The Journal of Open Source Software

Monte Carlo / Dynamic Code (MC/DC): An accelerated Python package for fully transient neutron transport and rapid methods development

Joanna Piper Morgan [1,2¶], **Ilham Variansyah** [1,2¶], **Samuel L. Pasmann** [1,3], **Kayla B. Clements** [1,2], **Braxton Cuneo** [1,5], **Alexander Mote** [1,2], **Charles Goodman**[1,4], **Caleb Shaw**[1,4], **Jordan Northrop** [1,2], **Rohan Pankaj** [1,6], **Ethan Lame** [1,2], **Benjamin Whewell** [1,3], **Ryan G. McClarren** [1,3], **Todd S. Palmer** [1,2], **Lizhong Chen** [1,2], **Dmitriy Y. Anistratov**[1,4], **C. T. Kelley**[1,4], **Camille J. Palmer** [1,2], and **Kyle E. Niemeyer** [1,2]

1 Center for Exascale Monte Carlo Neutron Transport **2** Oregon State University, Corvallis, OR, USA **3** University of Notre Dame, South Bend, IN, USA **4** North Carolina State University, Raleigh, NC, USA **5** Seattle University, Seattle, WA, USA **6** Brown University, Providence, RI, USA ¶ Corresponding author

DOI: 10.21105/joss.06415

Software

- Review
- Repository
- Archive

Editor: Kelly Rowland

Reviewers:

- @lewisfish
- @szabo137

Submitted: 28 January 2024
Published: 09 April 2024

Summary

Predicting how neutrons move through space and time, and change speed and direction of travel, are important considerations when modeling inertial confinement fusion systems, pulsed neutron sources, and nuclear criticality safety experiments, among other systems. This can be modeled with a Monte Carlo simulation, where particles with statistical importance are created and transported to produce a particle history (Lewis & Miller, 1984). A particle's path and the specific set of events that occur within its history are governed by pseudo-random numbers, known probabilities (e.g., from material data), and known geometries. Information about how particles move and/or interact with the system is tallied to construct a histogram solution of parameters of interest with an associated statistical error from the Monte Carlo process. Simulating dynamic systems that vary in time requires novel numerical methods to compute a solution performantly. We designed Monte Carlo / Dynamic Code (`MC/DC`) to explore such novel numerical methods on modern high-performance computing systems. We avoid the need for a compiled or domain-specific language by using the Numba compiler for Python to accelerate and abstract our compute kernels to near compiled code speeds. We have implemented novel algorithms using this scheme and, in some verification tests, have approached the performance of industry-standard codes at the scale of tens of thousands of processors.

Statement of need

`MC/DC` is a performant software platform for rapidly developing and applying novel, dynamic, neutron-transport algorithms on modern high-performance computing systems. It uses the Numba compiler for Python to compile compute kernels to a desired hardware target, including support for graphics processing units (GPUs) (Lam et al., 2015). `MC/DC` uses `mpi4py` for distributed-memory parallelism (Dalcin & Fang, 2021) and has run at the scale of tens of thousands of processors (Variansyah et al., 2023). These acceleration and abstraction techniques allow `MC/DC` developers to remain in a pure Python development environment without needing to support compiled or domain-specific languages. This has allowed `MC/DC` to grow from its initialization less than two years ago into a codebase that supports full performant neutron transport and investigation of novel transport algorithms, with development mostly from relative novices.

Many traditionally developed neutron-transport codes are export-controlled (e.g. `MCNP` (Rising et al., 2023), `Shift` (Hamilton & Evans, 2019), and `MCATK` (Adams et al., 2015)) and some are known to be difficult to install, use, and develop in. `MC/DC` is open-source, and thus, similar to other open-source Monte Carlo neutron-transport codes (e.g., `OpenMC` (Romano et al., 2015)), it promotes knowledge sharing, collaboration, and inclusive, community-driven development. What makes `MC/DC` unique is that its code base is exclusively written in Python, making it a good method exploration tool and an excellent entry point for students. Furthermore, `MC/DC` is wrapped as a Python package that can be conveniently installed via the `pip` distribution, and its development is assisted by a suite of unit, regression, verification, and performance tests, which are mostly run using continuous integration via GitHub Actions. This all together makes `MC/DC` ideal for use in an academic environment for both research and education.

`MC/DC` has support for continuous and multi-group energy neutron transport physics with constructive solid geometry modeling. It can solve k-eigenvalue problems (used to determine neutron population growth rates in reactors) as well as fully dynamic simulations. It also supports some simple domain decomposition, with more complex algorithms currently being implemented. In an initial code-to-code performance comparison, `MC/DC` was found to run about 2.5 times slower than the Shift Monte Carlo code for a simple problem and showed similar scaling on some systems (Variansyah et al., 2023).

`MC/DC`-enabled explorations into dynamic neutron transport algorithms have been successful, including quasi-Monte Carlo techniques (Pasmann et al., 2023a), hybrid iterative techniques for k-eigenvalue simulations (Pasmann et al., 2023b), population control techniques (Variansyah & McClarren, 2022a; Variansyah & McClarren, 2022b), continuous geometry movement techniques that model transient elements (Variansyah & McClarren, 2023b) (e.g., control rods or pulsed neutron experiments) more accurately than step functions typically used by other codes, initial condition sampling technique for typical reactor transients (Variansyah & McClarren, 2023a), hash-based random number generation (B. S. Cuneo & Variansyah, 2024), uncertainty and global sensitivity analysis (K. Clements et al., 2023; K. B. Clements et al., 2024), residual Monte Carlo methods, and machine learning techniques for dynamic node scheduling, among others.

