Summary report for special agreement TAL-NAP20220323-005

prepared by
Paul K. Romano

July 2022

1 Introduction

The goal of this special agreement was to generate a series of OpenMC [1] models and associated input/output files for benchmark problems for the Compilation of Nuclear Data Experiments for Radiation Charactisation (CoNDERC) project. The "Beyond Keff" section of the CoNDERC webpage currently features benchmark inputs and outputs for both the MCNP [2] and TRIPOLI [3] codes. Adding OpenMC models to the CoNDERC database will give the community another option for exploring code-to-code comparisons and will further support nuclear data validation and verification (V&V) activities.

OpenMC is a community developed, open-source Monte Carlo (MC) neutron and photon transport code. It has been under active development for over 10 years and has a wide array of features. OpenMC is capable of performing fixed source, k-eigenvalue, and subcritical multiplication calculations. Solid models are built using either a constructive solid geometry or CAD representation. Finally, it features a rich Python application programming interface (API) that enables programmatic pre- and post-processing, multigroup cross section generation, coupled transport–depletion calculations, and analysis/conversion of nuclear data files.

2 Benchmark Model Preparation

To run OpenMC, the user is required to produce a set of XML files:

- `materials.xml` — describes the material definitions;
- `geometry.xml` — describes the volumes in the model and what materials are associated with them;
- `settings.xml` — describes simulation settings, e.g., the number of particles to simulate; and
- `tallies.xml` (optional) — specifies physical quantities that should be tallied during execution.

Although one could in theory write each of the XML files manually, in practice they are typically generated programmatically using OpenMC's Python API. Thus, instead of writing XML files, the user writes a Python script that generates the necessary XML files.

For each of the benchmark models developed under this project, a Python script named `make_model.py` was written that generates the aforementioned XML files. To the extent possible, tallies are specified to match the tallies for equivalent MCNP models already available in the CoNDERC database. These tallies include:

- A CCFE 709-group flux spectrum in important regions (example in Fig. 1)
- A radial flux profile across fissionable regions (example in Fig. 2)

In addition to the `make_model.py` script, some of the benchmark models are accompanied by a `postprocess.py` script that reads the OpenMC output (statepoint file) and produces plots based on the tallies that were present. This provides users of the CoNDERC database some typical examples of how to post-process and visualize tally information produced by OpenMC.

Many of the Python scripts showcase the power and utility of OpenMC's Python API. Some examples of this include the following:

- When specifying the energy group structure for a flux spectrum tally, the scripts make use of the `openmc.EnergyFilter.from_group_structure` method, which enables the user to select a group structure by name ("CCFE-709" in this case) rather than explicitly listing the group boundaries.

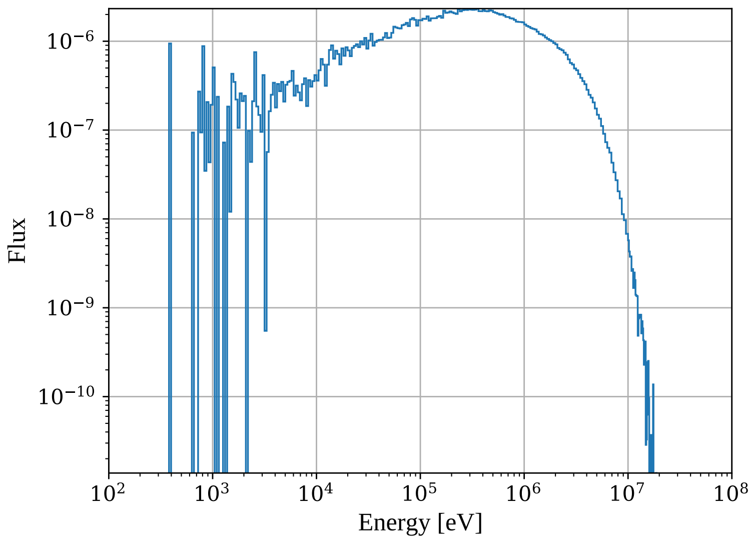

Figure 1: Neutron flux spectrum for the PU-MET-FAST-001 benchmark model as calculated by OpenMC.

