LA-UR-22-28341
Approved for public release; distribution is unlimited.

Title: MCNP Unstructured Mesh Verification: Oktavian Models

Author(s): Dzur, Micky Ray
Armstrong, Jerawan Chudoung
D'Angelo, Chelsea Ann

Intended for: Presentation and summary of summer work for several LANL meetings.
Report

Issued: 2022-08-09

MCNP Unstructured Mesh Verification: Oktavian Models

Micky Dzur, Jerawan Armstrong, Chelsea D’Angelo

MCNP Unstructured Mesh Verification: Oktavian Models

Micky Dzur
XCP-3

August 2022

Introduction

- Mentors
  - Jerawan Armstrong (XCP-3)
  - Chelsea D'Angelo (W-13)
- Education
  - Nuclear Engineering Undergraduate Student, Texas A&M University (2020-present)
  - Eldorado High School, Albuquerque, NM (class of 2020)
- Goals
  - Use CUBIT to create unstructured mesh models for verification and testing of the MCNP UM feature.
  - Focus primarily on linear hexahedral element models and the comparison to constructive solid geometry (CSG) as well as linear tetrahedral element models.
  - Use models from OKTAVIAN benchmark experiments to perform analyses.
  - Present at 2022 LANL Student Symposium.
  - Write summary for 2023 ANS Annual Meeting.

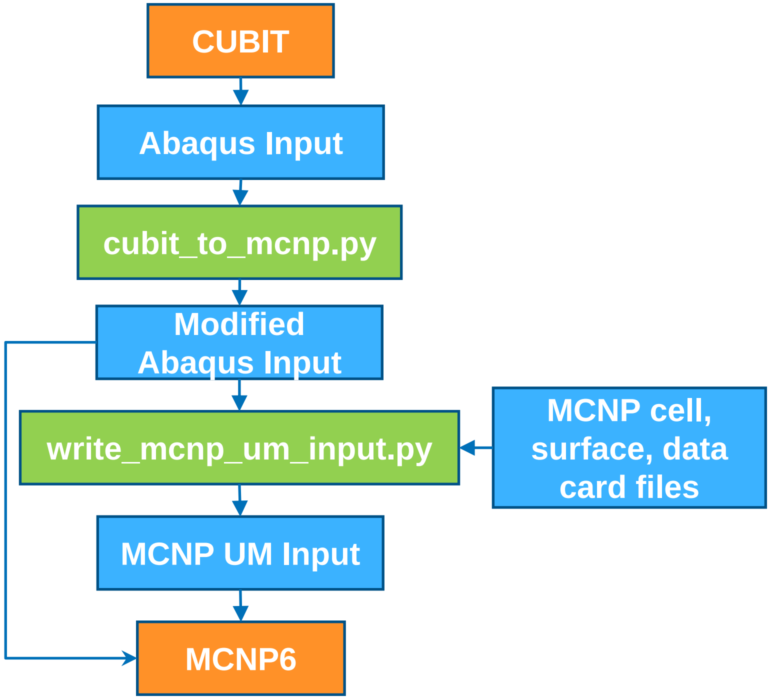
NUCLEAR ENGINEERING
TEXAS A&M UNIVERSITY

MCNP Unstructured Mesh Feature

- The MCNP unstructured mesh (UM) feature was implemented to allow MCNP to read in an Abaqus mesh geometry file, exported by external software packages such as Abaqus and CUBIT.
  - Abaqus is a commercial finite element analysis (FEA) software suite.
  - CUBIT is a mesh generation software developed by Sandia National Laboratory.
- The use of a UM model in MCNP allows for multiphysics simulations such as coupling MCNP with Abaqus.
  - MCNP is used to perform neutronics analysis. Other FEA analysis such as heat transfer can be performed with Abaqus.
- Unstructured meshes can be useful for defining complex geometries that would be otherwise very difficult to create using standard CSG.

