OpenMC Course Introduction

NEA OpenMC Course
November 24, 2025

NEA
NUCLEAR ENERGY AGENCY

OpenMC

Course Logistics

Course Goals

- Provide a solid understanding of how to use OpenMC and its many features
- Understand the unique features and advantages of OpenMC relative to other software
- Build problem solving skills: working through difficult problems, understanding error messages, etc.
- Have an enjoyable time!

Logistics

- This course is primarily **interactive**: mix of live demonstrations, exercises, and open discussion
- Daily schedule:
  - 9am-3:45pm — Live demonstrations
  - Morning coffee break, lunch, afternoon coffee break
  - 3:45-5pm — exercises, discussion, quizzes
- Instructors:
  - Paul Romano
  - April Novak
- Social dinner on Tuesday at 7 pm at Attabler

Interactive Sessions

- You will be using **Jupyter Lab** for demonstrations and in-class exercises
- Instructor will give live demo for each session, and you can follow along in your own Jupyter Lab instance
- You don’t need to have OpenMC installed on your system, but if you prefer to use your installation, feel free to!
- The URL provided to you will be available all week but will be shutdown at the end of the week — “notebooks” can be downloaded at any time

Quizzes, Exercises, and Exam

- For the last session of the day, there will be:
  - Exercises to reinforce learning and put your knowledge to use
  - Quizzes on the material we went over

- Quizzes can be taken as many times as you want and provide feedback on areas you might want to review, and help prepare for final exam

- At the end of the course:
  - There will be an exam you can take to receive a certificate
  - Material will be draw from end-of-day quizzes
  - One week to take, 70% to pass
  - You are allowed one re-do

OpenMC Intro

What is OpenMC?

- Community developed, open source **Monte Carlo** neutron–photon transport code, primarily target at applications in nuclear science and engineering

- Monte Carlo method: directly simulate life of individual particles using known probability distributions

- Project objectives:
  - **Open source** contribution model
  - **Extensible** for research purposes
  - Adopt software development **best practices**
  - **Ease** of use
  - High **performance** and scalable