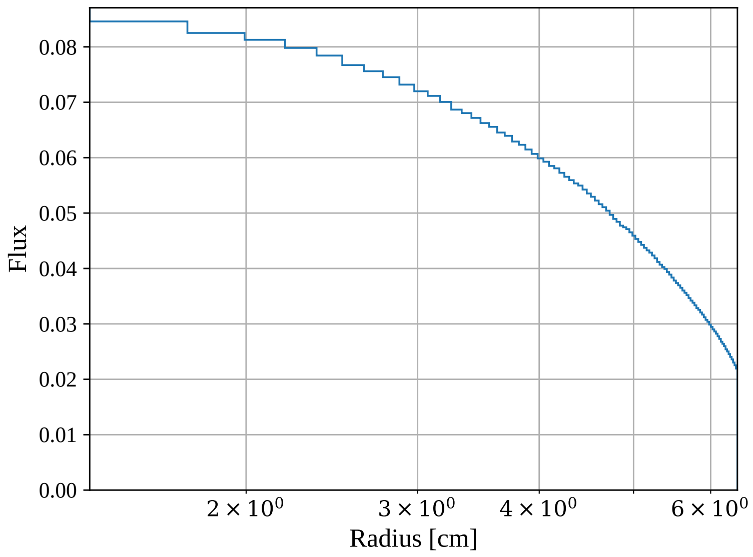

Figure 2: Radial flux profile for the PU-MET-FAST-001 benchmark model as calculated by OpenMC.

Table 1: Benchmarks from ICSBEP and IRPhEP for which OpenMC models were created.

Benchmark	**Cases**	**Description**
DIMPLE-LWR-EXP-001	1	Light water moderated/reflected low-enriched UO_2 rod lattices
DIMPLE-LWR-EXP-002	2	Light water moderated/reflected low-enriched UO_2 rod lattices
HEU-MET-FAST-001	2	Bare, highly enriched uranium sphere (Godiva)
HEU-MET-FAST-028	1	^{235}U sphere reflected by normal uranium using Flattop
IEU-MET-FAST-007	1	Big Ten, A large mixed-uranium-metal cylindrical core with 10% ^{235}U enrichment, surrounded by a thick ^{238}U reflector
PU-MET-FAST-001	1	Bare sphere of ^{239}Pu metal (Jezebel)
PU-MET-FAST-002	1	Bare sphere of ^{239}Pu metal (20.1 at% ^{240}Pu, 1.01 wt% Ga)
PU-MET-FAST-006	1	Plutonium sphere reflected by normal uranium using Flattop
PU-MET-FAST-008	2	Critical experiment of a thorium reflected plutonium sphere
TCA-LWR-EXP-001	39	Temperature effects on reactivity in light water moderator UO_2 core with soluble poisons at TCA
U233-MET-FAST-001	1	A bare sphere of ^{233}U metal (Jezebel)
U233-MET-FAST-006	1	^{233}U sphere reflected by normal uranium with Flattop
ZEBRA-FUND-RESR-001	1	K-infinity measurements in ZEBRA core 8

- Some of the scripts make use of the `openmc.Material.get_mass` method to do a sanity check on the quantity of fissionable material present in the model, which can be checked against information from the benchmark specification.

- Several scripts use Python functions to repeat common operations, e.g., to define a cell that is bounded by two cylinders and two planes in the IEU-MET-FAST-007 benchmark.

- Use of the NumPy [4] third-party package to define various grids, e.g., a radial grid for a spherical mesh tally that results in equal-volume mesh elements.

- The script written to generate models for the TCA-LWR-EXP-001 benchmark creates inputs for all 39 configurations in one go. These models also exercise OpenMC's ability to interpolate between cross section datasets at multiple temperatures.

2.1 Benchmark Selection

OpenMC models were built for a variety of benchmarks from the ICSBEP [5] and IRPhEP [6] handbooks. Benchmark selection was based on the currently available MCNP models on CoNDERC site. Many of these are benchmark models for which reaction rate or pin power data were available. The final list of benchmarks for which OpenMC models were built is shown in Table 1.

2.2 Nuclear Data Preparation

OpenMC relies on its own HDF5 format for representing nuclear data. These HDF5 files can be produced directly from an ACE file using the `openmc.data` module from OpenMC's Python API. Additionally, ENDF files can be converted to HDF5 using `openmc.data` (NJOY [7] is seamlessly executed under the hood to first produce an ACE file).

For each benchmark model, three runs were carried out with each of the following nuclear data libraries:

1. **ENDF/B-VIII.0** — This library was produced by converting ACE files from the Lib80x and ENDF80SaB libraries distributed by Los Alamos National Laboratory [8]. This library includes incident neutron cross sections at 0 K, 250 K, 293.6 K, 600 K, 900 K, 1200 K and 2500 K.

2. **TENDL-2021** — This library was produced by converting ACE files distributed on the TENDL website (`https://tendl.web.psi.ch/tendl_2021/tendl2021.html`). For benchmark models that contain light water as a moderator, the light water thermal scattering evaluation from JEFF-3.3 was used. The source ACE files are provided only for a single temperature, 293.6 K.