MCNP UM Process with CUBIT

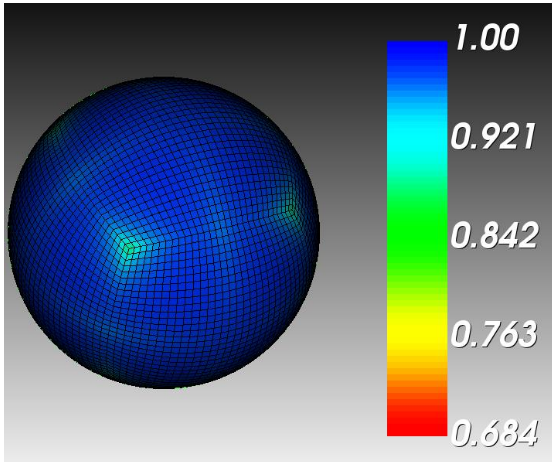
- As CUBIT is used, commands are logged in a journal file that can later be read to replicate the commands. A Python script, that leverages the CUBIT module, was developed to generate an Abaqus-formatted mesh geometry by reading and executing a journal file.
- MCNP requires an Abaqus file as the geometry input in UM problems, however the file created by CUBIT requires changes to be readable by MCNP.
- The Python script CUBIT_TO_MCNP is used to convert this Abaqus file to a format MCNP can process.
- WRITE_MCNP_UM_INPUT is another Python script that uses the mesh information from the Abaqus file along with any user supplied data cards to write an MCNP UM input file.

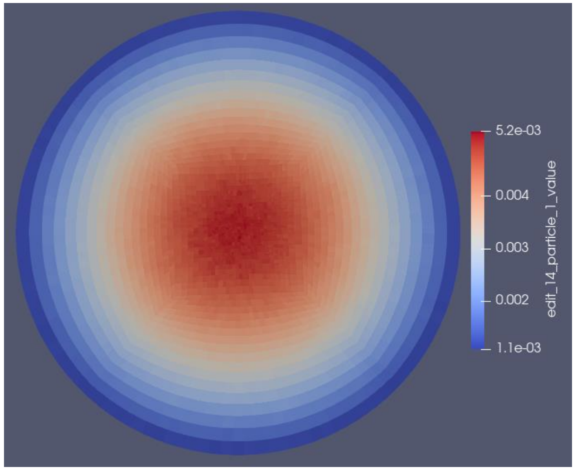


UM Experimentation: GODIVA

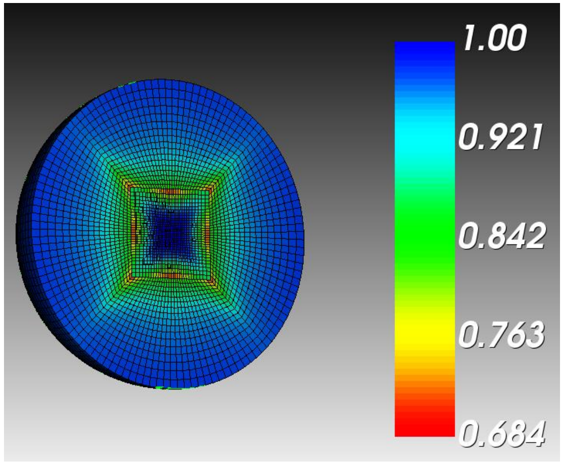
- To gain experience with the MCNP UM feature and to create a potential example problem for CUBIT to MCNP, the Godiva model is used to create a linear hex model.
- Godiva was easily meshed as a linear hex model in CUBIT using the sphere meshing algorithm.
- A CUBIT journal file was used to create an Abaqus file which was then reformatted and used to create an MCNP UM input file.
- The Godiva Abaqus and MCNP input files were used to run MCNP6.3, resulting in an HDF5 and XDMF files that can be visualized in Paraview to analyze the elemental edit outputs.
- Elemental edit outputs can be placed on meshes similar to a cell tally to record flux or energy deposition.
- Keff comparison to CSG:
  - Hex mesh: 0.999435 ± 0.000168, CSG: 1.000026 ± 0.000634

