducted a verification process to ascertain the accuracy and reliability of the OpenMC model relative to an MCNP model of the same configuration. This involved comparing the computed neutron multiplication factor with the values obtained from the ATR benchmark [6]. The results exhibit a high degree of agreement, as shown in Table 3, which compares the results of the OpenMC calculation to those of MCNP for both the full 3D calculation and the calculation for the 2D (axially infinite) model. This effectively confirms the fidelity of the OpenMC model relative to MCNP. Based on the evaluated biases and uncertainties in the benchmark specification [6], the critical state would have a value of  $1.0018 \pm 0.0035$ . Both the OpenMC and MCNP models fall within this range.

Table 3: Neutron multiplication factor between the OpenMC model and benchmark.

Case	$k_{eff}$	Standard Deviation	Difference (pcm)
<b>MCNP 3D</b>	0.99935	0.00014	—
<b>OpenMC 3D</b>	0.99931	0.00002	-4
<b>MCNP 2D</b>	1.03591	0.00014	—
<b>OpenMC 2D</b>	1.03457	0.00002	-134

However, The core eigenvalue is a global parameter and does not capture the distribution of reaction rates across the core. To this end, Table 4 provides a normalized fission power comparison for fuel elements in the 3D models. Columns 2–4 provide measured data reported for fission wire averages within each fuel element in the core [6], OpenMC-calculated powers, and MCNP-computed powers (also from Reference [6]), respectively. The last three columns compare the relative errors between OpenMC calculations and measured data, MCNP calculations and measured data, and OpenMC to the reference MCNP calculation. Table 5 demonstrates a good level of agreement between the OpenMC model and lobe powers (fuel element numbers associated with each lobe are also provided in this Table). Specifically, the relative differences lie within 14.0% and 8.0% when comparing with measurement results and MCNP simulation results, respectively. Both codes are found to underpredict powers in the SE lobe, but OpenMC shows a larger error.

These results are also illustrated in Figure 7. Both the table and the plot show good agreement. However, there seem to be trends in differences between MCNP and OpenMC. To examine this more closely, the difference between OpenMC and MCNP (relative to MCNP) is shown in Figure 8. The figure shows that OpenMC tends to trend low in the SE lobe region and offsets with higher



power predictions in the opposite NW lobe region. This may be a result of an OpenMC modeling difference relative to MCNP in one of those lobes.

Table 4: Normalized power distribution for fuel elements between OpenMC model and benchmark result.

Element No.	Measured	OpenMC	MCNP	% Rel Err (OpenMC-Meas.)	% Rel Err (MCNP-Meas.)	% Rel Err (OpenMC-MCNP)
1	0.0266	0.0279	0.028	4.89	0.36	-0.36
2	0.0247	0.0259	0.026	4.86	0.39	-0.38
3	0.0223	0.0231	0.0229	3.59	-0.87	0.87
4	0.0173	0.0177	0.0177	2.31	0.00	0.00
5	0.0136	0.0147	0.0146	8.09	-0.68	0.68
6	0.0145	0.0145	0.0145	0.00	0.00	0.00
7	0.0171	0.0179	0.018	4.68	0.56	-0.56
8	0.0243	0.0242	0.0246	-0.41	1.65	-1.63
9	0.0261	0.0264	0.0274	1.15	3.79	-3.65
10	0.0277	0.0281	0.029	1.44	3.20	-3.10
11	0.0307	0.0296	0.0305	-3.58	3.04	-2.95
12	0.0324	0.0292	0.0306	-9.88	4.79	-4.58
13	0.0296	0.0271	0.0288	-8.45	6.27	-5.90
14	0.0258	0.0223	0.0236	-13.57	5.83	-5.51
15	0.0206	0.0189	0.0199	-8.25	5.29	-5.03
16	0.0206	0.0191	0.0199	-7.28	4.19	-4.02
17	0.0258	0.0234	0.024	-9.30	2.56	-2.50
18	0.0296	0.029	0.0298	-2.03	2.76	-2.68
19	0.0324	0.0305	0.0314	-5.86	2.95	-2.87
20	0.032	0.031	0.0315	-3.13	1.61	-1.59
21	0.032	0.0317	0.0316	-0.94	-0.32	0.32
22	0.0322	0.0312	0.0311	-3.11	-0.32	0.32
23	0.0309	0.0297	0.0295	-3.88	-0.67	0.68
24	0.025	0.0246	0.0238	-1.60	-3.25	3.36
25	0.0196	0.0204	0.0199	4.08	-2.45	2.51
26	0.0202	0.0205	0.0197	1.49	-3.90	4.06
27	0.024	0.0252	0.0234	5.00	-7.14	7.69
28	0.0285	0.0303	0.0282	6.32	-6.93	7.45
29	0.0307	0.0313	0.0302	1.95	-3.51	3.64
30	0.0297	0.0309	0.0302	4.04	-2.27	2.32
31	0.0283	0.0297	0.0291	4.95	-2.02	2.06
32	0.0273	0.0284	0.0278	4.03	-2.11	2.16
33	0.0246	0.0269	0.026	9.35	-3.35	3.46
34	0.0203	0.0214	0.021	5.42	-1.87	1.90
35	0.0164	0.0174	0.0172	6.10	-1.15	1.16
36	0.0172	0.0173	0.0171	0.58	-1.16	1.17
37	0.0202	0.0212	0.0205	4.95	-3.30	3.41
38	0.0252	0.0257	0.0252	1.98	-1.95	1.98
39	0.0266	0.0275	0.0273	3.38	-0.73	0.73
40	0.0275	0.0284	0.0281	3.27	-1.06	1.07