3. **JENDL-5** — This library was produced from the original ENDF files on the JENDL-5 website (`https://wwwndc.jaea.go.jp/jendl/j5/j5.html`) by running them through the `openmc.data` module, executing NJOY as part of the process. Note that NJOY version 2016.67 was used, which contains important updates that are needed in order to process JENDL-5. Problems were encountered with NJOY when trying to generate ACE data for some nuclides at multiple temperatures; consequently, this library was generated for only a single temperature, 293.6 K.

When a model specifies a material composition in terms of element densities rather than individual nuclide densities, OpenMC allows a user to directly specify the density of the element. In this case, the element will be automatically expanded into its constituent isotopes based on the natural abundances. When this expansion is performed, OpenMC will also look at what nuclides are available in the cross section library and select them accordingly. For example, when carbon is added to a material and the cross section library is ENDF/B-VII.1, it will automatically add the elemental evaluation of carbon (since ENDF/B-VII.1 does not have individual evaluations for ^{12}C and ^{13}C). Consequently, the XML input files produced may be different for each data library as they only contain nuclide densities, not element densities.

3 Results

For each benchmark case, an OpenMC simulation was executed using a developmental version[1] of OpenMC. This version has equivalent physics to the latest release of OpenMC (version 0.13.0 [9]) but has several usability enhancements and interface bug fixes. Each simulation was run with 1 million neutrons per batch, 50 inactive batches, and 500 active batches. Thus, each benchmark case had 500 million active neutron histories. This number of neutrons was sufficient to converge the k_{eff} value to a statistical uncertainty of 5 pcm or less. Table 2 shows the k_{eff} value calculated by OpenMC for each benchmark case for the three nuclear data libraries. Note that TCA-LWR-EXP-001 was excluded from Table 2 since it was simulated only with the ENDF/B-VIII.0 library. While some of the results between libraries are in close agreement, others merit further investigation (e.g., there is a ∼600 pcm difference between ENDF/B-VIII.0 and JENDL-5 on the ZEBRA-FUND-RESR-001 benchmark).

[1] git SHA1 hash: b72c4593a328d9a16f6d3ab1285854527d4c1045

Table 2: k_{eff} as calculated by OpenMC for all benchmark cases other than TCA-LWR-EXP-001.

Benchmark	**Case**	**ENDF/B-VIII.0**	**JENDL-5**	**TENDL-2021**
DIMPLE-LWR-EXP-001	1	1.08720 (4)	1.08732 (4)	1.09063 (4)
DIMPLE-LWR-EXP-002	S06A	1.09411 (4)	1.09388 (4)	1.09816 (4)
DIMPLE-LWR-EXP-002	S06B	1.08101 (4)	1.08093 (4)	1.08583 (4)
HEU-MET-FAST-001	1	1.00009 (3)	0.99917 (3)	1.00102 (3)
HEU-MET-FAST-001	2	1.00006 (3)	0.99921 (3)	1.00102 (3)
HEU-MET-FAST-028	1	1.00087 (3)	1.00105 (4)	1.00495 (3)
IEU-MET-FAST-007	1	1.00424 (3)	1.00020 (3)	1.00518 (3)
PU-MET-FAST-001	1	1.00064 (3)	1.00040 (3)	1.00046 (3)
PU-MET-FAST-002	1	1.00145 (3)	0.99911 (3)	1.00376 (3)
PU-MET-FAST-006	1	0.99969 (4)	1.00174 (4)	1.00363 (3)
PU-MET-FAST-008	1	0.99789 (3)	0.99953 (3)	0.99740 (3)
PU-MET-FAST-008	2	0.99745 (3)	0.99902 (3)	0.99698 (3)
U233-MET-FAST-001	1	1.00041 (3)	0.99957 (3)	1.00101 (3)
U233-MET-FAST-006	1	0.99991 (4)	1.00051 (3)	1.00342 (4)
ZEBRA-FUND-RESR-001	1	1.02311 (3)	1.01702 (3)	1.02573 (3)

Results for the TCA-LWR-EXP-001 benchmark are shown separately in Table 3 for each core configuration and critical temperature. Most of the calculated k_{eff} values are quite close to unity. However, there are some exceptions. In particular the C-series cores had calculated k_{eff} values that were several hundreds of pcm away from 1.0. Again, further investigation of these particular cases is warranted.

For each benchmark case that was simulated, several input/output files were generated. A description of each of these files is as follows:

- `geometry.xml`, `materials.xml`, `settings.xml`, `tallies.xml` — These are the XML input files that are generated by `make_model.py` and were described in Section 2.
- `tallies.out` — This file is an ASCII output file that lists tally results.
- `statepoint.*.h5` — This HDF5 file contains all the OpenMC results including user-defined tallies, global tallies such as k_{eff}, as well as other information such as execution time, date/time of the simulation, the OpenMC version, etc. This file can be analyzed using the `openmc.StatePoint` Python class.
- `summary.h5` — This HDF5 files contains a representation of the model (geometry and materials) and can be analyzed using the `openmc.Summary` Python class. `*.stdout` — This file is a log of the information that was printed to standard output during the simulation.