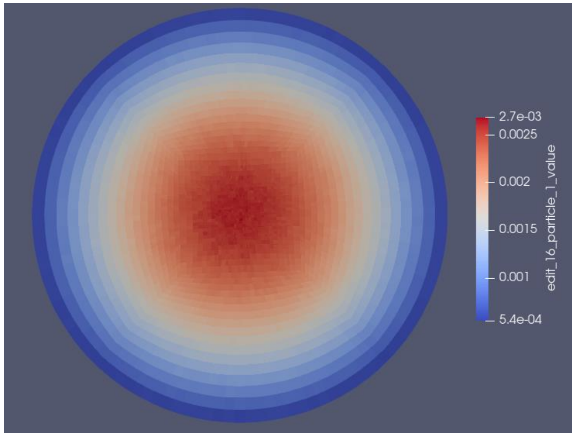
GODIVA Mesh and Results





Energy Deposition





Flux

Oktavian Background

- The Oktavians were shielding experiments consisting of inner and outer spherical steel shells encasing some shielding sample.
- The experiments used a D-T fusion source at the center of the shield, with an aperture leading to a hollow interior containing a tritium target. A deuteron beam was fired through the aperture to strike the target and cause fusion.

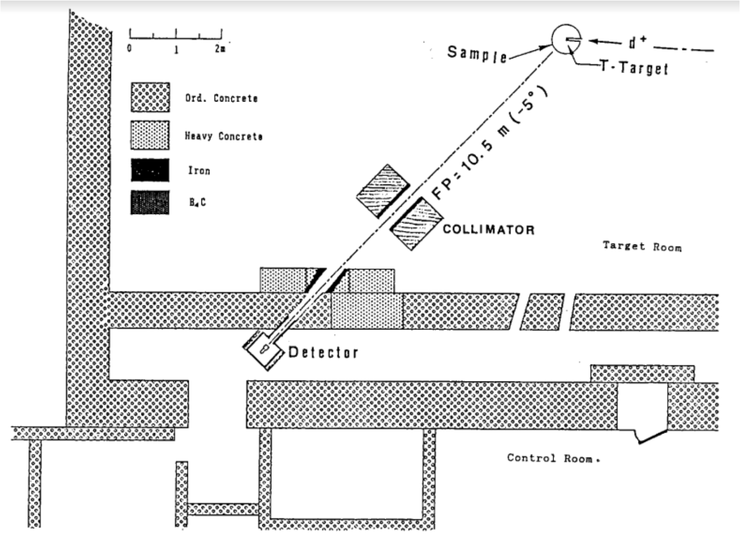


Fig. 1. Experimental arrangement in the OKTAVIAN Facility.

