

The MCNP[®]6 code: A decade of progress

Michael E. Rising^{*ID}, Jerawan C. Armstrong, Simon R. Bolding, Jeffrey S. Bull, Laura Casswell, Alexander R. Clark^{ID}, R. Arthur Forster, Cole S. Frederick, Jesse F. Giron^{ID}, Fred B. Jones, Colin J. Josey, Timothy M. Kelley, Joel A. Kulesza^{ID}, Michael A. Lively^{ID}, Robert C. Little, Sriram Swaminarayan, Jeremy E. Sweezy^{ID}, Pablo A. Vaquer, Colin A. Weaver^{ID} and Anthony J. Zukaitis

Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, USA

Received: 23 October 2024 / Received in final form: 23 December 2024 / Accepted: 30 January 2025

Abstract. After several years of effort involved in merging the Los Alamos National Laboratory MCNP5 and MCNPX codes, in 2013 the first production release of version 6 of the Monte Carlo N-Particle[®], or MCNP[®], code MCNP6.1 was distributed publicly. Since then, three significant releases have been issued: MCNP6.1.1beta in 2014, MCNP6.2 in 2018, and MCNP6.3 in 2023. While each release always contains new features, code enhancements, and bug fixes, each version has had a different primary focus, ranging from improved calculational efficiency to new powerful utilities and tools, to software modernization of the code base. With all that has been learned over the first decade of the MCNP6 code, continuous progress is being made toward a modernized, general-purpose Monte Carlo radiation transport code that remains a trusted resource for the global community of practitioners. This paper describes these first 10+ years of the MCNP6 code and its continually improving data libraries, and gives some insight into how the next decade is expected to unfold.

1 Introduction

In 2013, the first production version of the¹MCNP[®] code, MCNP6.1 [1,2] was released to the public after the completion of the MCNP5 [3] and MCNPX [4] merger. A year later, a follow-on “beta” version of the MCNP6.1 code, named MCNP6.1.1beta [5], was released with additional features and improvements to the initial MCNP6.1 production version. The next significant releases, MCNP6.2 [6] and MCNP6.3 [7], were completed in 2018 and 2023, respectively. These four releases contained numerous new features, code enhancements, and bug fixes, as well as revisions to the user and theory manuals and extensive verification and validation. The MCNP code is well-documented, and numerous references are available on the MCNP website at <https://mcnp.lanl.gov>. The remainder of this paper is organized as follows: Section 2 describes the individual MCNP6 code versions and highlights the major accomplishments within each, Section 3 discusses several notable innovations across all

code versions, Section 4 shows a variety of code comparisons related to verification, validation, and performance, and Section 5 provides some insights into the ongoing and future developments of the MCNP6 code.

2 Major MCNP6 code releases

The complete history of Monte Carlo code development at Los Alamos National Laboratory (LANL), starting from the Manhattan Project time frame to present day, spans many decades and has been well documented [8,9]. In the early years of the MCNP code, dating back to 1980, prior to the first public release of the MCNP code, Thompson and Cashwell predicted the future development of the code was correlated to its overall purpose by noting:

“MCNP is not a static code. It is under constant scrutiny and development by X-6 [the current Monte Carlo Codes group, XCP-3]. [...] If MCNP ever becomes static, it will be so because there is no further use for it. We do not anticipate this happening; rather, the opposite seems to be the case.” [10]

Following in the footsteps of our predecessors, and due to the ever-evolving needs of the global nuclear science community, the MCNP code has been continuously developed since its inception, with many major releases over the past five decades. In this paper, our focus is on the latest series

* e-mail: mrising@lanl.gov

¹ MCNP[®] and Monte Carlo N-Particle[®] are registered trademarks owned by Triad National Security, LLC, manager and operator of Los Alamos National Laboratory. Any third party use of such registered marks should be properly attributed to Triad National Security, LLC, including the use of the designation as appropriate.

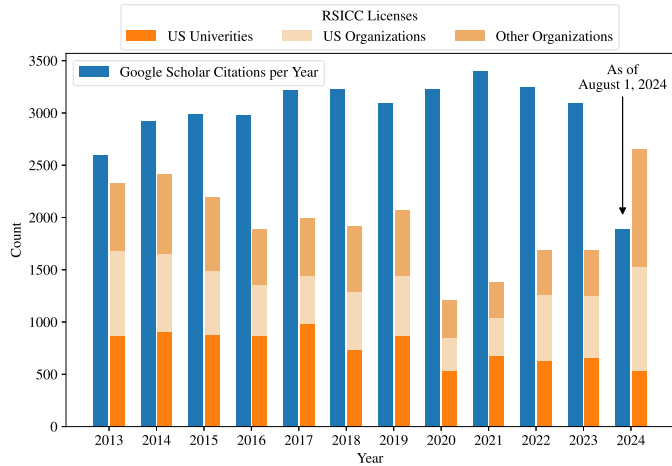


Fig. 1. The yearly number of Google Scholar citations (blue) and RSICC licenses issued (shades of orange) for the MCNP6 code since 2013.

of MCNP6 code versions and corresponding nuclear and atomic data that have been released during the last decade of MCNP code development at LANL.

Figure 1 shows the annual trends of both Google Scholar citations and licenses issued by the Radiation Safety Information Computational Center (RSICC) at Oak Ridge National Laboratory (ORNL). RSICC, along with the Organization for Economic Cooperation and Development Nuclear Energy Agency (OECD/NEA) and the Research Organization for Information Science and Technology (RIST) for various periods of time, have been responsible for licensing and distributing the MCNP software to end users for the past 40+ years with the MCNP3 code first distributed in 1983. Since late 2011, over 25 000 licenses of the MCNP code have been issued by RSICC showing a constant need and use of the code by the community.

2.1 MCNP6.1

In general, the MCNP6.1 code represents a combination of all of the capabilities in the MCNP5 and MCNPX codes. While the merger process took around 6 years and many dozen human-years to complete, several new features and code enhancements were also introduced into the first production release of the MCNP6 code that did not exist in any of its predecessors. The release notes of MCNP6.1 [1] provide sufficient details on all of the new features in this first release of MCNP6. A few of these features are highlighted here in part because of their overall impact and continued development over the last decade.

Based on the Abaqus Computer Aided Engineering (CAE) [11] mesh-file format, the MCNP6 code is able to embed an unstructured mesh (UM) geometry into a universe within the standard constructive solid geometry (CSG) hierarchical geometry capability. Initially this capability was intended for and tested with neutron and photon applications within a hybrid CSG and Abaqus-formatted UM [12] with aspirations to extend the UM

tracking capabilities to all particles. This is a powerful technique that ultimately enables a new user workflow using external computer aided design (CAD) or CAE and mesh generation tools to define geometry for use in MCNP6 [13]. Additionally, the ability to directly compute fluxes, reactions rates, and energy deposition on the UM elements allows for improved multi-physics coupling, described in some detail in Section 3.1.

Figure 2 shows a radiograph image computed by MCNP6.3 of a high-fidelity UM model of the ICRP-145 adult phantom [14], highlighting one of many applications where the UM modeling and radiation transport capability can be enormously powerful.

Research into adjoint-based k -eigenvalue tally capabilities was already underway within the MCNP5 code. For example, the ability to compute adjoint-weighted point kinetics parameters, such as neutron generation time (A), effective delayed neutron fraction (β_{eff}), and the Rossi- α , were already established [15] using the Iterated Fission Probably (IFP) method. New in MCNP6.1 was the ability to compute the adjoint-weighted k -eigenvalue sensitivity coefficients to nuclear data [16], such as cross sections, neutron multiplicity, and fission neutron energy spectra. Additionally, a new adjoint-weighted reactivity perturbation capability was implemented, where all of these adjoint-weighted k -eigenvalue capabilities made use of the IFP method. The use of the k -eigenvalue nuclear data sensitivity capability would continue to grow throughout the decade, ultimately becoming an integral part of sensitivity/uncertainty-based criticality safety tools, such as Whisper, described further in Section 2.3.

At the time, the latest Cascade-Exciton Model (CEM), CEM03.03, and Los Alamos Quark-Gluon String Model (LAQGSM), LAQGSM03.03, high-energy physics event generator models were integrated and tested in MCNP6.1 [17]. Like the transition between the tabulated low-energy cross section data and the intermediate-energy model physics, there is a default transition point between the CEM and LAQGSM models that depends on the projectile particle being transported and the target material. In general, the LAQGSM model is best suited for the highest energy projectiles, up to 1 TeV/nucleon, where CEM is recommended to be used up to 1–5 GeV depending on the projectile-target system. The inclusion of the latest CEM and LAQGSM models into the first release of MCNP6 is noteworthy because much of the development and changes within these standalone event generators and the integrated versions within MCNP6.1 are very similar to the models in present day MCNP6.3.