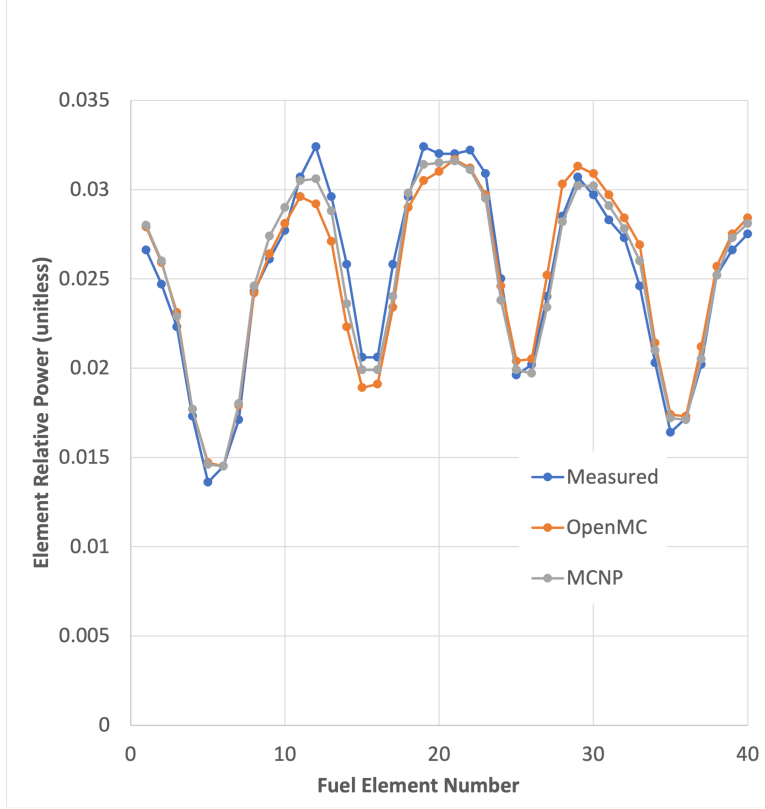


Figure 7: Comparison of measured and calculated relative powers using OpenMC and MCNP.

Table 5: Normalized power distribution by lobe.