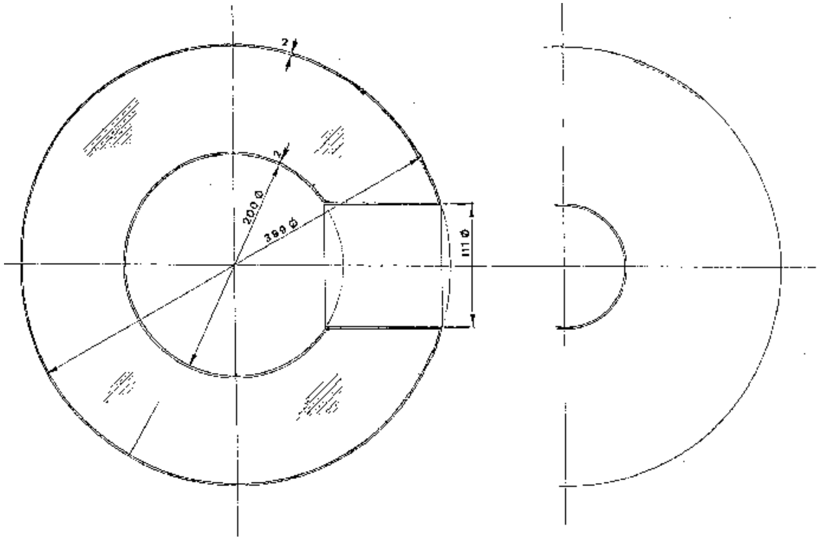
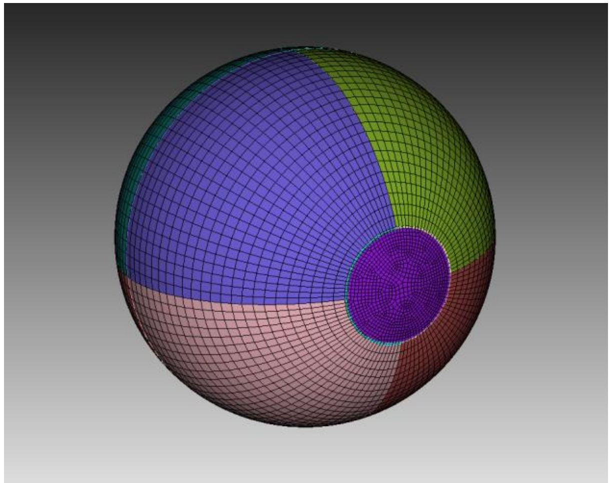
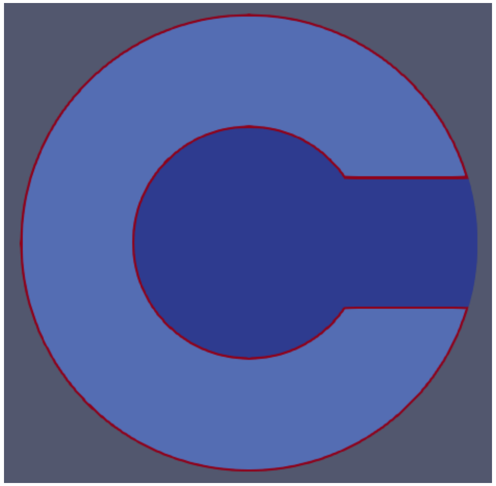


Fig. 2. 40 cm diameter vessel (Type-II).

Initial Meshing

- In general, hex meshing can be difficult, requiring a geometry to be split into simpler components in order to be meshed.
- CUBIT was used to create linear hex Oktavian models.
  - A model must be split into pieces that can be "swept" (i.e. a meshing algorithm that CUBIT employs).
- The meshing method used for the Oktavians was to cut a core out of the center using the cylindrical aperture as a guide so that the remaining spherical components could then be cut into eighths and the entire model then meshed.