The nuclear and atomic data improvements made available with MCNP6.1 included new data for electron, photon, and neutron physics. The first introduction of more detailed electron-photon-relaxation (EPR) data, `eprdata12`, and the corresponding code enhancements not only made it possible to track both electrons and photons to a lower energy than they had previously been able, but also resulted in the addition of a new single-event electron transport algorithm that could be used where the condensed-history electron transport algorithm tends to fail [18,19]. For other applications involving low-energy neutron physics, the most recent release

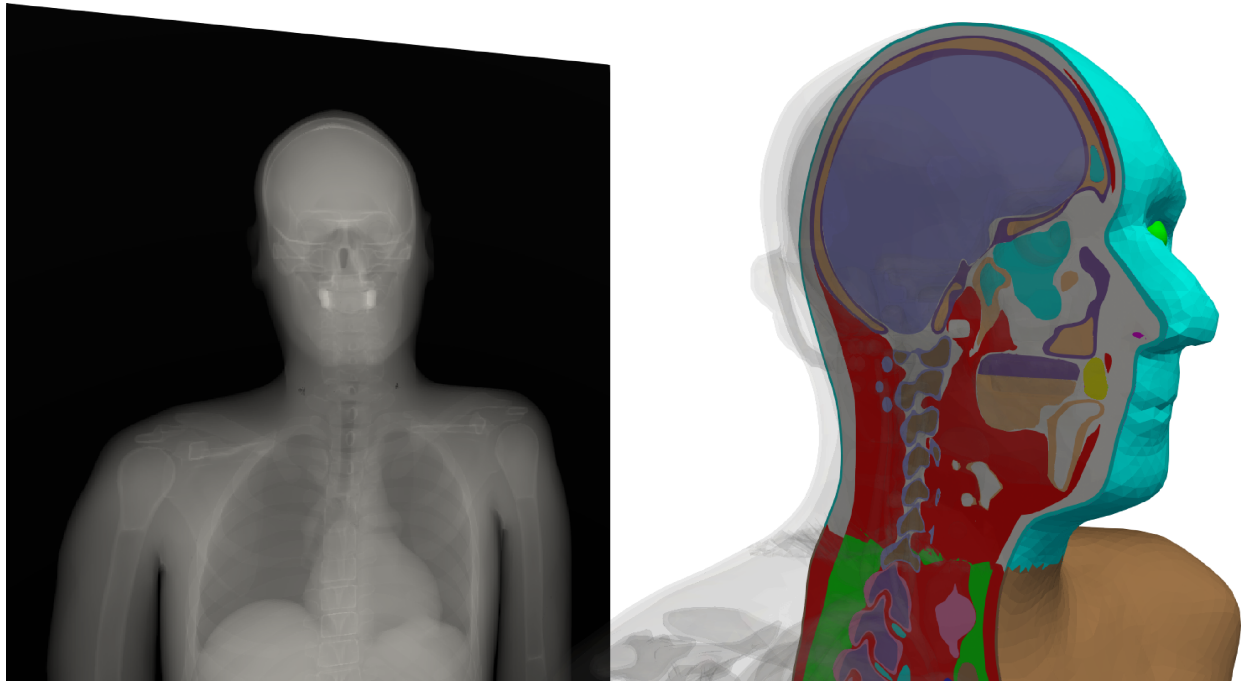


Fig. 2. A MCNP6.3 simulated radiograph (left) of the ICRP-145 adult phantom [14] UM model (right).

of the U.S. Evaluated Nuclear Data Library, ENDF/B-VII.1 [20], was processed and released with the MCNP6.1 code. This includes neutron reaction information for 423 nuclides [21] and 21 thermal neutron scattering materials [22] which contained a new continuous representation of the secondary energy distributions.

2.2 MCNP6.1.1beta

Even with a fresh release of MCNP6.1 one year before, a backlog of work that was done or in progress had already accumulated in part because of the lengthy merger process. With many code enhancements already lined up for another release, the decision to create the MCNP6.1.1beta was made [23]. In comparison, and not just because of its name, MCNP6.1.1beta differed from all other releases of the MCNP6 code to date. Although there were official “beta” releases of the MCNP6 code prior to MCNP6.1, the use of “beta” from standard software development terminology was accurate in that context, referring to a version provided to a group of external advanced users (outside the development team) to test, provide feedback, and suggest potential fixes for the software functionality. The use of the term “beta” in the context of the MCNP6.1.1beta release was solely intended to convey the message that some of the newer features in the code had not undergone rigorous developer team testing like most of the features developed in previous versions. This confusion may have contributed to the slow adoption of this newer version of the code in various user communities, such as nuclear criticality safety.

A version of the CINDER’90 depletion code [24] was originally integrated in the MCNPX version 2.6.0

code [25] to be used for nuclear reactor core material depletion simulations. A natural extension toward activation and delayed particle emission physics was immediately made with additional features added in nearly every version of the code since that time. For example, after the delayed beta emission from both particle-induced reactions and unstable radionuclides was completed in MCNP6.1, delayed alpha particle emissions were added in MCNP6.1.1beta [26]. With each subsequent MCNP6 release, however, the development and maintenance of the capabilities and data offered by the CINDER’90 code have slowly decreased. In Section 5, the plans for a new, modern set of depletion and activation capabilities are briefly mentioned.

Among other side effects from the merger process, some portions of the code were found to be computationally slower and less efficient than their predecessors. This was true for much of the neutron physics, in the low-energy tabulated data regime, where it was observed that the MCNP6.1 version of the code ran 20–30% slower than MCNP5 [27]. In an effort to gain back the computational efficiencies, the code was heavily hand-optimized such that MCNP6.1.1beta ran faster than both MCNP6.1 and MCNP5, while still performing correctly for criticality calculations.

2.3 MCNP6.2

After the releases of MCNP6.1 and MCNP6.1.1beta were completed within a year, the next significant release, MCNP6.2, did not come until 2018 [28]. While a few new features were introduced into the code, there were many incremental extensions to existing code features and

modest improvements throughout. For example, the activation and delayed particle production features of the code included a new delayed line emission capability [29] as well as a new positron emission treatment [30].

Some of the most substantial additions in the MCNP6.2 code product came through external efforts, outside of the MCNP6 code base. These efforts in standalone physics library developments and tools to support both specific and general-purpose end-user applications foreshadowed some of the future development practices carried into MCNP6.3 and beyond.

Models of standalone physics libraries, such as the LANL Cascading Gamma-ray Multiplicity and Fission (CGMF) code [31] and the Lawrence Livermore National Laboratory (LLNL)/Lawrence Berkeley National Laboratory (LBNL) Fission Reaction Event Yield Algorithm (FREYA) code [32] were in active development and preparing to be included in the MCNP6 code. These models were integrated into MCNP6.2 to support nuclear non-proliferation applications interested in modeling unique signatures of the spontaneous and neutron-induced fission process through these high-fidelity low-energy fission event-generator models [33]. Section 3.2 describes the non-proliferation applications of these new fission models.

The Whisper code [34] version 1.1 [35] is a standalone sensitivity/uncertainty-based application used to compute upper subcritical limits for nuclear criticality safety applications. Using a combination of new MCNP6 capabilities, nuclear data covariances [36], and criticality safety benchmark experiments [37], Whisper provides quantitative values for overall calculational bias and margins for individual criticality safety applications of interest. Development of this utility would not have been possible without the advent of the adjoint-based k -eigenvalue nuclear data sensitivity capability in MCNP6.1 (see Sect. 2.1). Beyond Whisper's primary use as a tool for nuclear criticality safety practitioners, it has also been used as a database of information useful to train machine learning algorithms to predict MCNP bias and to find potential causes of this bias within the nuclear data [38].

The multipurpose MCNPTools [39] library first released with MCNP6.2 provided an application programming interface (API) to access many of the MCNP6 outputs. These tools allow users to write their own code (e.g., scripts) to extract results from MCNP6 to efficiently manipulate the data for their own post processing and analysis needs. One example of the power of the MCNPTools API is the development of a standalone Detector Response Function Toolkit (DRiFT) [40] that interfaces with MCNPTools to extract information from an MCNP simulation for high-fidelity detector response simulations and analysis.

The Intrinsic Source Constructor (ISC) library [41] and MCNP Intrinsic Source Constructor (MISC) utility [42] (built from the ISC library), offer end-users the ability to prepare a source definition of an intrinsically radioactive material for use in their transport simulations. Compiled with natural abundance, radioactive decay and particle emission data, and a Bateman equation solver, the ISC library is a standalone tool that can be used to prepare sources for MCNP6 and other transport software, or it

can be used to compare to the built-in delayed particle emission capabilities of the MCNP6 code, for example.

Not many major nuclear or atomic data updates were made for the MCNP6.1.1beta code release. However, new EPR data [43], `eprdata14`, and validation testing [44] were included as part of the MCNP6.2 code release. Additionally, the new CP2011 library, consisting of incident light ion (i.e., protons, deuterons, tritons, helions, and alphas) charged-particle transport data tables, was released with the code [45] along with the ability for MCNP6.2 to use these tables for light ion transport for the first time.

