OpenMC Visualization & Data Extraction

OpenMC Application to Tokamak Neutronics Analysis Meeting
May 28, 2025

Argonne NATIONAL LABORATORY

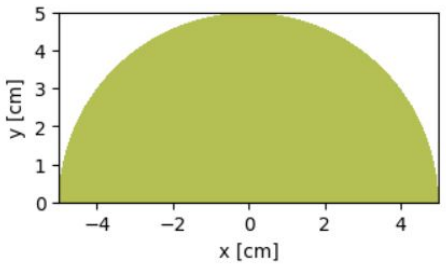
Visualization & Plotting

Inline Plotting

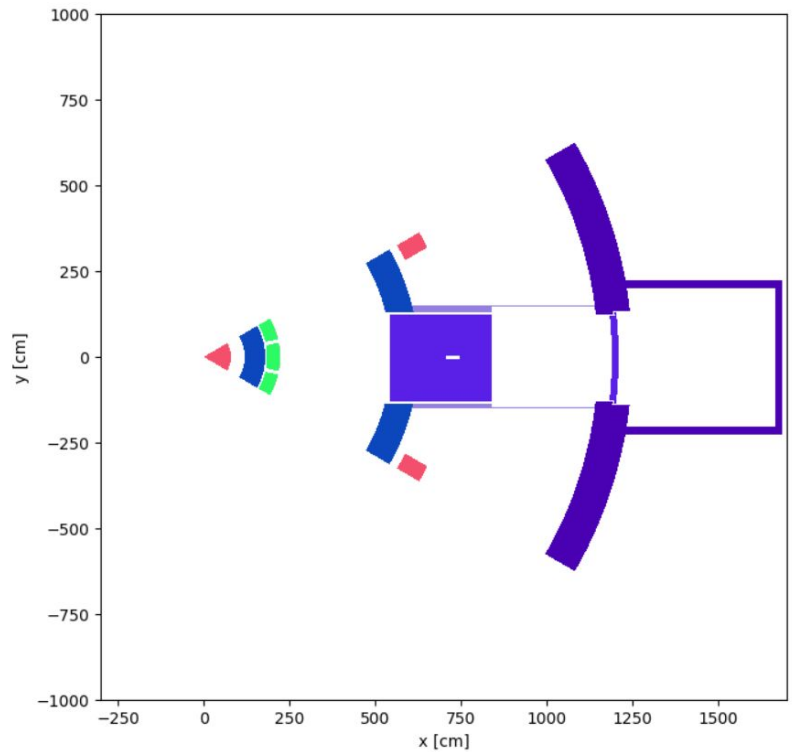
Plots can be generated using many objects:

Region, Cell, Universe, Model

```
[1]: import openmc

[2]: sphere = openmc.Sphere(r=5)
     y_plane = openmc.YPlane(y0=0)
     region = -sphere & +y_plane
     region.plot()

[2]: <Axes: xlabel='x [cm]', ylabel='y [cm]'>
```

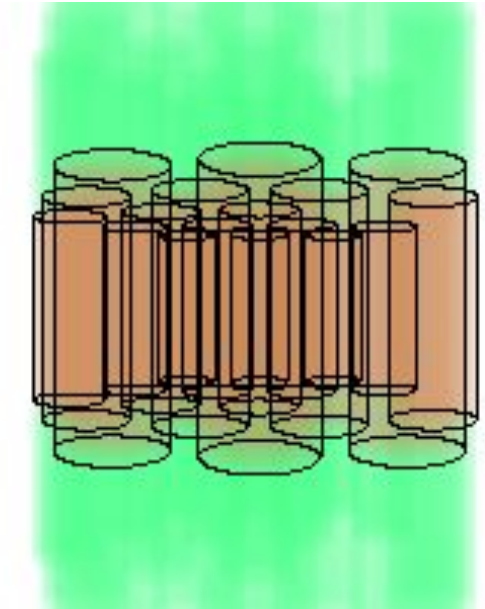



```
[14]: model.plot(color_by='material', width=(2000, 2000), origin=(700, 0, 0), pixels=(600, 600))

[14]: <Axes: xlabel='x [cm]', ylabel='y [cm]'>
```