Lobe No.	Lobe Fuel Elements	Measured	OpenMC	MCNP (OpenMC-Meas.)	% Rel Err (MCNP-Meas.)	% Rel Err (OpenMC-MCNP)	% Rel Err
NE	3-8	0.1598	0.1644	0.1659	2.88	3.82	-0.90
SE	13-18	0.2167	0.1995	0.2081	-7.94	-3.97	-4.13
NW	23-28	0.1778	0.1858	0.182	4.50	2.36	2.09
SW	33-38	0.2111	0.2132	0.2059	0.99	-2.46	3.55
Center	1-2, 9-12, 19-22, 29-32, 39-40, 39-40	0.2346	0.2373	0.2381	1.15	1.49	-0.34

With confidence in the OpenMC model, we generated multi-group cross sections using the approach described in Sect. 3.1 and then to convert those cross sections to the *ISOXML* format used by Griffin, detailed in Sect. 3.2. We started with a 2D OpenMC model to allow us to work with a simpler finite element mesh for Griffin. The 2D mesh discussed in Sect. 3.3 was used in a Griffin model to verify that Griffin could reproduce the OpenMC eigenvalue. The initial mesh contained  $1.46 \times 10^6$  elements. The OpenMC 2D model calculation yielded an eigenvalue of  $1.03457 \pm 0.00002$ , with Griffin calculating 1.03504 for a 47 pcm difference.

Providing confidence in the Griffin model, we then proceeded to conduct a 2D mesh sensitivity study by reducing the number of mesh cells to determine the relationship between mesh size and the neutron multiplication factor, with results provided in Table 6. The table provides execution times for each mesh size, showing a very linear relationship between mesh size and

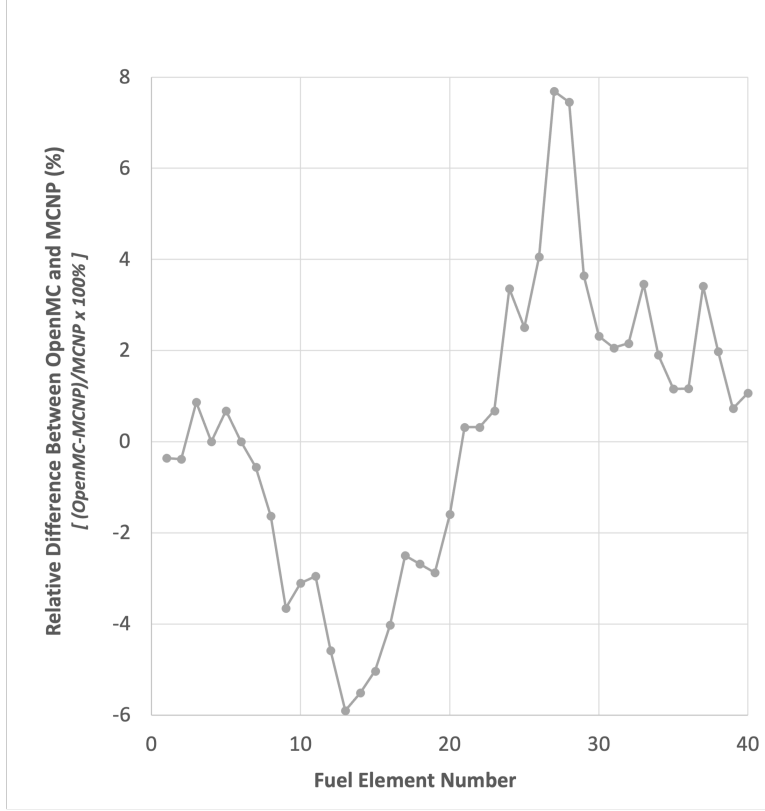


Figure 8: Differences between the calculated relative powers by OpenMC and MCNP by ATR element.

Table 6: Mesh size sensitivity of neutron multiplication factor in ATR 2D case.

Case	# Cell	Execute Time (h)	$k_{eff}$	Dif. (pcm)
01	1,455,946	0.422	1.03504	—
02	648,894	0.194	1.03501	-3.0
03	465,802	0.138	1.03508	4.0
04	272,730	0.079	1.03807	303.0
05	238,076	0.061	1.03737	233.0

computational time. The simulations were performed utilizing 14 processors on INL's Sawtooth supercomputer [9]. Computational time using a the second mesh coarsening step down to 465,802 cells was three times faster, with only around a 4 pcm difference in the neutron multiplication factor compared to the fine mesh. As would be expected, further reduction of the mesh size yielded increasing error. Nevertheless, the results showed a difference of up to 300 pcm in the neutron multiplication factor between the finest and coarsest mesh configurations tested. Notably, the coarse mesh ran seven times faster while still maintaining a reasonable level of accuracy.