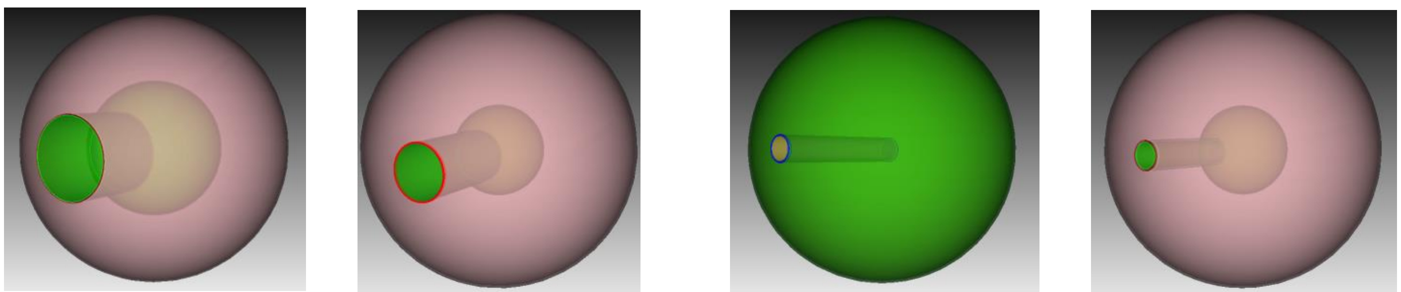




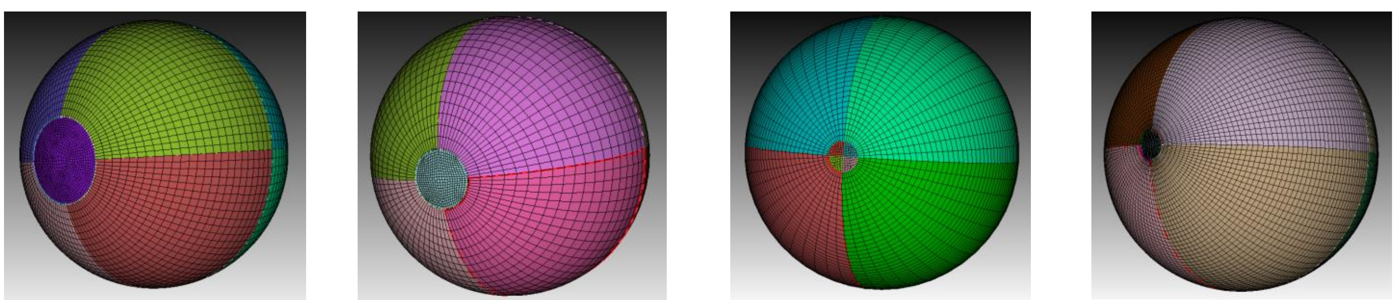
Oktavian geometry cross section, red borders are steel, interior dark blue is air and light blue ring composed of aluminum powder

Verify Linear Hex Oktavian Models

- Use CUBIT to mesh four linear tet and hex Oktavian models
  - It is easier to mesh linear tet models, but these models are computationally more expensive.
- Run MCNP6.3 to compare the tally results from linear tet/hex and CSG models.

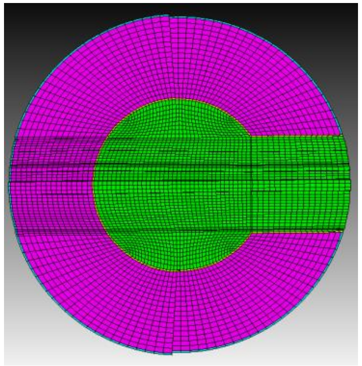


Aluminum Silicon Copper Molybdenum

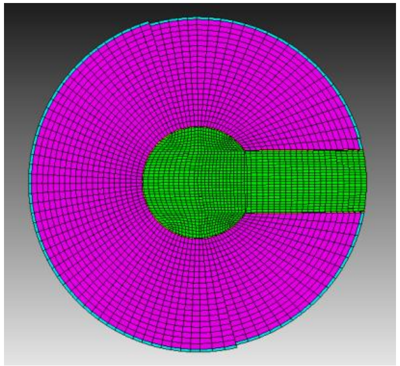


Oktavian UM Models

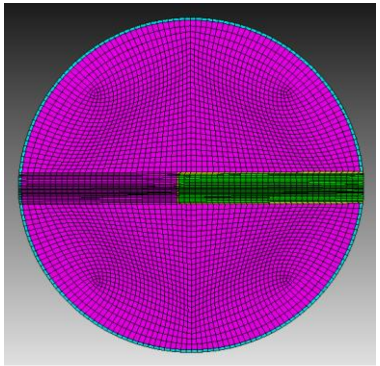
	Al	Si	Cu	Mo
Volume Difference (%)	0.13	0.21	0.27	0.07
Number of Elements	195848	111200	93788	702800
Average Quality	0.85	0.84	0.70	0.80