2.4 MCNP6.3

After the MCNP6.2 code release, the development team made some significant changes to their software development practices in part to facilitate a dedicated code modernization effort. The sustainability and maintenance of the software was becoming more costly partly because of the complexity of the code base after the MCNP6 merger was complete. Recognizing the importance of the MCNP6 code at LANL and elsewhere, the institution began providing dedicated resources to maintain, improve, and modernize the code so that the next generation of developers and users would have access to this essential resource for the foreseeable future. Outside the source code, this institutional support has also resulted in more dedicated user-facing services, including having a user-support specialist on the development team, providing an overhauled website and user forum (<https://mcnp.lanl.gov/forum.html>), extending training opportunities with many MCNP classes available year-round (<https://mcnp.lanl.gov/classes.html>), and providing an annual MCNP symposium venue for users and developers to directly interact (<https://mcnp.lanl.gov/symposia.html>).

During the development of MCNP6.3 [46] the development team transitioned to using more standardized software development practices, including peer review of all code changes, automated regression and unit testing, and tracking of all issues with direct connections between the code changes and the issues they address. Installing these practices into the software development workflow was done with the intention to improve the overall quality of the software as well as to allow for multiple developers, including new team members, to be or become proficient across many parts of the code base. By also pivoting to other industry standard software development tools, such as relying on the CMake build system generator as a replacement to the legacy GNU make build system in MCNP6.2 and before, the current and future development of MCNP6 would be able to move more quickly toward its ultimate modernization goals. In Section 3.6, more discussion is provided on the impacts these new tools and software development practices have had on current development of MCNP6.

After multiple years of research and development, including the initial release of the fission matrix capability within MCNP6.2, the MCNP6.3 code included a more complete implementation of the fission matrix

capability, with robust statistical testing enhancements, and options to allow the code to determine the correct number of discard cycles and to accelerate the convergence of the fission source during inactive cycles [47]. These new features discussed further in Section 3.3 have the potential to revolutionize the workflow for all criticality calculation practitioners, including those focused on nuclear criticality safety and reactor physics applications.

With an ever-increasing demand for more high-fidelity simulations, a new implementation was introduced in MCNP6.3 to efficiently compute and fit large tally structures, e.g. superimposed mesh tallies, onto modern computing hardware [48]. Through this standalone C++ library, multiple tallying algorithms and parallelization capabilities are implemented to support more general tallying needs within MCNP6 now and in the future. This new capability exemplifies the efforts made toward code modernization and improved software development practices by developing a modern software library that is reusable, extensible, interoperable, and fully unit tested. Figure 8 shows the breakdown of what languages the MCNP6 and its predecessors MCNPX and MCNP5 use where the code has been steadily transitioning the Fortran 77 code base to more modern Fortran and C/C++ source code.

Another major transition made during the MCNP6.3 code development cycle included the integration of some HDF5-formatted binary files [49], replacing Fortran binary formatted restart, unstructured mesh output, and particle track files. To support more advanced, 3D visualization, the XDMF format [50,51] was also used in conjunction with the upgrades to the superimposed mesh tally and the UM model and elemental edits HDF5 upgrades. In short, a couple of the most often high-dimensional datasets worth viewing in three dimensions are now able to be immediately visualized with open-source tools like ParaView [52,53] using the XDMF reader capabilities.

Recognizing that including all of the current and historical nuclear and atomic data within each release of the MCNP6 code was becoming a burden, all of the data processed for the MCNP code are now provided on the <https://nucleardata.lanl.gov> website. By releasing new data separately from the code, users of each code version can now update their installed data when new libraries are available for download. For example, shortly after the MCNP6.2 release the ENDF/B-VIII.0 library was released [54] and made available to download. Within ENDF/B-VIII.0, a new set of neutron free-gas reaction physics data tables for 557 nuclides were processed [55] along with 34 thermal neutron scattering materials [56]. Finally, the most recent release of data available prior to the MCNP6.3 release includes an updated library of light-ion reaction data, CP2020 [57], representing an update to the CP2011 data library.

Throughout the development of all MCNP code versions, the nuclear and atomic data development, processing, and testing have been tightly integrated [58–61] ensuring that the code performs correctly for many nuclear applications.

3 Major innovations

One aspect of the MCNP code development process that has always been present is the ability of the team to perform fundamental and application-specific scientific research to ultimately maintain the relevance of the code and its capabilities for modern applications. Many innovative research ideas have made their way into the production releases of the MCNP6 code, including:

- Unstructured mesh particle tracking (see Sect.3.1)
- Pre-collision next-event estimators [62]
- Cosmic-ray [63] and background sources [64]
- k -eigenvalue adjoint-based nuclear data sensitivity tallies [16]
- Correlated neutron (non-fission) reaction physics with the Cascading Gamma-ray Multiplicity (CGM) code [65]
- Correlated fission physics (see Sect. 3.2)
- Fission matrix-based convergence testing and acceleration (see Sect. 3.3)
- Multigroup cross section tally options for reactor physics applications [66]
- New tally algorithms for advanced parallelism (see Sect. 3.4)
- Statistical testing methods for tally verification [67]

Much of the always ongoing research into new methods, algorithms, and capabilities generally involves young staff, student researchers, and collaborations with university partners where the MCNP code has a significant impact across the academic community [68]. Some of these research efforts have turned into major developments in the past decade setting the stage for the future of the MCNP code. Only a few items are highlighted here which span multiple versions of the MCNP6 code releases.

3.1 Unstructured mesh

The Abaqus-based UM capability first introduced in the production MCNP6.1 code has seen some of the most dedicated code development over the past decade. These improvements have come in the form of bug fixes, code extensions for more cross-feature compatibility, and in new and/or modernized features. In early versions, such as MCNP6.1.1beta and MCNP6.2, the majority of the effort was focused on feature compatibility, extending the UM capabilities to be compatible with more physics options and tally capabilities already established within the code. In the latest MCNP6.3 release, the UM capability was subject to as much modernization effort as the rest of the code base, including improved accessibility through new file formats (see Sect. 3.5) and faster initialization and tracking algorithms [69].

In Figure 3, the UM performance of the most famous bare highly enriched uranium (HEU) critical experiment, Lady Godiva, is shown. Perhaps most importantly, a huge bottleneck in the overall UM calculation time was reduced from an $\mathcal{O}(N^2)$ to an $\mathcal{O}(N)$ initialization algorithm. This can be seen in the top panel of Figure 3, where the initialization time has been dramatically reduced in MCNP6.3

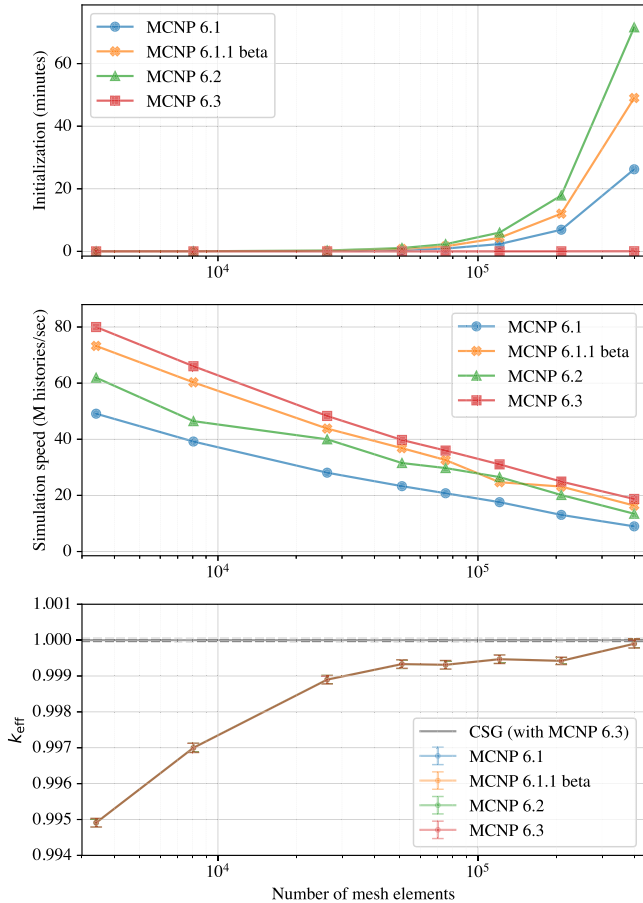


Fig. 3. UM computational and accuracy performance for the bare HEU Lady Godiva critical experiment. The initialization time (top panel), particle tracking rate (center panel), and k -effective predictions (bottom panel) are given as a function of the number of UM linear tetrahedral elements in the model.

for UM models with a large maximum number of elements in any part/instance. Additionally, from the initial MCNP6.1 release, some modest particle tracking speed improvements have remained in the latest releases seen in the center panel of Figure 3.