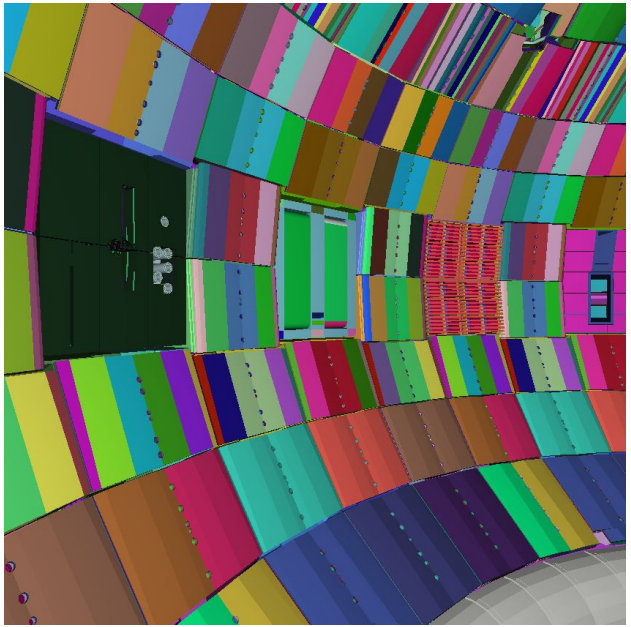


RayTrace Plotting

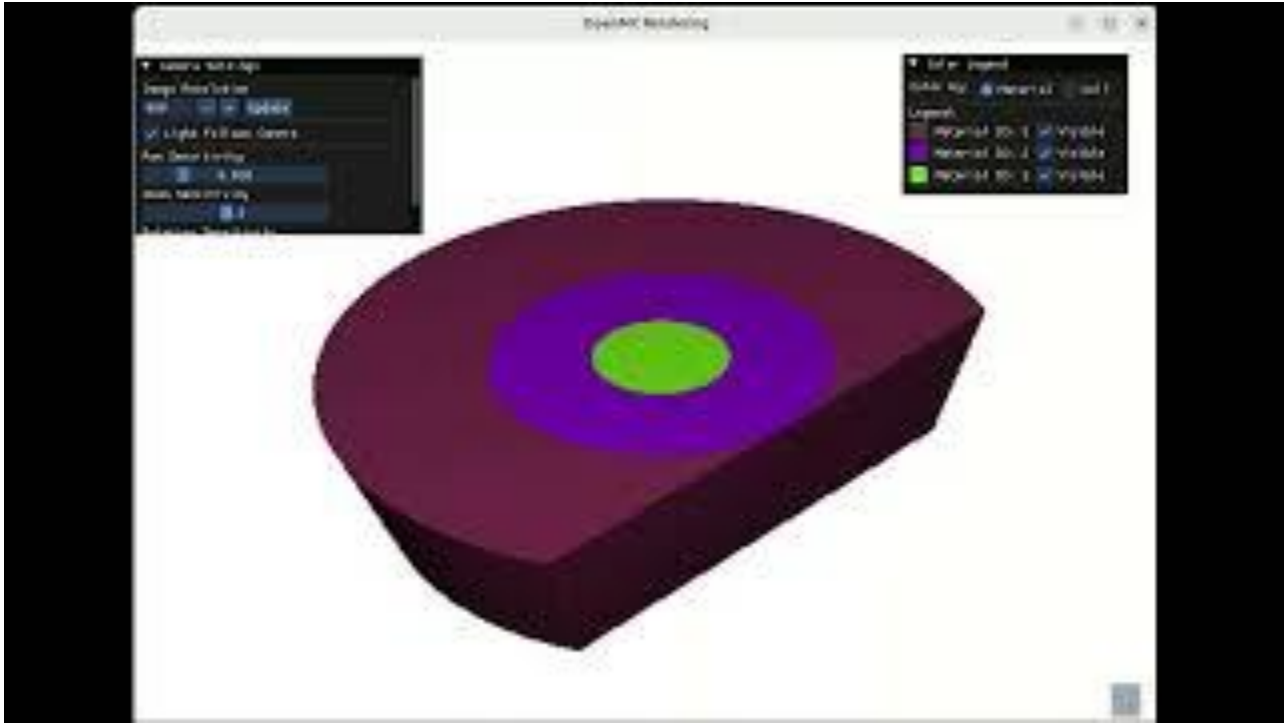
Wireframe Plots



Solid Plots



Native Renderer?