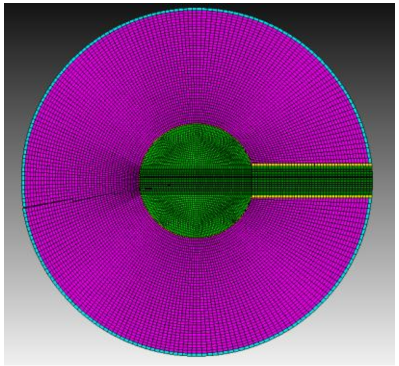
Aluminum



Silicon



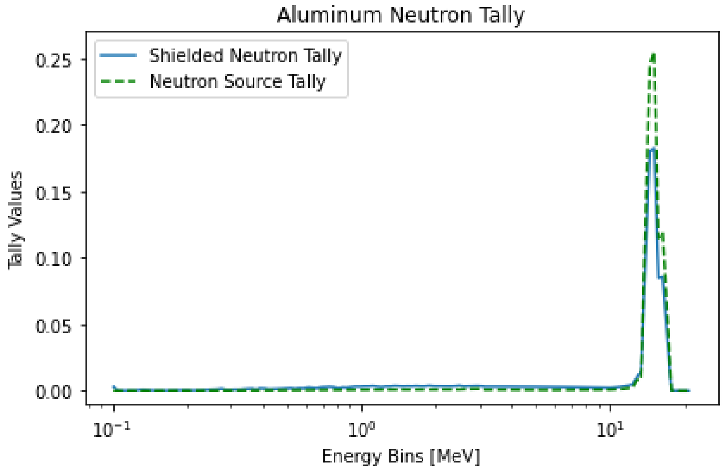
Copper



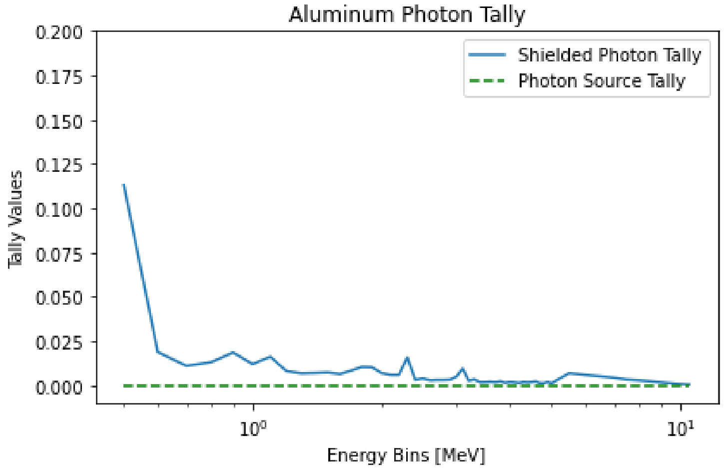
Molybdenum

Aluminum Oktavian Tally Results

Number of histories (NPS): 1E8



F1 neutron tally results compared to source neutrons. Tally on surface just outside the Oktavian.

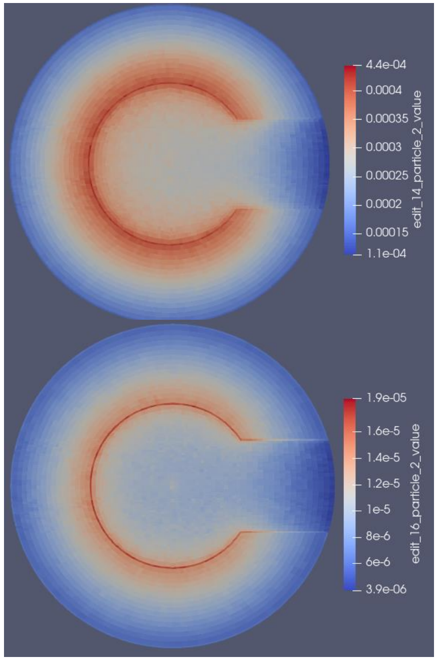


F1 photon tally results. Photons generated from neutron inelastic scattering.

Los Alamos NATIONAL LABORATORY

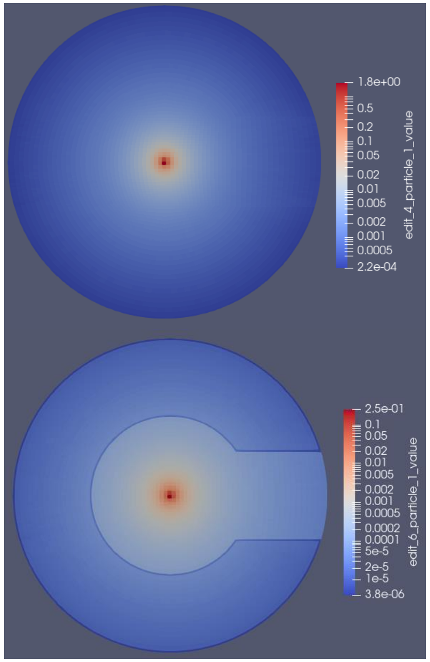
Aluminum Oktavian MCNP Results

- Mesh geometries allow the calculations of elemental edits on elements, functioning similarly to tally over cells.
- HDF5 elemental edit output (eeout) files created by MCNP 6.3 allow results to be viewed using a 3D visualization program such as Paraview.