4 Conclusions

This project expands the CoNDERC database by including new OpenMC models for a variety of benchmarks from the ICSBEP and IRPhEP handbooks. Simulations have been executed with three of the most recent nuclear data libraries: ENDF/B-VIII.0, TENDL-2021, and JENDL-5. While good agreement was observed for many benchmarks, there are quite a few cases where larger differences merit further investigation. There are several well known deficiencies in the ENDF/B-VIII.0 library that have since

Table 3: k_{eff} as calculated by OpenMC for TCA-LWR-EXP-001 with the ENDF/B-VIII.0 library.

Core	**Temperature [°C]**	k_{eff}
A-1a	25.5	0.99869 (5)
	34.7	0.99898 (4)
	45.6	0.99936 (4)
	60.0	0.99984 (4)
A-2a	26.6	0.99903 (4)
	38.5	0.99931 (4)
	48.4	0.99944 (4)
	59.8	0.99961 (5)
A-3	14.9	0.99942 (4)
	27.0	0.99949 (5)
	43.5	0.99978 (4)
	56.0	0.99998 (4)
B-1	25.2	0.99945 (4)
	43.2	0.99937 (4)
	51.7	0.99969 (4)
	58.9	1.00003 (4)
B-2	27.4	0.99864 (4)
	38.1	0.99872 (4)
	48.1	0.99890 (5)
	57.7	0.99917 (4)
B-3	27.2	0.99905 (4)
	38.8	0.99921 (4)
	48.7	0.99941 (4)
	56.9	0.99983 (4)
C-1	22.8	0.99492 (4)
	34.1	0.99504 (4)
	43.7	0.99535 (5)
	53.3	0.99564 (4)
	62.4	0.99583 (4)
C-2	16.4	1.00196 (4)
	26.9	1.00169 (4)
	39.9	1.00201 (4)
	46.5	1.00230 (4)
	54.2	1.00250 (4)
C-3	20.3	0.99677 (4)
	30.4	0.99693 (4)
	39.2	0.99723 (4)
	49.5	0.99765 (4)
	59.8	0.99815 (4)

been addressed in candidate evaluations for the upcoming ENDF/B-VIII.1 library that resulted from collaboration through the International Nuclear Data Evaluation Network (INDEN).

Beyond the comparison of results with several nuclear data libraries, the models produced through this project also help to showcase some of the unique features of OpenMC for programmatic generation of constructive solid geometry models for Monte Carlo simulation and subsequent analysis of results, taking advantage of the Python scientific computing ecosystem,

References

[1] P. K. ROMANO, N. E. HORELIK, B. R. HERMAN, A. G. NELSON, and B. FORGET, "OpenMC: A state-of-the-art Monte Carlo code for research and development," *Ann. Nucl. Energy*, **82**, 90 (2015); https://doi.org/10.1016/j.anucene.2014.07.048.

[2] T. GOORLEY et al., "Initial MCNP6 Release Overview," *Nucl. Technol.*, **180**, *3*, 298 (2012); https://doi.org/10.13182/NT11-135.

[3] E. BRUN et al., "TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code," *Ann. Nucl. Energy*, **82**, 151 (2015); https://doi.org/10.1016/j.anucene.2014.07.053.

[4] C. R. HARRIS et al., "Array programming with NumPy," *Nature*, **585**, *7825*, 357 (2020); https://doi.org/10.1038/s41586-020-2649-2.

[5] NEA Nuclear Science Committee, "International Handbook of Evaluated Criticality Safety Benchmark Experiments," NEA/NSC/DOC(95)03, OECD Nuclear Energy Agency (2015).

[6] J. D. BESS, T. IVANOVA, I. HILL, and L. SCOTT, "The 2019 Edition of the IRPhEP Handbook," *Trans. Am. Nucl. Soc.*, **121**, *1*, 1565 (2019).

[7] R. MACFARLANE, D. W. MUIR, R. M. BOICOURT, A. C. KAHLER, and J. L. CONLIN, "The NJOY Nuclear Data Processing System, Version 2016," LA-UR-17-20093, Los Alamos National Laboratory (2017).

[8] J. CONLIN, W. HAECK, D. NEUDECKER, D. K. PARSONS, and M. C. WHITE, "Release of ENDF/B-VIII.0-Based ACE Data Files," LA-UR-18-24034, Los Alamos National Laboratory (2018); https://doi.org/10.2172/1438139.

[9] P. K. ROMANO et al., "OpenMC 0.13.0," 2022; https://doi.org/10.5281/zenodo.6079155.