Beyond UM code enhancements, the notion of mesh quality [70], verification [71–75], and validation [76–79] have been a constant subject of investigation that will undoubtedly continue into the future. It is important to continue educating the community of UM users on how to prepare and check a high quality mesh that sufficiently captures the important features of a model for accurate radiation transport. For example, the bottom panel in Figure 3 demonstrates the crossroads a user faces where an accurate UM model may be more computationally expensive to simulate as seen in the panels above. At least $\sim 400\,000$ linear tetrahedral elements are needed to be within a standard deviation of the CSG solution for Lady Godiva (k -effective = 1.00000 ± 0.00005). The approach to the CSG solution comes down to how refined the UM model must be to preserve the true volume of the Lady Godiva sphere. For the Lady Godiva model this is not

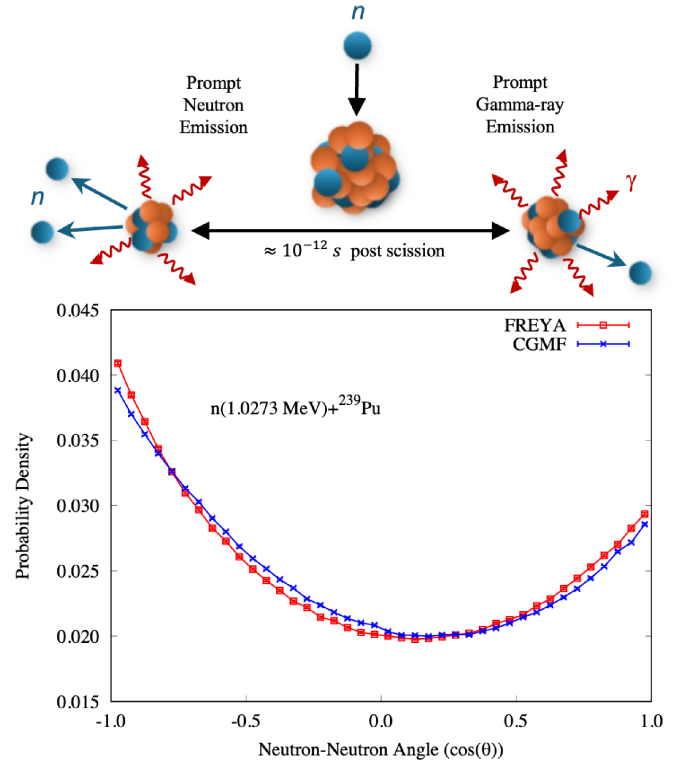


Fig. 4. On top is a pictorial representation of the neutron-induced fission process and the subsequent emission of prompt neutron and γ rays. The lower plot shows the CGMF and FREYA event-by-event correlation predictions between neutron angular emissions in a single fission event due to their emission from the accelerated light and heavy fission fragments.

necessarily a huge computational burden, but for other more complex UM models the number of mesh elements can become extremely large, requiring significant computational resources.

3.2 Correlated fission physics

Since the initial inclusion of the CGMF and FREYA fission event generators in MCNP6.2, the ability to simulate more complex signatures of the fission process has been possible.

Figure 4 shows one of the possible observable signatures of the fission process. In the fission process, the fragments accelerate away from each other where the neutrons emitted are boosted in the direction of the fission fragments, leading to the event-by-event neutron angular correlations in the laboratory frame shown in Figure 4.

When attempting to measure or simulate time-coincident neutron or γ ray emissions from fission in an array of independent detectors, event-by-event correlations are important to consider. By default in MCNP6, the neutrons and γ rays emitted from fission are all independently sampled in their energy and direction of emission, meaning they are all uncorrelated from each other. However, in reality these emitted particles are correlated

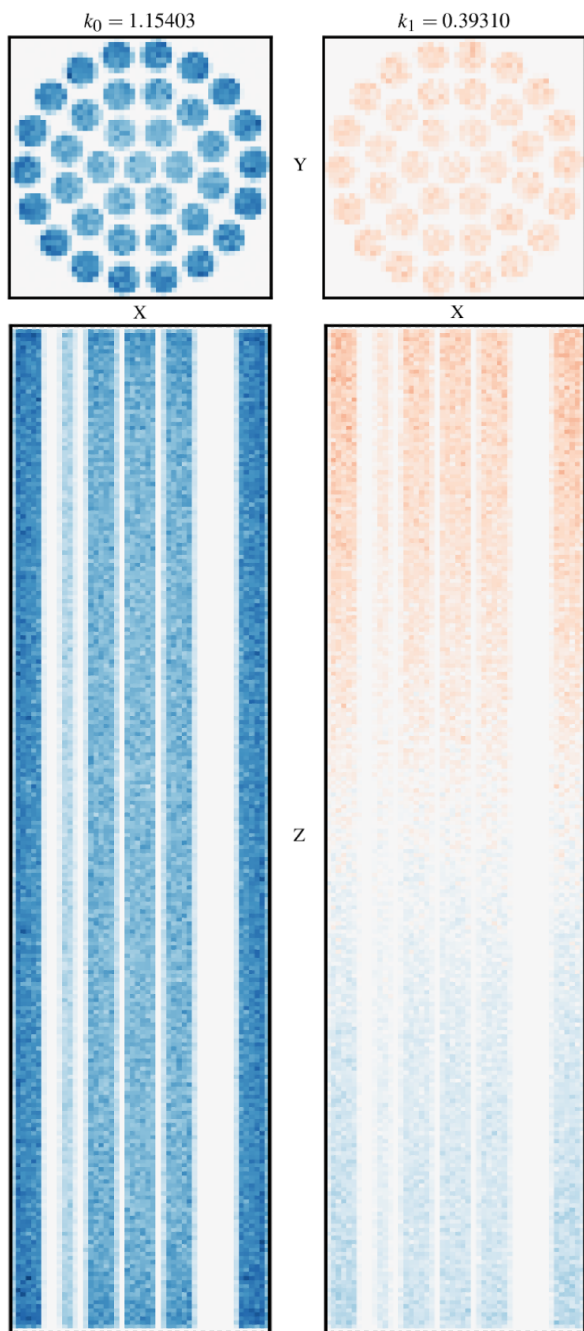


Fig. 5. The first two k -eigenmodes of a CANDU core, with the fundamental eigenmode on the left and the second eigenmode on the right. The blue and red represent positive and negative spatial flux solutions to the k -eigenvalue equation, respectively. Shown on top are the x - y slices 3/4 up the z -axis of the core and on bottom are the x - z slices through center of the y -axis.

to each other in their multiplicity, energy, and direction. Some of these particle emission correlations have been observed experimentally [80,81]. In the latest MCNP6.3 release, updates to CGMF have been incorporated and have been released in the open-source community [82].

3.3 Fission matrix

The new fission matrix options in MCNP6.3 have been introduced to support more robust k -eigenvalue criticality calculations. The three primary functions introduced in the latest version of the code (1) turn on the fission matrix calculation for use in improved source convergence testing, (2) automatically detect the proper number of inactive cycles to discard when convergence has been reached, and (3) accelerate the convergence of the power iteration solution using the fission matrix as a reference solution [47].

With the fission matrix capability turned on, the fission matrix is computed and used as a reference solution of the source distribution. By calculating both the power-iteration and fission-matrix fission source distribution metrics (e.g., Shannon entropy), the code has more information to test that the source has reached a converged distribution and that the number of histories simulated per cycle are sufficient to avoid undersampling and clustering issues [83]. Because of the enhanced statistical checking with the fission matrix solution, the code can automatically determine how many power iteration cycles must be discarded before active cycles begin. During inactive cycles the fission source distribution can use importance sampling based on the reference fission-matrix distribution to accelerate the convergence of the actual fission distribution. With all of these options working together, a fission-matrix-based simulation can lead to solutions where convergence is more robust and potentially faster depending on how challenging it is to manually optimize the number of user-selected inactive cycles. In a recent study [84], the latest fission matrix features in MCNP6.3 have been verified to work accurately and efficiently against a number of criticality safety k -eigenvalue problems.

In addition to the k -eigenvalue calculation improvements enabled through the fission matrix, the extraction of the fission matrix and subsequent analysis of the k -eigenmodes is now possible and simple to accomplish [85]. Figure 5 shows the first two k -eigenmodes in both the radial (top) and axial (bottom) dimensions of a CANDU reactor core simulation [73]. From this information the convergence of the system can be better understood with respect to how an initial guess may excite some of the higher order spatial modes, and how quickly these higher order modes will die away, using the dominance ratio (k_1/k_0) as a measure of the convergence rate.

3.4 High-performance mesh tally options

A consequence of having continuously growing high-performance computing (HPC) capabilities is the increased demands by the user community to perform more high-fidelity calculations. Unfortunately, many of the recent HPC systems deployed at LANL and elsewhere tend to pack in more central processing units (CPUs) with less memory available per CPU. For example, the latest HPC machine at LANL, Rocinante, includes 256 GB of memory shared among 112 CPU cores on each node. This poses a challenge when trying to fit large high-fidelity

simulations onto modern HPC systems. In MCNP6.3, a standalone tally library, `turbotally`, was developed and integrated to handle all of the backend tallying for the superimposed mesh tally (FMESH) capability.

Within `turbotally`, four separate algorithms are implemented. These include:

hist A standard history-based algorithm using numerically stable statistics [86]. This algorithm has no optimizations and is therefore computationally slower than the default algorithms used in the mesh tally in previous MCNP6 versions.

fast_hist Statistically equivalent to the history-based algorithm implemented in MCNP6.2 and earlier, with optimizations in place for increased computational speed. This is the default algorithm in MCNP6.3.

batch A new batch statistics algorithm is available which computes mean values over a fixed batch size and combines the batch mean values throughout the simulation. The batch statistics algorithm will require less memory than the history-based algorithms.

rma_batch Statistically equivalent to the batch algorithm enabled using the remote memory access (RMA) capability of the message passing interface (MPI). With this algorithm exceedingly large tallies can be modeled because the tally data structures are effectively decomposed across all memory available across all nodes.