Photon Flux

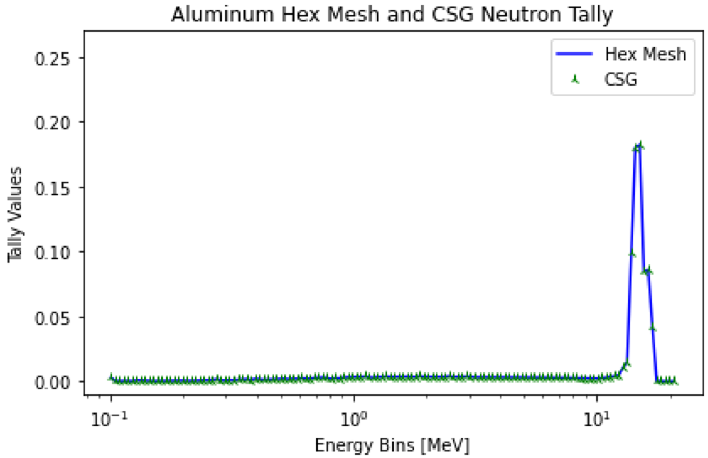
Photon Energy Deposition

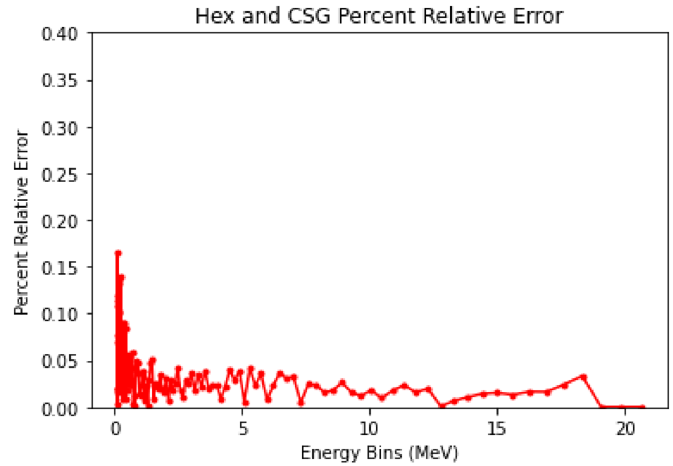


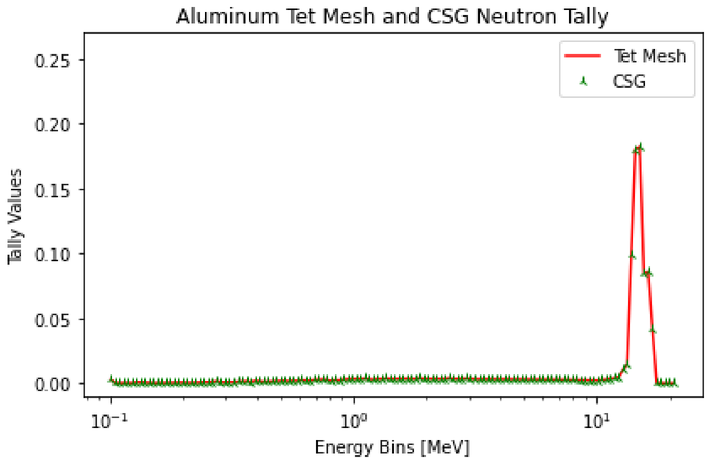
Neutron Flux

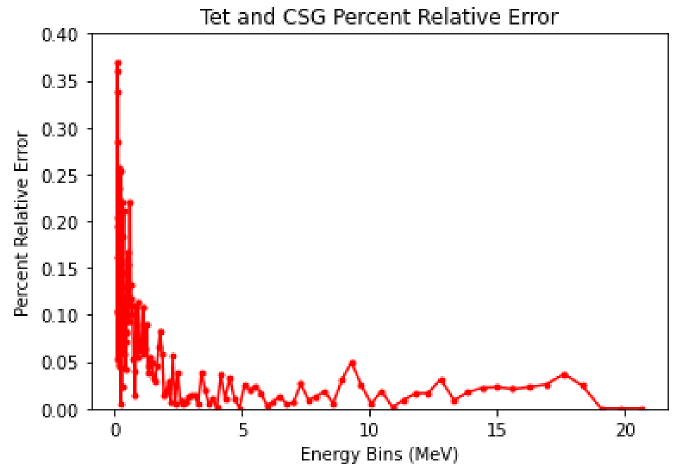
Neutron Energy Deposition

Aluminum Oktavian Comparison F1 Results



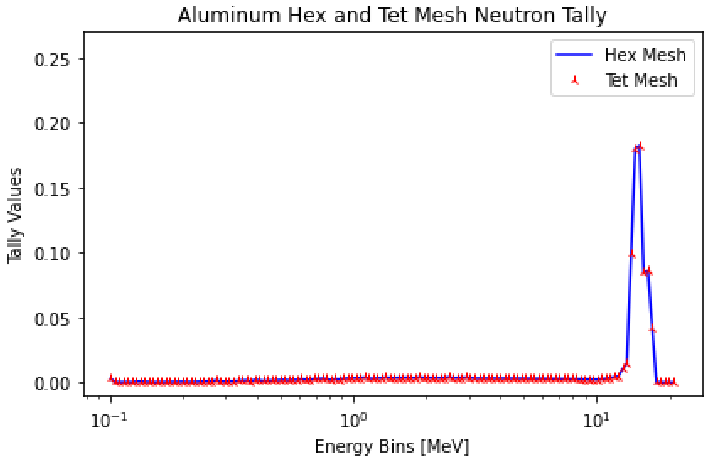


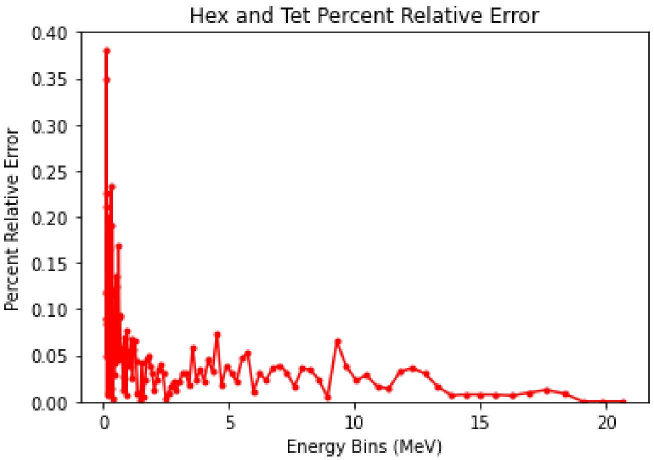




Los Alamos
NATIONAL LABORATORY

Aluminum Oktavian Comparison F1 Results





- MCNP Results and Information
  - Larger error in tet mesh could be due to coarse mesh size in air region of least importance for faster computation. Despite this, tet mesh still has 31% more elements
  - Computer times, CSG: 42.08 minutes Hex mesh: 818.28 minutes, Tet mesh: 1420.7
  - Used 48 processors with MPI on ORGA

Conclusion

- Results
  - The UM feature shows rather low difference when compared to CSG or tetrahedral models
  - The hex mesh seems to provide more accurate and faster results than a tet mesh, although more analysis is needed with fewer variables
  - The UM feature allows analysis and display of otherwise unavailable high fidelity results
- Summer Learning
  - How to use CUBIT
  - Learned MCNP through MCNP class and lots of usage
  - How to use Paraview
  - Gained more experience with Python
- Future Work
  - Embed mesh geometry in CSG universe defining collimators and detector setup reflecting actual experiment
  - Use for more verification comparing CSG to mesh geometry in a hybrid setting under different circumstances
  - Potential for comparison to experimental results to perform validation