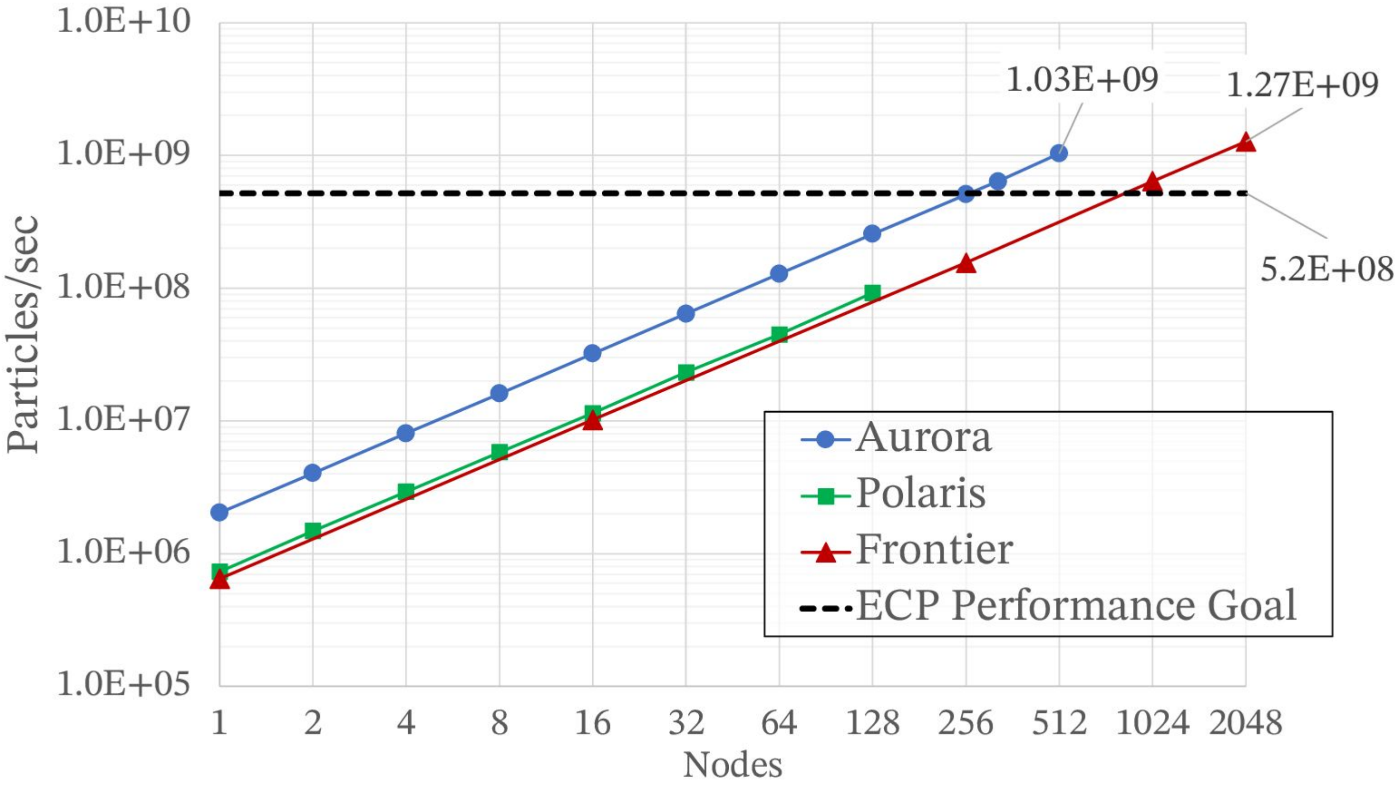
High-level features

- Monte Carlo transport
  - **Sources**: Fixed-source and *k*-eigenvalue calculations
  - **Geometry**: Constructive solid geometry, CAD-based geometry via DAGMC
  - **Particles**: Neutrons, photons, (thick-target bremsstrahlung)
  - **Data**: Continuous energy or multigroup, multipole for Doppler broadening
  - **Parallelism**: MPI + OpenMP, experimental GPU support
  - **Tallies**: Flexible user-defined tally system, post-processing
  - **Variance reduction**: weight windows, source biasing, survival biasing
- Python and C/C++ API
- Depletion/activation solver
- Random ray solver
- Geometry visualization

What makes OpenMC unique?

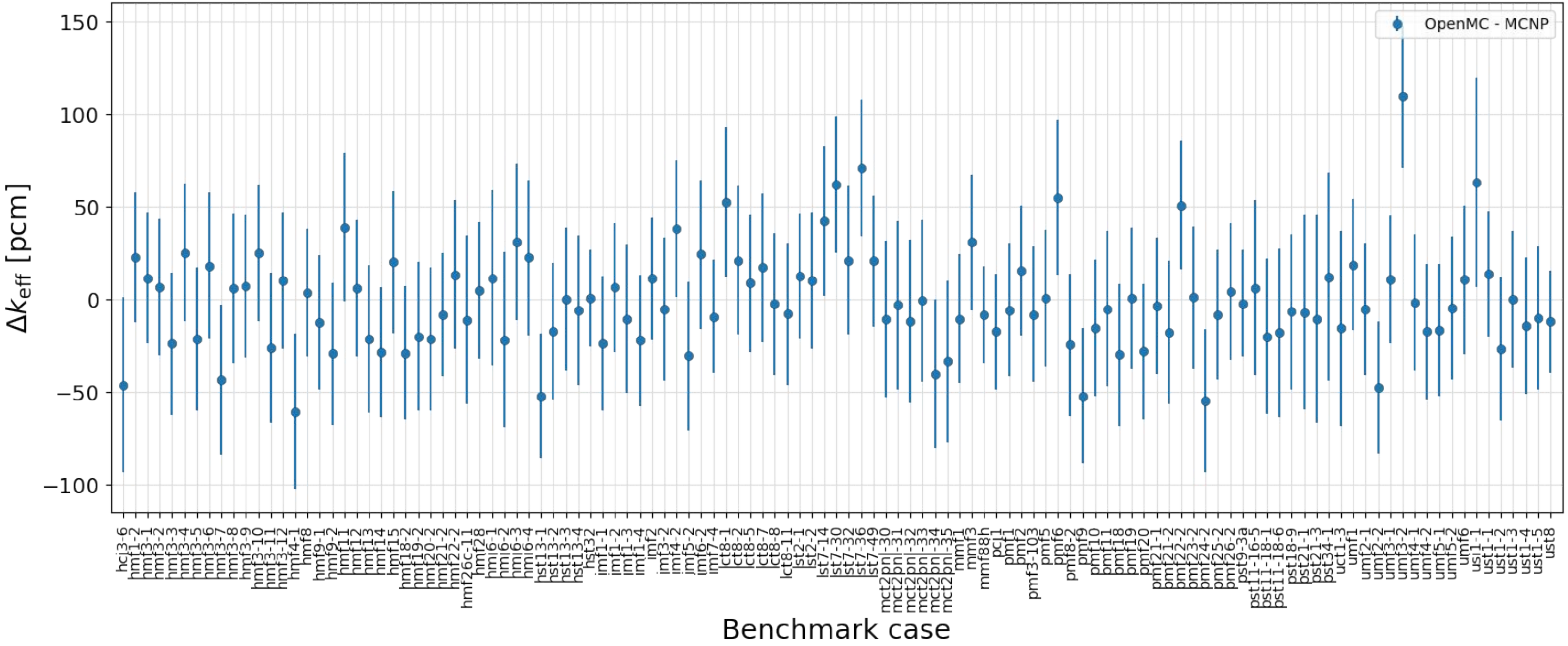
- ➢ Programming interfaces (Python and C/C++)
- ➢ Random ray solver for variance reduction
- ➢ Workflow automation (multigroup cross sections, shutdown dose rate, etc.)
- ➢ Nuclear data interfaces and representation
- ➢ Tally abstractions
- ➢ Parallel performance
- ➢ Development workflow and governance

Parallel Performance (full-core SMR)



V&V: *k*-eigenvalue

- ➢ OpenMC vs MCNP on criticality benchmarks



- ➢ OpenMC vs Serpent for depletion

Software Architecture

- Mixed **C++14** and **Python 3** codebase
  - C++: particle transport, plotting, volume calculations
  - Python: model building, pre/post-processing, depletion
- **CMake** build system for portability
- Version control through **git**
- Code hosting, bug tracking through **GitHub**
- Test suite runs on **GitHub Actions** CI platform

Resources

- Code: https://github.com/openmc-dev/openmc
- Docs: https://docs.openmc.org
- Nuclear Data: https://openmc.org
- Forum: https://openmc.discourse.group

Questions?