Table 1 shows the expected peak memory used for each tally algorithm for a given tally size, T_{size} , number of MPI ranks, n_{ranks} , number of OpenMP threads, n_{threads} , and computational nodes, n_{nodes} .

Using the performance test configuration on the LANL Rocinante HPC machine discussed further in Section 4.2, where $n_{\text{ranks}} = 8$ per node, $n_{\text{threads}} = 13$, and $n_{\text{nodes}} = 10$, the maximum number of tally regions are shown in Table 2 as an example of the suitability of each tally algorithm with respect to memory consumption. The parenthetical values in Table 2 are the approximate cube root values of the total tally size indicating the number of tally bins along a single dimension within a three dimensional tally structure (i.e., x , y , and z spatial bins in a cartesian mesh tally). Of course, the speed of the simulation must also be considered. In general the speed of the algorithms, from fastest to slowest, go as follows: `batch`, `fast_hist`, and `hist`, with `rma_batch`, dependent on problem size and the MPI library in use. In summary, there are now options to find a suitable balance between mesh tally computational speed and memory consumption for every application.

3.5 File format upgrades and visualization improvements

By relying on standard, well-supported third-party libraries and tools, the ability to access, process, and analyze MCNP6 results has never been easier. In the development cycle for MCNP6.3, the HDF5 format [49] was selected to replace many of the Fortran binary files, produced by the code as part of the code modernization efforts. Because HDF5 binary formatted files are portable across operating systems with many freely available APIs

(e.g., C++, Fortran, Python, Perl), the replacement of the Fortran-formatted binary restart (`runtp`) and particle track (`ptrac`) files increases the accessibility of the data within. The UM mesh model and elemental edit file has also been cast into a HDF5 formatted file that can be used as input in place of the standard Abaqus file and/or can be used to store the elemental edit values.

To augment the HDF5 formatted data the XDMF format [50,51] provides a defined file specification that can be used to easily visualize high-dimensional data. For example, the superimposed mesh tallies and the UM model and elemental edits can be visualized in ParaView [52,53] like the image of the UM ICRP model shown on the right side of Figure 2.

While the superimposed mesh and UM models are naturally formatted into a 3-dimensional dataset, the `ptrac` data would generally need some kind of format conversion to be easily viewed.

Listing 1. Simple Python script used to convert the new HDF5 `ptrac` data into a comma separated value format readable by ParaView

```
#!/usr/bin/env python
import h5py
import numpy as np

def ptrac_points_to_csv(fname, gname):
    h5file = h5py.File(fname, "r")

    group = f'ptrack/{gname}'

    x = h5file[group]["x"]
    y = h5file[group]["y"]
    z = h5file[group]["z"]
    e = h5file[group]["energy"]
    d = np.vstack((x, y, z, e))

    np.savetxt(f'{gname}s.csv',
               d.T,
               delimiter=",",
               header="x,y,z,e")

ptrac_points_to_csv("ptrac.h5",
                    "Source")
ptrac_points_to_csv("ptrac.h5",
                    "SurfaceCrossing")
ptrac_points_to_csv("ptrac.h5",
                    "Collision")
ptrac_points_to_csv("ptrac.h5",
                    "Bank")
ptrac_points_to_csv("ptrac.h5",
                    "Termination")
```

The Python script shown in Listing 1 demonstrates a simple way to convert the HDF5 `ptrac` data into a column separated value (CSV) file readable in ParaView. Figure 6 shows the CSV data plotted in ParaView. In each of the panels, the black dots are the particle surface crossing events which can be used to visualize the geometry. The colored dots represent the energy of source, collision, secondary bank, and termination events.

The direction of MCNP6 development of visualization capabilities has generally steered toward more accessible

Table 1. Approximate peak **turbotally** memory usage

Algorithm	Node Memory Usage (Bytes)
<code>hist</code>	$16T_{\text{size}}n_{\text{ranks}} + 8T_{\text{size}}n_{\text{ranks}}n_{\text{threads}}$
<code>fast_hist</code>	$16T_{\text{size}}n_{\text{ranks}} + 8.8T_{\text{size}}n_{\text{ranks}}n_{\text{threads}}$
<code>batch</code>	$16T_{\text{size}} + 8T_{\text{size}}n_{\text{ranks}}$
<code>rma_batch</code>	$24T_{\text{size}}/n_{\text{nodes}}$

Table 2. Maximum theoretical tally size in performance testing discussed in Section 4.2 ($n_{\text{ranks}} = 8$ per node, $n_{\text{threads}} = 13$, and $n_{\text{nodes}} = 10$)

Algorithm	Maximum tally size
<code>hist</code>	2.67×10^7 ($\sim 299^3$)
<code>fast_hist</code>	2.45×10^7 ($\sim 291^3$)
<code>batch</code>	3.90×10^8 ($\sim 731^3$)
<code>rma_batch</code>	1.067×10^{11} ($\sim 4743^3$)

formats and conversion tools that allow for simple integration into common user workflows that make use of third-party visualization software.

However, recognizing that the internal plotter within MCNP6 still remains a valuable tool to help with geometry debugging, tally result visualization, and cross section plotting, a dedicated modernization effort has led to a modern, cross-platform Qt plotter converted from the X11-based plotter. Figure 7 shows the new Qt interface to the plotter released within the MCNP6.3 code distribution. This new Qt plotter will continue to be maintained, improved, and extended, for the foreseeable future.

3.6 Build system, code reviews, and continuous testing

One aspect of the MCNP6 code that has seen dramatic changes over the past decade includes the build and test system and the overall code integration and review practices. Around the time of the MCNP6.2 release, code development efforts began to take place outside of the main MCNP6 source code. The intention of this development was to build standalone, unit-tested components (e.g., **turbotally**) that could then be integrated into parts of MCNP6 to replace legacy software. In order to facilitate this development and integration, using a new, modern build and test system was prioritized. For MCNP6.3, the build and test system converted from the legacy GNU `make` system to the CMake build system generator. With this new system, building across platforms with multiple compilers is typically possible without modifications to the build system. More importantly, the integration of standalone or third-party libraries require minimal changes to the build system infrastructure as compared to the legacy build system. Beyond building the

code, the entirety of the regression testing suite is available through CTest, and the code can be packaged through the CPack capabilities. Overall, developing, testing, and packaging the code is much easier than it was 5–10 years ago.

As stated in the initial MCNP6.1 release document [1], there was a continuous build and test system in place for MCNP6 development. However, the code review process at the time was not as formalized as it is today. Now, every change made to the code is both automatically regression and unit tested as well as reviewed by another development team member before it is merged. The primary motivation with these changes to software development practices is to ultimately provide a better, more robust end product while bringing together multiple team members into the development process to support training and knowledge retention of the code base.

4 Evolution of MCNP6 code, testing, and performance

4.1 Code evolution

To maintain the predictive and trusted capabilities of the MCNP code, the code has been steadily evolving with a primary focus on code modernization. While the goals of code modernization do occasionally lead to more predictive capabilities or more computationally performant algorithms, these are not typical end goals of each modernization project. At the present time, maintaining the code performance (see Sect. 4.2) and the predictive physics through verification and validation (see Sect. 4.3) testing are sufficient when code modernization improvements are suggested.

In Figure 8, the lines of code of the entire source code package are broken down by a few primary languages. While there has been growth in the overall source code base from the time of the MCNP5 and MCNPX merger, the majority of this growth has been through the introduction of or transition to more modern languages. Note that MCNP6.3 is the only version that includes source code from standalone “dependencies”, represented by the right-most bar. In previous MCNP6 versions, prior to MCNP6.3, most code development was focused on direct integration into the monolithic source code repository, represented by the largest bar for each code version in Figure 8. Significant, ongoing modernization efforts in recent years after the MCNP6.2 release have led to the development of more general software libraries written and linked to the primary MCNP6.3 source code.

4.2 Computational performance

Because of the extensive set of features and capabilities in the MCNP code, highly efficient computational performance is generally difficult to achieve for all end-user applications. To maintain the performance of the code on massively parallel HPC platforms, scaling studies are routinely performed to judge the code performance [87].

