ORIGINAL ARTICLE

INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE

ISSN: 2692-515-x, Impact Factor: 9.23

American Academic publishers, volume 6, issue 01,2026

Journal: https://www.academicpublishers.org/journals/index.php/ijai

OPENMC: ARCHITECTURAL EVOLUTION AND NEW HORIZONS IN STOCHASTIC RADIATION TRANSPORT MODELING

Azizov Sh.M.

expert of the State Ecological Expertise, Tashkent

E-mail: cproangel@gmail.com

Abstract

This paper provides a comprehensive analysis of the OpenMC software suite – a next-generation tool for solving the neutral particle transport equation using the Monte Carlo method with continuous-energy data. The transition from traditional imperative input formats to object-oriented experiment programming via the Python API is examined. Particular attention is paid to the mathematical formalism of the solved equations (Boltzmann transport equation, eigenvalue problem, depletion kinetics), the implementation of hybrid parallelism (MPI/OpenMP), and the efficiency of handling nuclear data in HDF5 format. The role of the code's open architecture in addressing scientific verification challenges, CAD system integration, and multiphysics modeling of nuclear power plants is discussed.

Keywords

Monte Carlo method, nuclear reactor physics, OpenMC, neutron transport, HDF5, parallel computing, Python API, fuel depletion, CRAM, DAGMC.

Introduction

For decades, the "gold standard" in neutron physics calculations has been codes developed at US national laboratories in the mid-20th century, such as MCNP or the SCALE family. Despite high physical fidelity, these tools suffer from architectural limitations inherent to their era: rigid reliance on legacy Fortran standards, cumbersome text-based I/O formats, and restrictive licensing that hampers algorithm verification by the broader scientific community. OpenMC was developed by the Computational Reactor Physics Group at MIT as a response to the need for modernizing the computational base and adopting High-Performance Computing (HPC) practices.

The fundamental difference of OpenMC lies in its modular architecture. The computational kernel, responsible for particle transport, is written in C++ to ensure maximum performance and efficient memory management. Meanwhile, the user interface, preprocessing, and result post-processing are offloaded to a high-level Python environment. This approach allows researchers to integrate radiation transport calculations directly into complex data analysis pipelines and utilize machine learning and optimization libraries without creating intermediate parsers, effectively transforming modeling from a "black box" execution into a programmable experiment.

Mathematical Formalism and Numerical Methods

ORIGINAL ARTICLE

INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE

ISSN: 2692-515-x, Impact Factor: 9.23

American Academic publishers, volume 6, issue 01,2026

Journal: https://www.academicpublishers.org/journals/index.php/ijai

The physical model of OpenMC is based on solving the linear stationary integro-differential Boltzmann transport equation. Unlike deterministic methods that discretize phase space, the Monte Carlo method constructs the solution to the integral form of this equation through stochastic simulation of particle trajectories.

The Transport Equation and the Eigenvalue Problem

For a phase space defined by position **r**, direction **Ω**, and energy E, the balance equation for neutral particles is written as follows:

$$\Omega \nabla \psi(\mathbf{r},\Omega,E)+\Sigma_t(\mathbf{r},E)\psi(\mathbf{r},\Omega,E)=\int_0 dE' \int_{4\pi} d\Omega'\Sigma_s(\mathbf{r},E'\to E,\Omega'\to\Omega)\psi(\mathbf{r},\Omega',E')+S(\mathbf{r},\Omega,E), \quad (1)$$

where ψ is the angular flux density, Σ_t is the macroscopic total cross-section, and the right-hand side describes the neutron source due to scattering and external factors. For modeling critical systems (reactors), OpenMC solves the homogeneous form of the eigenvalue equation:

$$\mathbf{L}\psi+\mathbf{T}\psi=\frac{1}{k_{eff}}\mathbf{F}\psi. \quad (2)$$

Here, **L** and **T** are the transport and collision operators, respectively, and **F** is the fission operator. The numerical solution is obtained using the power iteration method, where fission neutrons from the current generation serve as the source for the next. To assess the convergence of the fission source distribution to the fundamental mode, OpenMC implements Shannon entropy calculation, minimizing result bias due to insufficient convergence in the initial simulation stages.

