Verification and Validation Activities with OpenMC

Paul K. Romano
Computational Scientist, Argonne National Laboratory

Technical Meeting on CoNDERC
October 12, 2022
What is OpenMC?

A community developed, open source Monte Carlo particle transport code, primarily targeted at applications in nuclear science and engineering.
OpenMC: Overview of features

- **Modes**: Fixed source, \( k \)-eigenvalue calculations, volume calculations, geometry plotting
- **Geometry**: Constructive solid geometry, CAD-based, unstructured meshes
- **Solvers**: Neutron and photon transport, depletion
- **Data**: Continuous energy or multigroup cross sections, multipole for Doppler broadening
- **Parallelism**: Distributed/shared-memory via MPI/OpenMP
OpenMC: Unique attributes

What sets OpenMC apart from other codes?

- Programming interfaces (C/C++ and Python)
- Nuclear data interfaces and representation
- Tally abstractions
- Parallel performance
- Development workflow and governance
OpenMC: Community Resources

- Code: https://github.com/openmc-dev/openmc
- Documentation: https://docs.openmc.org
- Nuclear Data: https://openmc.org
- Forum: https://openmc.discourse.group
- Slack: https://join.slack.com/t/openmc/signup
V&V and QA Activities

1. Continuous integration testing
2. Criticality benchmarks
3. Depletion comparisons
4. Shielding benchmarks
5. Automated model conversion
Continuous integration testing

Every time a pull (change) request is made, a comprehensive set of regression and unit tests are run for a matrix of build/run configurations.

Works well when no changes are expected in reference results.
When a change in our reference results are expected (new physics, random number stream, etc.), we kick off a set of benchmark simulations to ensure everything agrees within uncertainty:
ICSBEP Benchmarks

- MIT hosts a collection of **reactor physics benchmark models** for OpenMC, including ~400 ICSBEP models:

<table>
<thead>
<tr>
<th>Fuel</th>
<th>COMP</th>
<th>MET</th>
<th>SOL</th>
<th>MISC</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEU</td>
<td>7</td>
<td>72</td>
<td>48</td>
<td>0</td>
</tr>
<tr>
<td>LEU</td>
<td>6</td>
<td>0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>IEU</td>
<td>1</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PU</td>
<td>1</td>
<td>39</td>
<td>121</td>
<td>0</td>
</tr>
<tr>
<td>U233</td>
<td>1</td>
<td>10</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>MIX</td>
<td>8</td>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fuel</th>
<th>THERM</th>
<th>INTER</th>
<th>FAST</th>
<th>MIXED</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEU</td>
<td>48</td>
<td>11</td>
<td>68</td>
<td>0</td>
</tr>
<tr>
<td>LEU</td>
<td>38</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IEU</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>PU</td>
<td>121</td>
<td>2</td>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>U233</td>
<td>7</td>
<td>1</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>MIX</td>
<td>6</td>
<td>1</td>
<td>28</td>
<td>0</td>
</tr>
</tbody>
</table>
CoNDERC Contributions

• A collection of ICSBEP/IRPhEP benchmark models were contributed to CoNDERC under the “beyond $k_{\text{eff}}$” section

• Emphasis placed on utilizing additional functionality:
  - Plotting flux spectra in various materials
  - Built-in energy group structures
  - Radial flux profile in a spherical mesh
  - Automated volume/mass checks compared to benchmark specifications
  - Complex modeling functionality in OpenMC
  - Use of third-party Python packages
  - Automated generation of parameterized models
We also maintain a repository of validation scripts
- `openmc-run-benchmarks` — Run a collection of ICSBEP benchmarks using either OpenMC or MCNP and collect results
- `openmc-plot-benchmarks` — Plot results from `openmc-run-benchmarks`

Running hundreds of benchmarks is as simple as executing a single command (and having a big computer!)
OpenMC 0.13.1 vs MCNP 6.2 on ICSBEP benchmarks:
Validation scripts also include simple physics comparisons:

- Neutron/photon spectrum from a monoenergetic point source in a sphere
- Secondary photon spectrum in broomstick model (coupled n–γ)
Depletion Code-to-code Comparison

Extensive code-to-code comparisons for depletion have been carried out with OpenMC and Serpent.
SINBAD Benchmarks

- As part of a new project funded by DOE/Fusion Energy Sciences, OpenMC models for SINBAD benchmarks are being developed.
- First one is “FNG dose”:
FNG Dose: Early Results

![Graph showing dose rate over time]

**Dose rate [Sv/h]**

- **Time [d]**
  - 10
  - 6
  - 5
  - 4

**Experiment**

- OpenMC
Recently, we released a utility for converting MCNP models to OpenMC.

Unlike `csg2csg`, directly utilizes OpenMC’s Python API.

Some MCNP features are not supported:
- Periodic boundary conditions
- Hexagonal lattices
- Some macrobodies
- Source definitions, tallies, and most cards in the “data” section
Model Conversion: ITER E-lite
• General capability for model conversion opens up new opportunities:
  • Maintaining a single “canonical” set of inputs
  • Minimizing/eliminating model differences
  • Volume comparisons
  • Mass comparisons

• Many code-to-code differences (esp. outliers) I’ve seen tend to be a result of modeling errors or inconsistencies

• Study different approaches in csg2csg, t4_geom_convert, and openmc_mcnp_adapter to learn from one another
Thank you!
Acknowledgments

• This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation’s exascale computing imperative.

• This material is based upon work supported by the US Department of Energy, Office of Science, Office of Fusion Energy Sciences under Award DE-AC02-06CH11357.

• We gratefully acknowledge the computing resources provided on Bebop, a high-performance computing cluster operated by the Laboratory Computing Resource Center at Argonne National Laboratory.