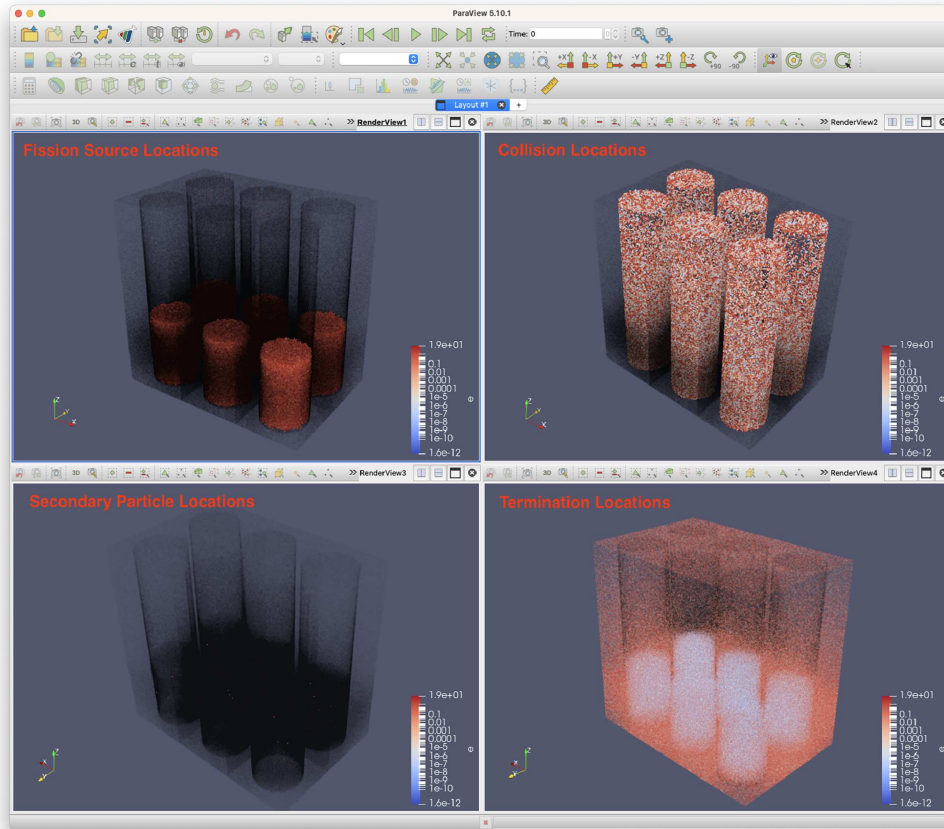


Fig. 6. ParaView visualization of the fission source, collision, secondary particle, and termination locations in a criticality calculation converted from a HDF5 ptrac file. Black points represent particle surface crossing locations and colored points represent the energy of different particle events within the simulation.

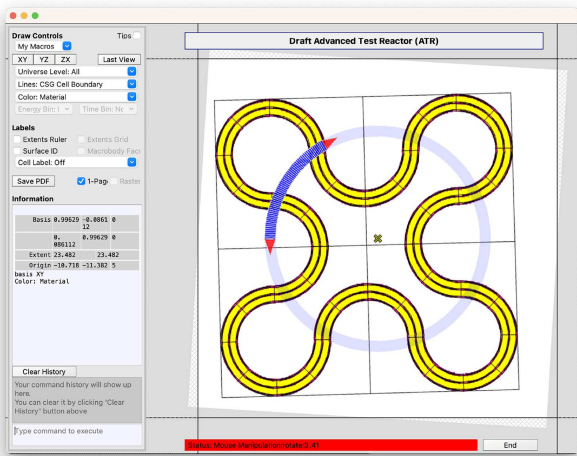


Fig. 7. An image of the modernized Qt-based geometry plotter released in MCNP6.3

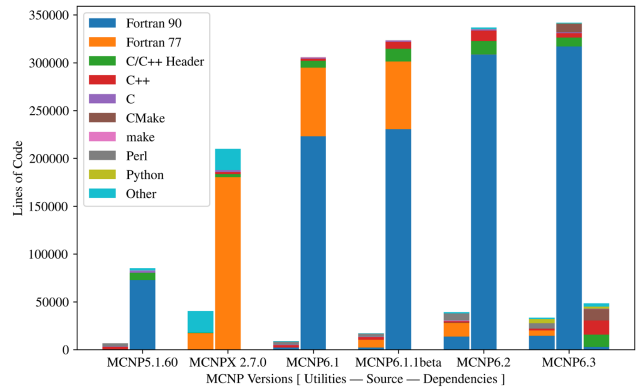


Fig. 8. Lines of code within the main source (largest bar), utilities (left-most bar), and additional standalone dependencies (right-most bar for MCNP6.3, not present for previous versions) of various MCNP code versions.

The most recent LANL HPC machine, Rocinante, includes Intel Xeon Platinum 8479 processors with 112 CPU cores and 256 GB memory per node. Because the processors

are composed of 8 logical domains with 14 CPU cores each, there can be significant non-uniform memory access (NUMA) computational penalties. That means, scaling up the number of OpenMP threads (i.e., `tasks` command-line option) on a single node will not be very performant. On

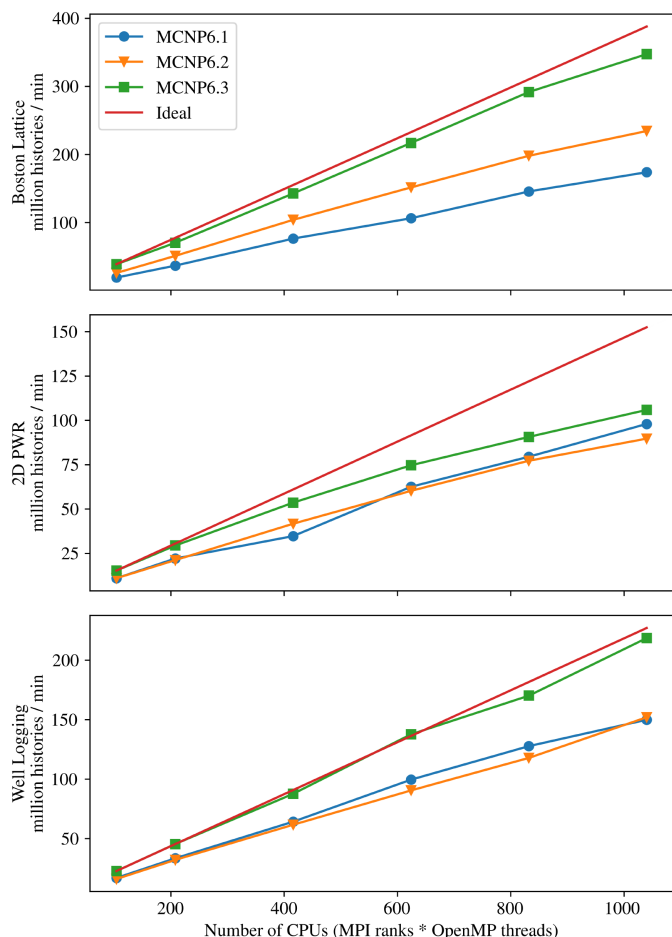


Fig. 9. Strong scaling of the MCNP6 code versions on the LANL Rocinante HPC platform.

systems like this, a mixture of MPI and OpenMP parallelism will provide the best computational performance for most applications.

In Figure 9 three separate real-world test problems are used to study the code performance, including:

Boston Lattice A fixed-source photon transport problem in a ~ 100 million element lattice model of the city of Boston.

2D PWR A k -eigenvalue neutron transport problem of a 2D pressurized water reactor, infinite in the axial direction.

Well Logging A fixed-source neutron transport problem with several reaction rate and energy spectrum tallies throughout the model.

Using a configuration that included 1 MPI rank assigned to each logical (NUMA) domain, 13 OpenMP threads for each MPI rank, scaling from 1 to 10 nodes, the strong scaling performance results are plotted for each of the MCNP6.1, MCNP6.2, and MCNP6.3 codes for each of the three test problems. The ideal lines on Figure 9 use the single node calculation times and scale the expected calculation rate for the number of CPUs in each of the subsequent calculation data points. For each test problem, the latest MCNP6.3 code is the most performant in part

due to several code improvements associated with fixing various NUMA issues and better chunk size management when passing messages across MPI ranks. This is encouraging to see good performance results in the latest code versions since this tends to mean that the code modernization and improvements made, even without a dedicated focus on improving code performance, are impacting the code in a positive direction.

4.3 Verification & validation

One of the most important aspects of the code that must be maintained or improved throughout all code changes now and going forward includes verification and validation (V&V) of the code capabilities. Verification is intended to test that the algorithms and numerical methods within the code can predict analytic or otherwise known results for simplified radiation transport problems. Validation includes measured benchmarks that are modeled within MCNP6 and compared directly to see how well the code and the nuclear data predict real world applications. In all releases of the MCNP6 code, substantial V&V has been performed and reported including the latest MCNP6.3 V&V report [46].

The overall quality and predictive power of MCNP6 is directly connected to the results of the V&V benchmark testing. Figure 10 shows that the k -eigenvalue predictions across multiple versions of MCNP6 have largely remained unchanged. For other suites of V&V test problems, including neutron time-of-flight, electron energy deposition, reactor kinetics, and high-energy physics benchmarks, consistency across MCNP6 code versions has also been observed where changes that occur in the MCNP6 predictions can often be tied back to particular bug fixes, code enhancements, or compiler changes.