Depletion Kinetics and the CRAM Method

A key strength of OpenMC is its built-in depletion module, which couples transport calculations with the solution of the Bateman differential equation system describing isotopic composition changes over time:

$$\frac{dN_i(\mathbf{r},t)}{dt}=\sum_j(\lambda_{j\to i}+\sigma_{j\to i}\phi)N_j-(\lambda_i+\sigma_{a,i}\phi)N_i. \quad (3)$$

In matrix representation, the system takes the form: $\frac{d\mathbf{n}}{dt}=\mathbf{A}\mathbf{n}$. The depletion matrix **A** is characterized by high stiffness and immense dimensionality (thousands of isotopes). To solve this system, OpenMC employs the Chebyshev Rational Approximation Method (CRAM). The solution, formally written via the matrix exponential: $\mathbf{n}(t)=e^{\mathbf{A}t}\mathbf{n}(0)$, is approximated by a rational function on the negative real axis. This allows for calculating isotopic composition with near-machine precision and high computational stability, significantly outperforming traditional Taylor series expansion methods.

Computational Architecture and Data Handling

ISSN: 2692-515-x, Impact Factor: 9.23
American Academic publishers, volume 6, issue 01,2026
Journal: https://www.academicpublishers.org/journals/index.php/ijai

The efficiency of stochastic modeling directly depends on memory access speed and nuclear data handling. OpenMC supports the HDF5 (Hierarchical Data Format) for storing cross-section libraries, abandoning the archaic ACE format. HDF5 ensures parallel I/O and structured data access, which is critical for depletion calculations when data for hundreds of nuclides must be held in RAM. The code's algorithms enable continuous-energy calculations, accounting for resonant scattering and Doppler effects. notably, the "windowed multipole" formalism is implemented, allowing on-the-fly Doppler broadening of resonant cross-sections for any temperature, making the code promising for multiphysics calculations with thermal feedback.

To scale calculations, OpenMC uses a hybrid parallelism model. MPI is used for domain decomposition by particles between cluster nodes, while OpenMP is utilized within a node for multi-threaded shared-memory processing. This scheme minimizes communication overhead and demonstrates near-linear speedup. Active work is also underway to adapt particle tracking algorithms for Graphics Processing Units (GPUs) using the event-based tracking concept.

Geometric Modeling and Tally System

Geometric flexibility in OpenMC is provided by two complementary approaches. The first is classical Constructive Solid Geometry (CSG), based on Boolean operations on quadratic surfaces. This method is ideal for regular reactor core lattices and supports stochastic modeling of double-heterogeneity media (e.g., TRISO fuel). The second approach uses unstructured meshes and CAD models via the DAGMC library. This allows particle tracking directly on engineering 3D models, eliminating the labor-intensive process of geometry conversion, which is indispensable for designing complex fusion device shielding and research reactors.

The result collection system (tallies) in OpenMC is implemented through a flexible filter and score mechanism. Instead of rigid output forms, the researcher programmatically constructs the necessary detectors by applying filters (energy, angle, cell, material) to various physical scores (flux, reaction rates, heating). Various estimator types are supported, including track-length and collision estimators, allowing optimization of statistical error depending on the optical thickness of the medium.

Conclusion

OpenMC represents a qualitative leap in scientific software methodology for nuclear physics. Combining the rigor of Monte Carlo physical models with the flexibility of modern programming languages and the efficiency of parallel computing, it provides researchers with unprecedented control over the experiment. The open-source nature not only facilitates verification of scientific results but also stimulates community development, allowing for rapid tool adaptation to specific tasks–from Small Modular Reactor (SMR) design to Tokamak blanket analysis.

References

1. Romano P.K., Forget B. "The OpenMC Monte Carlo particle transport code". *Annals of Nuclear Energy*. 2013. 51: 274–281.

ISSN: 2692-515-x, Impact Factor: 9.23
American Academic publishers, volume 6, issue 01,2026
Journal: https://www.academicpublishers.org/journals/index.php/ijai

2. Romano P.K., Horelik N.E., Herman B.R., et al. "OpenMC: A state-of-the-art Monte Carlo code for research and development in nuclear engineering". *Annals of Nuclear Energy*. 2015. 82: 90–97.

3. Pusa M. "Rational approximations to the matrix exponential in burnup calculations". *Nuclear Science and Engineering*. 2011. 169(2): 155–167.

4. OpenMC Documentation https://docs.openmc.org.