Additionally, there was strong agreement between the OpenMC 2D and Griffin models, particularly when using the finely generated mesh from Cubit and mentioned earlier and as shown in Table 7. Here, reported differences in Griffin are relative to the corresponding OpenMC model

for 2D and 3D geometries. It is seen in the table that for the 3D geometry, while using the diffusion solver to speed up the calculations, its performance was poor as it showed a large discrepancy in the neutron multiplication factor when compared to the OpenMC results. This discrepancy will warrant further investigation.

Table 7: Comparison of the computed neutron multiplication factor between Griffin and the reference solution OpenMC for the ATR critical state.

Model	Neutron Multiplication Factor ( $k_{eff}$ )	Dif. (pcm)
Griffin 2D	1.03504	46.7
OpenMC 2D	1.03457±0.00002	—
Griffin 3D	0.92125	7806.0
OpenMC 3D	0.99931±0.00002	—

## 5 CONCLUSIONS AND FUTURE WORK

In this work two OpenMC models of the ATR were created - a full 3D model based on the benchmark specification of Reference [6], and an axially infinite, effectively 2D version based on the geometry at the axial midplane of the fuel region. A comparison of neutron multiplication factors showed very good agreement with benchmark results. A comparison of power distributions in the 3D core models between MCNP and OpenMC shows a small variation in the power distribution, with MCNP closer to the measured data. This difference will be studied in future work. Both OpenMC models were also used to generate 18-group cross sections, and a Python-based script converted the OpenMC HDF5-based cross sections to the *ISOXML* format used by Griffin. Comparisons between the 2D standalone OpenMC model and the Griffin fine mesh model showed a less than 50 pcm difference in the neutron multiplication factor using the diffusion solver option for to accelerate the computations. However, for the 3D geometry case unsatisfactory results were obtained, where the Griffin result exhibited a significant discrepancy in the neutron multiplication factor compared to the OpenMC result for the critical system. The 7800 pcm difference suggests that either (1) an as yet undiscovered error exists in the model or modeling approach, (2) inadequate axial discretization is being used, or (3) a higher order transport model is required to achieve more accurate results.

Research is ongoing to better understand and improve on the 3D power distribution results relative to MCNP and measured data, and to determine the cause of an unacceptably large error in the Griffin 3D simulation of the critical configuration and to improve the model based on findings. Future work will seek to simplify the model by applying SPH factors to allow homogenization of portions of the model to further reduce grid size. This model will then be used to investigate application of the model to enhance simulation of in-core experiments.

## **Acknowledgments**

Authors would like to thank Joshua Hanophy, Mustafa Jaradat and Ishita Trivedi of INL for their assistance and insights for this work. We also wish to thank the ATR Experiment Engineering Division within the Advanced Test Reactor Directorate for providing the funding for Mr. Nguyen's internship. This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under contract no. DE-AC07-05ID14517.