Future Work

The main `MC/DC` branch currently only supports CPU architectures enabled by Numba (`x86-64`, `arm64`, and `ppc64`) but we are rapidly extending support to GPUs. We currently have operability on Nvidia GPUs (supported via Numba), and work is ongoing to enable compilation for AMD GPUs. On GPUs, `MC/DC` will use the `harmonize` asynchronous GPU scheduler to increase performance (B. Cuneo & Bailey, 2023). `harmonize` works by batching jobs during execution such that similar operations get executed simultaneously, reducing the divergence between parallel threads running on the GPU.

We will continue to explore novel methods for dynamic neutron transport and will keep pushing to make `MC/DC` not only a proven platform for rapidly exploring neutron-transport methods, but also a fully-fledged simulation code for academic and industrial use.

Acknowledgements

This work was supported by the Center for Exascale Monte-Carlo Neutron Transport (CEMeNT) a PSAAP-III project funded by the Department of Energy, grant number: DE-NA003967.

References

Adams, T., Nolen, S., Sweezy, J., Zukaitis, A., Campbell, J., Goorley, T., Greene, S., & Aulwes, R. (2015). Monte Carlo Application ToolKit (MCATK). *Annals of Nuclear Energy*, *82*, 41–47. https://doi.org/10.1016/j.anucene.2014.08.047

Clements, K. B., Geraci, G., Olson, A. J., & Palmer, T. S. (2024). A variance deconvolution estimator for efficient uncertainty quantification in Monte Carlo radiation transport applications. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *319*, 108958. https://doi.org/10.1016/j.jqsrt.2024.108958

Clements, K., Geraci, G., Olson, A. J., & Palmer, T. (2023, August). Global sensitivity analysis in Monte Carlo radiation transport. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. https://doi.org/10.48550/arXiv.2403.06106

Cuneo, B. S., & Variansyah, I. (2024). An alternative to stride-based RNG for Monte Carlo transport. In *Submitted to Transactions of the American Nuclear, Annual meeting 2024*. https://doi.org/10.48550/arXiv.2403.06362

Cuneo, B., & Bailey, M. (2023). Divergence reduction in Monte Carlo neutron transport with on-GPU asynchronous scheduling. *ACM Trans. Model. Comput. Simul.* https://doi.org/10.1145/3626957

Dalcin, L., & Fang, Y.-L. L. (2021). mpi4py: Status Update After 12 Years of Development. *Computing in Science & Engineering*, *23*(4), 47–54. https://doi.org/10.1109/MCSE.2021.3083216

Hamilton, S. P., & Evans, T. M. (2019). Continuous-energy Monte Carlo neutron transport on GPUs in the Shift code. *Annals of Nuclear Energy*, *128*, 236–247. https://doi.org/10.1016/j.anucene.2019.01.012

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based Python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 1–6. https://doi.org/10.1145/2833157.2833162

Lewis, E. E., & Miller, W. F. (1984). *Computational methods of neutron transport*. John Wiley; Sons, Inc. https://www.osti.gov/biblio/5538794

Pasmann, S., Variansyah, I., Kelley, C. T., & McClarren, R. (2023a). A quasi-Monte Carlo method with Krylov linear solvers for multigroup neutron transport simulations. *Nuclear Science and Engineering*. https://doi.org/10.1080/00295639.2022.2143704

Pasmann, S., Variansyah, I., Kelley, C. T., & McClarren, R. (2023b). iQMC: Iterative quasi-Monte Carlo with Krylov linear solvers for k-eigenvalue neutron transport simulations. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. https://doi.org/10.48550/arXiv.2306.11600

Rising, M. E., Armstrong, J. C., Bolding, S. R., Brown, F. B., Bull, J. S., Burke, T. P., Clark, A. R., Dixon, D. A., Forster, R. A., III, Giron, J. F., Grieve, T. S., Hughes, H. G., III, Josey, C. J., Kulesza, J. A., Martz, R. L., McCartney, A. P., McKinney, G. W., Mosher, S. W., Pearson, E. J., … Zukaitis, A. J. (2023). *MCNP$^{®}$ Code Version 6.3.0 Release Notes* (LA-UR-22-33103, Rev. 1). Los Alamos National Laboratory. https://doi.org/10.2172/1909545

Romano, P. K., Horelik, N. E., Herman, B. R., Nelson, A. G., Forget, B., & Smith, K. (2015). OpenMC: A state-of-the-art Monte Carlo code for research and development. *Annals of Nuclear Energy*, *82*, 90–97. https://doi.org/10.1016/j.anucene.2014.07.048

Variansyah, I., & McClarren, R. G. (2022a). Analysis of population control techniques for time-dependent and eigenvalue Monte Carlo neutron transport calculations. *Nuclear Science and Engineering*, *196:11*, 1280–1305. https://doi.org/10.1080/00295639.2022.2091906

Variansyah, I., & McClarren, R. G. (2022b). Performance of population control techniques in Monte Carlo reactor criticality simulation. *International Conference on Physics of Reactors*. https://doi.org/10.13182/physor22-37871

Variansyah, I., & McClarren, R. G. (2023a). An effective initial particle sampling technique for Monte Carlo reactor transient simulations. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. https://doi.org/10.48550/arXiv.2305.07646

Variansyah, I., & McClarren, R. G. (2023b). High-fidelity treatment for object movement in time-dependent Monte Carlo transport simulations. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. https://doi.org/10.48550/arXiv.2305.07641

Variansyah, I., Morgan, J. P., Northrop, J., Niemeyer, K. E., & McClarren, R. G. (2023). Development of MC/DC: A performant, scalable, and portable Python-based Monte Carlo neutron transport code. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. https://doi.org/10.48550/arXiv.2305.07636