5 The future of MCNP6

As was predicted, the MCNP code is not a static product. Even with a new release of MCNP6.3 in 2023, many code improvements and new features are already in progress or planned, including:

- Delta tracking for advanced reactor design calculations [88]
- A new SFC64 random number generator (`randgen=8`) [89]
- Improved parallelism for new computing architectures
- More microscopically correct particle physics, conserving event-by-event particle interaction-production correlations
- A Cinder code replacement for the legacy CINDER'90 code, including upgraded data, algorithms and testing
- Extensions of the burn-up depletion and UM tallies into the `turbotally` tally architecture
- Improved UM capabilities that facilitate better multi-physics coupling
- Charged particle transport improvements and extended validation
- A general perturbation/sensitivity capability for fixed-source and k -eigenvalue calculations

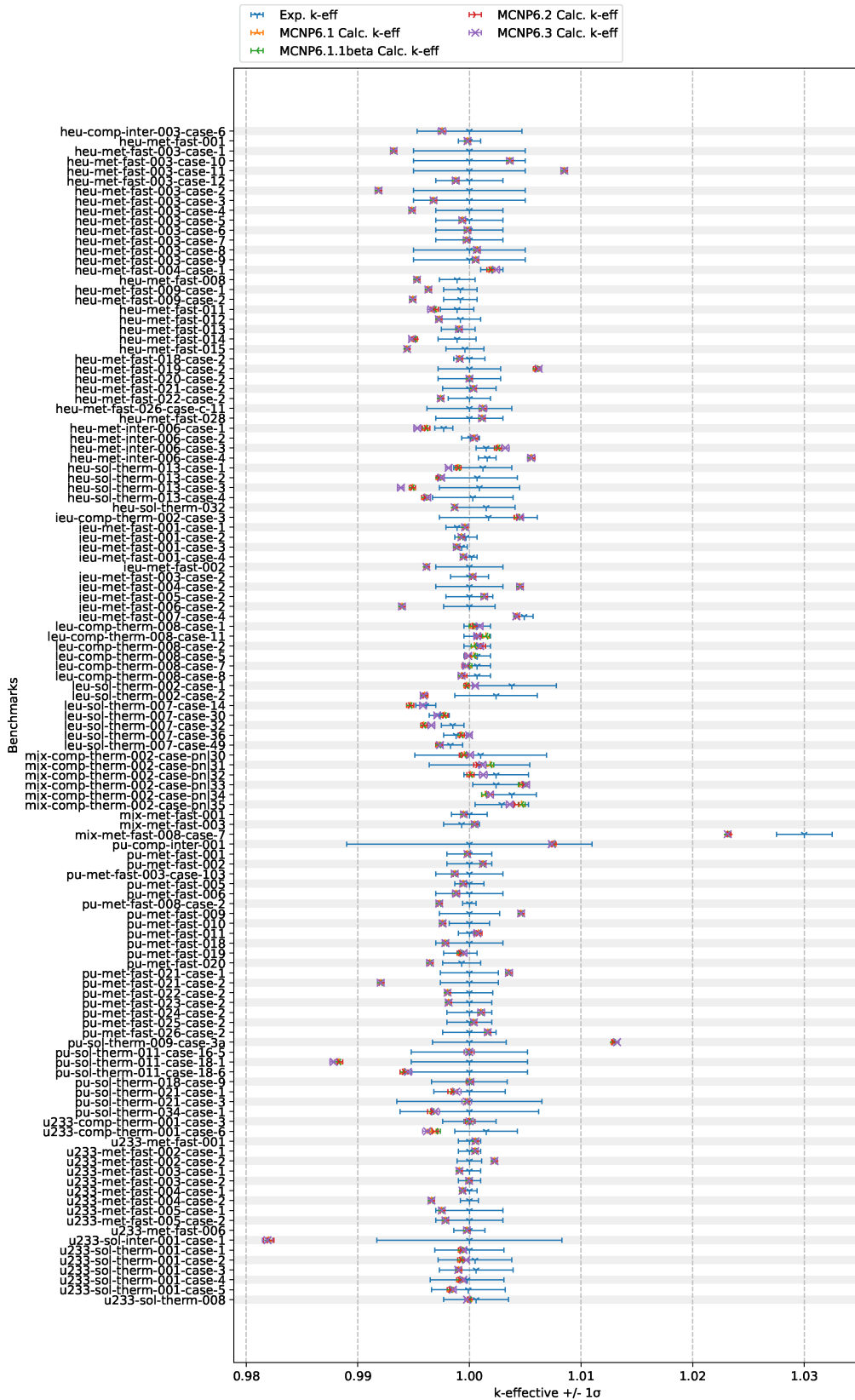


Fig. 10. The 119 criticality safety benchmark results for all versions of the MCNP6 code using the ENDF/B-VIII.0 nuclear data [54].

- More pre- and post-processing tools developed and released to the open-source community
- New analytic and semi-analytic benchmarks for extended verification testing
- Conversion of more MCNP formatted files to HDF5 and other standard formats
- A new Python-based input parser and interface to the code capabilities
- Improved adjoint-based variance reduction techniques (e.g., weight window enhancements)

The MCNP6 code is a fundamental product developed at LANL and is relied upon at LANL, within the National Nuclear Security Administration (NNSA) complex, the Department of Energy (DOE) Office of Science laboratories, and generally across the worldwide nuclear enterprise. The future of the MCNP6 code is bright with sustained efforts on code modernization, research and development, user support, education and outreach, and verification and validation, which will all undoubtedly continue on for the foreseeable future.

Acknowledgments

All of the MCNP6 development and progress made in the last decade would not have been possible without all of the efforts from previous developers of the code. Special thanks go to T. R. Adams, C. A. Anderson, T. E. Booth, F. B. Brown, T. P. Burke, L. J. Cox, D. A. Dixon, J. W. Durkee Jr., J. S. Elson, J. A. Favorite, A. J. Fallgren, M. L. Fensin, J. T. Goorley, T. S. Grieve, J. S. Hendricks, H. G. Hughes III, M. R. James, R. C. Johns, B. C. Kiedrowski, R. L. Martz, S. G. Mashnik, A. P. McCartney, G. W. McKinney, G. E. McMath, S. W. Mosher, E. J. Pearson, D. B. Pelowitz, R. E. Prael, C. J. Solomon Jr., A. Sood, T. J. Trahan, L. S. Waters, C. J. Werner, T. A. Wilcox, and S. C. Wilson. This work is supported by the Department of Energy through Los Alamos National Laboratory (LANL) operated by Triad National Security, LLC, for the National Nuclear Security Administration (NNSA) under Contract No. 89233218CNA000001.

Funding

This work is supported by the LANL MCNP Site Support Project, the LANL Engineering Campaigns, the DOE Nuclear Criticality Safety Program, and the DOE Advanced Scientific Computing Program.

Conflicts of interest

The authors declare that they have no competing interests to report.

Data availability statement

This article has no associated data generated and/or analyzed/data associated with this article cannot be disclosed due to legal/ethical/other reason.

Author contribution statement

M. E. Rising is responsible for MCNP6 project administration, is an active MCNP6 software developer, and lead the writing and editing of this manuscript. J. C. Armstrong, S. R. Bolding, J. S. Bull, L. Casswell, A. R. Clark, R. A. Forster, C. S. Frederick, J. F. Giron, F. B. Jones, C. J. Josey, T. M. Kelley,

J. A. Kulesza, M. A. Lively, R. C. Little, S. Swaminarayan, J. E. Sweezy, P. A. Vaquer, C. A. Weaver, and A. J. Zukaitis are active MCNP6 software developers, and reviewed and/or provided edits for this manuscript.

References

1. T. Goorley, M. James, T. Booth, F. Brown, J. Bull, L.J. Cox, J. Durkee, J. Elson, M. Fensin, R.A. Forster et al., *Nuclear Technol.* **180**, 298 (2012)
2. D.B. Pelowitz, J.T. Goorley, M.R. James, T.E. Booth, F.B. Brown, J.S. Bull, L.J. Cox, J.W. Durkee, Jr., J.S. Elson, M.L. Fensin et al., Tech. Rep. LA-CP-13-00634, Los Alamos National Laboratory, Los Alamos, NM, USA, 2013
3. J.E. Sweezy, T.E. Booth, F.B. Brown, J.S. Bull, R.A. Forster, III, J.T. Goorley, H.G. Hughes, III, R.L. Martz, R.E. Prael, A. Sood et al., Tech. Rep. LA-UR-03-1987 (Revised 2/1/2008), Los Alamos National Laboratory, Los Alamos, NM, USA, 2003, https://mcnp.lanl.gov/pdf_files/TechReport_2003_LANL_LA-UR-03-1987Revised212008_SweezyBoothEtAl.pdf
4. D.B. Pelowitz et al., Tech. Rep. LA-CP-11-00438, Los Alamos National Laboratory, Los Alamos, NM, USA, 2011
5. M.R. James, D.B. Pelowitz, A.J. Fallgren, G.E. McMath, T.E. Booth, F.B. Brown, J.S. Bull, L.J. Cox, J.S. Elson, J.W. Durkee, Jr. et al., Tech. Rep. LA-CP-14-00745, Los Alamos National Laboratory, Los Alamos, NM, USA, 2014
6. C.J. Werner, J.C. Armstrong, F.B. Brown, J.S. Bull, L. Casswell, L.J. Cox, D.A. Dixon, R.A. Forster, III, J.T. Goorley, H.G. Hughes, III et al., Tech. Rep. LA-UR-17-29981, Los Alamos National Laboratory, Los Alamos, NM, USA, 2017, https://mcnp.lanl.gov/pdf_files/TechReport_2017_LANL_LA-UR-17-29981_WernerArmstrongEtAl.pdf
7. J.A. Kulesza, T.R. Adams, J.C. Armstrong, S.R. Bolding, F.B. Brown, J.S. Bull, T.P. Burke, A.R. Clark, R.A. Forster, III, J.F. Giron et al., Tech. Rep. LA-UR-22-30006, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022, <https://www.osti.gov/biblio/1889957>
8. A. Sood, R.A. Forster, III, B.J. Archer, R.C. Little, *Nuclear Technol.* **207**, S100 (2021)
9. C.J. Josey, A. Sood, Presentation LA-UR-23-27907, Los Alamos National Laboratory, Los Alamos, NM, USA, 2023, Presented at International Topical Meeting on Industrial Radiation and Radioisotope Measurement Applications (IRRMA) in Bologna, Italy
10. W.L. Thompson, E.D. Cashwell, Tech. Rep. LA-UR-80-1158, Los Alamos National Laboratory, Los Alamos, NM, USA, 1980
11. ABAQUS/CAE User's Guide, Version 6.14, Dassault Systèmes Simulia, Inc., Providence, RI, USA (2014)
12. R.L. Martz, *Nuclear Technol.* **180**, 316 (2012)
13. J.C. Armstrong, J.A. Kulesza, C.J. Josey, S.W. Mosher, S. Swaminarayan, K.C. Kelley, J.L. Alwin, J.B. Spencer, V.K. Mehta, Presentation LA-UR-21-26436, Los Alamos National Laboratory, Los Alamos, NM, USA, 2021
14. ICRP, *Ann ICRP* **49**, 13 (2020)
15. B.C. Kiedrowski, F.B. Brown, P.P.H. Wilson, *Nucl. Sci. Eng.* **168**, 226 (2011)
16. B.C. Kiedrowski, F.B. Brown, *Nucl. Sci. Eng.* **174**, 227 (2013)
17. S.G. Mashnik, K.K. Gudima, R.E. Prael, A.J. Sierk, M.I. Baznat, N.V. Mokhov, Tech. Rep. LA-UR-08-02931, Los