Plotting Utility

OpenMC Plotter

```
$ pip install openmc-plotter
```

Data Extraction

Streamlining Tally Extraction

Input object:

`openmc.Tally`

Output object:

`openmc.Tally`

```
[14]: tally = openmc.Tally()
      tally.scores = ['flux']
      mesh = openmc.RegularMesh()
      mesh.lower_left = (0, -700, -1200)
      mesh.upper_right = (1800, 700, 1200)
      mesh.dimension = (200, 200, 200)

      mesh_filter = openmc.MeshFilter(mesh)
      material_filter = openmc.MaterialFilter(model.materials)

      tally.filters = [mesh_filter, material_filter]

      model.tallies = [tally]

[15]: sp_file = model.run(output=False)

[16]: with openmc.StatePoint(sp_file) as sp:
          new_tally = sp.tallies[tally.id]

[17]: print(new_tally.mean.max())

      0.045387690493174634
```

Streamlining Tally Extraction (v 0.15.2)

Single object for input and output

[18]:
```
tally = openmc.Tally()
tally.scores = ['flux']
mesh = openmc.RegularMesh()
mesh.lower_left = (0, -700, -1200)
mesh.upper_right = (1800, 700, 1200)
mesh.dimension = (200, 200, 200)

mesh_filter = openmc.MeshFilter(mesh)
material_filter = openmc.MaterialFilter(model.materials)

tally.filters = [mesh_filter, material_filter]

model.tallies = [tally]
```

[19]:
```
sp_file = model.run(output=False, apply_tally_results=True)
```

[20]:
```
print(tally.mean.max())
```

```
0.045387690493174634
```

Working with Mesh Data

Shape `openmc.MeshFilter` data into structured indices (when appropriate)

[5]:
```
tally = openmc.Tally()
tally.scores = ['flux']
mesh = openmc.RegularMesh()
mesh.lower_left = (0, -700, -1200)
mesh.upper_right = (1800, 700, 1200)
mesh.dimension = (200, 200, 200)

mesh_filter = openmc.MeshFilter(mesh)
material_filter = openmc.MaterialFilter(model.materials)

tally.filters = [mesh_filter, material_filter]

model.tallies = [tally]
```

[6]:
```
sp_file = model.run(output=False, apply_tally_results=True)
```

[9]:
```
reshaped_data = tally.get_reshaped_data()
reshaped_data.shape
```

[9]:
```
(8000000, 8, 1, 1)
```

[10]:
```
expanded_data = tally.get_reshaped_data(expand_dims=True)
expanded_data.shape
```