## REFERENCES

- [1] J. E. Sweezy, T. E. Booth, F. B. Brown, J. S. Bull, R. A. F. III, J. T. Goorley, H. G. H. III, R. L. Martz, R. E. Prael, A. Sood, A. J. Zukaitis, R. C. Little, H. R. Trellue, M. C. White, M. B. Lee, and S. M. Girard, "MCNP - Version 5, Vol. I: Overview and Theory," Tech. Rep. LA-UR-03-1987 (Revised 2/1/2008), Los Alamos National Laboratory, 2003.
- [2] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, and K. Smith, "OpenMC: A state-of-the-art monte carlo code for research and development," *Annals of Nuclear Energy*, vol. 82, pp. 90–97, Aug. 2015.
- [3] J. Ortensi, B. A. Baker, M. P. Johnson, Y. Wang, V. M. Labouré, S. Schunert, F. N. Gleicher, and M. D. DeHart, "Validation of the griffin application for treat transient modeling and simulation," *Nuclear Engineering and Design*, vol. 385, p. 111478, 2021.
- [4] Cubit, LLC, "Coreform Cubit (Version 15.7) [Computer software]."
- [5] R. Macfarlane, D. W. Muir, R. M. Boicourt, A. C. Kahler, III, and J. L. Conlin, "The njoy nuclear data processing system, version 2016," 1 2017.
- [6] S. S. Kim and B. G. Schnitzler, "Advanced Test Reactor: Serpentine Arrangement of Highly Enriched Water-Moderated Uranium-Aluminide Fuel Plates Reflected by Beryllium, ATR-FUND-RESR-001," *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, no. NEA/NSC/DOC(95)03, 2021.
- [7] VTT Finland, November 2017. Online, [https://serpent.vtt.fi/mediawiki/index.php/CASMO\\_18-group\\_structure](https://serpent.vtt.fi/mediawiki/index.php/CASMO_18-group_structure).
- [8] Y. Wang, S. Schunert, J. Ortensi, V. Laboure, M. DeHart, Z. Prince, F. Kong, J. Harter, P. Balestra, and F. Gleicher, "Rattlesnake: A MOOSE-based multiphysics multischeme radiation transport application," *Nuclear Technology*, vol. 207, no. 7, pp. 1047–1072, 2021.
- [9] Idaho National Laboratory, "Idaho's third supercomputer coming to collaborative computing center," May 2020. Online, <https://inl.gov/national-homeland-security/idahos-third-supercomputer-coming-to-collaborative-computing-center/>.

## Appendix: Input Files Summary

This appendix describes the various files needed for generating input for the OpenMC model, Cubit mesh, and Griffin model. All files and associated ReadMe files are archived on the INL GitHub repository located at <https://github.inl.gov/mark-dehart/OpenMC-Griffin.git>.

### OpenMC Model Input Generating Files

In order to run the model, OpenMC xml input file must be generated. The process of how to generate those files is previously described in Figure 2 in Section 4. The input generation exploited the Python API, which made it easy to maintain and update as needed. The scripts are:

1. **material\_gen.py**: Collects all material that might be used in the model.
2. **base\_uni.py**: Collects all material universes that might be used in the model.
3. **global\_surface.py**: Stores all global variables for surface that will be used repeatedly.
4. **fuel\_elements.py**: Generates 40 fuel assemblies universes.
5. **fuel\_region.py**: Applies the geometry and assign material for 40 fuel assemblies.
6. **basestructure.py**: Models structural components, such as B holes, A holes, aluminium house, etc.
7. **center.py** and **fluxtrap.py**: Generates models for flux traps in the core layout.
8. **OSCCs\_elements.py**: Models the control elements.
9. **core\_model.py**: Main script for setting simulation parameters, executing the run neutron transport calculation, and generating the multigroup cross sections. By default, the model will be executed with 40 threads and the output for multigroup cross section is named `mgxs.h5`.

Furthermore, a script to convert the cross section into the ISOXML format is also included, named **openmc2MGisoxml.py**.

### Cubit Meshing Input Files

Using Cubit to generate the mesh for ATR requires these scripts:

1. **util2D.py**: Collects all functions utilized in the model.
2. **Global\_var.py**: Stores all global variables for the coordinate, radius, mesh size and meshing scheme for all components.
3. **fuel\_elements.py**: Generates 40 fuel assemblies geometry.
4. **base\_structure.py**: Models the geometries of structural components, such as B holes, A holes, aluminium house, etc.
5. **flux\_trap.py**: Generates geometries for flux traps and the outer shim control cylinders components in the core layout.
6. **main.py**: Main script for creating blocks and mesh generation. By default, the mesh file name is `full_core2D3Dmod.h5`.

## Griffin Model Input Files

The steady-state inputs for Griffin are:

1. **2d.i**: ATR model in 2D geometry.
2. **3d.i**: ATR model in 3D geometry.

To execute the ATR model in Griffin, the pre-generated cross-section files are also included as:

1. **mgxs2D.xml**: Cross section for the ATR model in 2D geometry.
2. **mgxs3D.xml**: Cross section for the ATR model in 3D geometry.

Meshing scripts must be used to generate the base 2D fine-mesh geometry.