[10]:
```
(200, 200, 200, 8, 1, 1)
```

Mesh Visualization

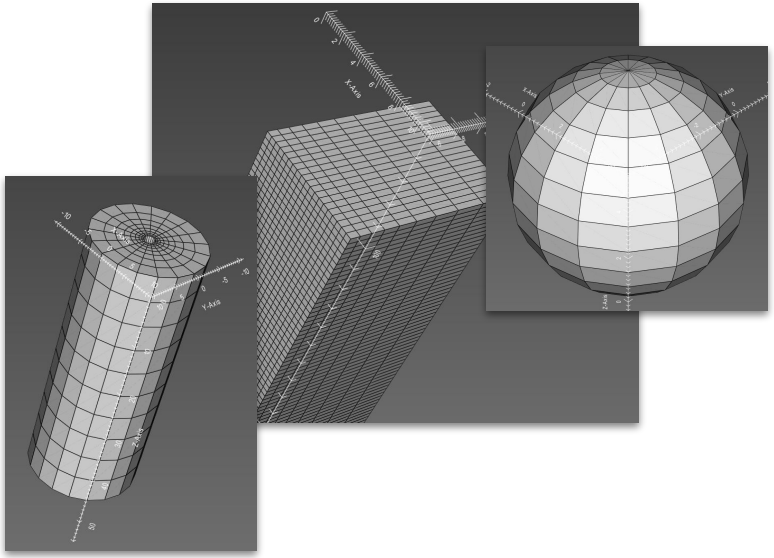
Structured/Periodic/Unstructured Meshes

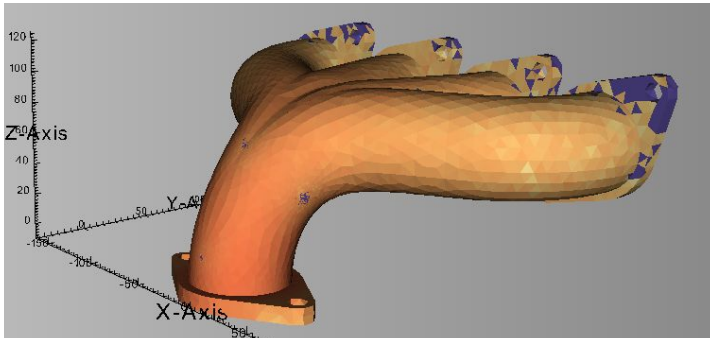
[18]:
```
material_1_data = expanded_data[:, :, :, 0, ...]

mesh.write_data_to_vtk('material_1_flux.vtk',
                       datasets={'mat_1_flux': material_1_data})
```

[18]:
```
<vtkmodules.vtkCommonDataModel.vtkStructuredGrid(0x6340cdc53960)
```

[19]:
```
ls *.vtk
```

```
material_1_flux.vtk
```





A Shame-filled Plug

Home

https://ossfe.org/OSSFE_2026/sponsor

Welcome to OSSFE 2025!

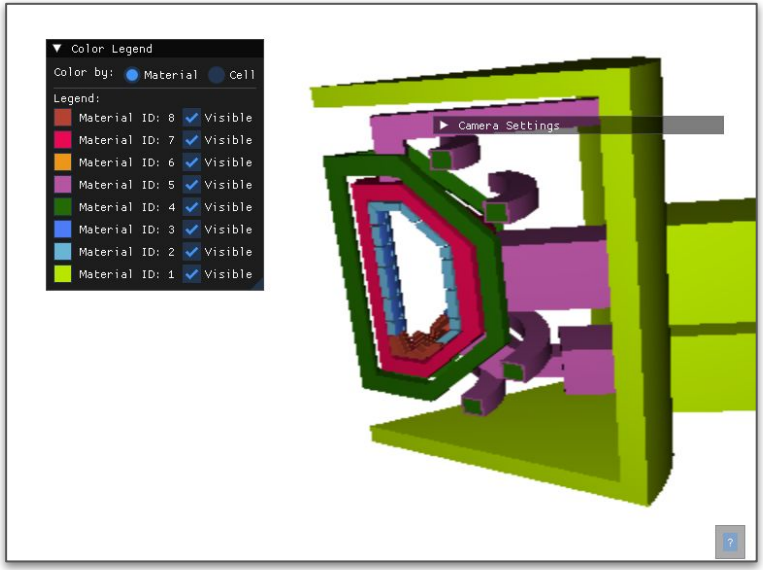
The **Open Source Software for Fusion Energy (OSSFE) Conference** is a first-of-its-kind event dedicated to the development, use, and advancement of open-source software in the fusion energy community. Whether you're an experienced researcher, developer, or new to the field, OSSFE offers a collaborative platform to share insights, tools, and innovations that push the boundaries of fusion energy research.

Registration is now open! more details can be found on the registration page.

Key Dates: Abstract Submission Deadline: January 15, 2025 Registration opens: February 5, 2025 SOLD OUT ! Notification of Acceptance: by February 14, 2025 Conference Dates: March 18, 2025

Acknowledgements

This work was supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences under Contract DE-AC02-06CH11357.



Argonne NATIONAL LABORATORY