- Los Alamos National Laboratory, Los Alamos, NM, USA, 2008, https://mcnp.lanl.gov/pdf_files/TechReport_2008_LANL_LA-UR-08-02931_MashnikPraelEtAl.pdf
18. H.G. Hughes, III, *Prog. Nucl. Sci. Technol.* **4**, 454 (2014)
 19. H.G. Hughes, III, *Tech. Rep. LA-UR-13-27377*, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2015
 20. M. Chadwick, M. Herman, P. Obložinský, M. Dunn, Y. Danon, A. Kahler, D. Smith, B. Pritychenko, G. Arbanas, R. Arcilla et al., *Nucl. Data Sheets* **112**, 2887 (2011)
 21. J.L. Conlin, D.K. Parsons, S.J. Gardiner, A.C. Kahler, III, M.B. Lee, M.C. White, M.G. Gray, *Tech. Rep. LA-UR-13-20137*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2013, https://mcnp.lanl.gov/pdf_files/TechReport_2013_LANL_LA-UR-13-20137_ConlinParsonsEtAl.pdf
 22. J.L. Conlin, D.K. Parsons, *Tech. Rep. LA-UR-14-21878*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2014, https://mcnp.lanl.gov/pdf_files/TechReport_2014_LANL_LA-UR-14-21878_ConlinParsons.pdf
 23. J.T. Goorley, *Tech. Rep. LA-UR-14-24680*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2014, https://mcnp.lanl.gov/pdf_files/TechReport_2014_LANL_LA-UR-14-24680_Goorley.pdf
 24. S.T. Cowell, W.B. Wilson, A.C. Hayes, P. Moller, T.R. England, *Tech. Rep. LA-UR-07-08412*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2007
 25. J.S. Hendricks, G.W. McKinney, M.L. Fensin, M.R. James, R.C. Johns, J.W. Durkee, Jr., J.P. Finch, D.B. Pelowitz, L.S. Waters, M.W. Johnson et al., *Tech. Rep. LA-UR-08-02216*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2008
 26. G.W. McKinney, F.B. Brown, H.G. Hughes, III, M.R. James, R.L. Martz, G.E. McMath, T.A. Wilcox, *Tech. Rep. LA-UR-14-23108*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2014, https://mcnp.lanl.gov/pdf_files/TechReport_2014_LANL_LA-UR-14-23108_McKinneyBrownEtAl.pdf
 27. F.B. Brown, B.C. Kiedrowski, J.S. Bull, *Tech. Rep. LA-UR-14-22480*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2014, https://mcnp.lanl.gov/pdf_files/TechReport_2014_LANL_LA-UR-14-22480_BrownKiedrowskiEtAl.pdf
 28. C.J. Werner, J.S. Bull, C.J. Solomon, Jr., F.B. Brown, G.W. McKinney, M.E. Rising, D.A. Dixon, R.L. Martz, H.G. Hughes, III, L.J. Cox et al., *Tech. Rep. LA-UR-18-20808*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2018, https://mcnp.lanl.gov/pdf_files/TechReport_2018_LANL_LA-UR-18-20808_WernerBulleEtAl.pdf
 29. J.R. Tutt, C.A. Anderson, G.W. McKinney, *Tech. Rep. LA-UR-16-24657*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2016, https://mcnp.lanl.gov/pdf_files/TechReport_2016_LANL_LA-UR-16-24657_TuttAndersonEtAl.pdf
 30. J.R. Tutt, G.W. McKinney, T.A. Wilcox, *Tech. Rep. LA-UR-16-23338*, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2016, https://mcnp.lanl.gov/pdf_files/TechReport_2016_LANL_LA-UR-16-23338Rev.1_TuttMcKinneyEtAl.pdf
 31. P. Talou, I. Stetcu, P. Jaffke, M.E. Rising, A.E. Lovell, T. Kawano, *Comput. Phys. Commun.* **269**, 108087 (2021)
 32. J.M. Verbeke, J. Randrup, R. Vogt, *Tech. Rep. LLNL-SM-705798*, Lawrence Livermore National Laboratory, Livermore, CA, USA, 2016
 33. P. Talou, R. Vogt, J. Randrup, M.E. Rising, S. Pozzi, J. Verbeke, M.T. Andrews, S. Clarke, P. Jaffke, M. Jandel et al., *Eur. Phys. J. A* **54**, 9 (2018)
 34. B.C. Kiedrowski, F.B. Brown, J.L. Conlin, J.A. Favorite, A.C. Kahler, III, A.R. Kersting, D.K. Parsons, J.L. Walker, *Nucl. Sci. Eng.* **181**, 17 (2015)
 35. F.B. Brown, M.E. Rising, J.L. Alwin, *Tech. Rep. LA-UR-17-20567*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2017, <https://www.osti.gov/biblio/1341834>
 36. R.C. Little, T. Kawano, G.M. Hale, M.T. Pigni, M.W. Herman, P. Obložinský, M.L. Williams, M.E. Dunn, G. Arbanas, D. Wiarda et al., *Nuclear Data Sheets* **109**, 2828 (2008)
 37. *Tech. Rep. ICSBEP-2015*, OECD Nuclear Energy Agency, Paris, France, 2015
 38. P.A. Grechanuk, M.E. Rising, T.S. Palmer, *J. Comput. Theor. Trans.* **47**, 552 (2018)
 39. C.J. Solomon, Jr., C.R. Bates, *Tech. Rep. LA-UR-17-21779*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2017, MCNPTools Version 3.8, https://mcnp.lanl.gov/pdf_files/TechReport_2017_LANL_LA-UR-17-21779_SolomonBates.pdf
 40. M.T. Andrews, C.R. Bates, E.A. McKigney, A. Sood, C.J. Solomon, Jr., *Presentation LA-UR-16-27166*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2016, https://mcnp.lanl.gov/pdf_files/TechReport_2016_LANL_LA-UR-16-27166_AndrewsBatesEtAl.pdf
 41. C.J. Solomon, Jr., *Tech. Rep. LA-UR-17-22234*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2017, ISC Version 1.3, https://mcnp.lanl.gov/pdf_files/TechReport_2017_LANL_LA-UR-17-22234_Solomon.pdf
 42. C.J. Solomon, Jr., *Tech. Rep. LA-UR-12-20252*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2012, https://mcnp.lanl.gov/pdf_files/TechReport_2012_LANL_LA-UR-12-20252_Solomon.pdf
 43. H.G. Hughes, III, *Tech. Rep. LA-UR-16-20840*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2016, https://mcnp.lanl.gov/pdf_files/TechReport_2016_LANL_LA-UR-16-20840_Hughes.pdf
 44. D.A. Dixon, H.G. Hughes, III, *Tech. Rep. LA-UR-16-29085*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2016, https://mcnp.lanl.gov/pdf_files/TechReport_2016_LANL_LA-UR-16-29085_DixonHughes.pdf
 45. D.K. Parsons, M.C. White, *Tech. Rep. LA-UR-14-23361*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2014, https://mcnp.lanl.gov/pdf_files/TechReport_2014_LANL_LA-UR-14-23361_ParsonsWhite.pdf
 46. M.E. Rising, J.C. Armstrong, S.R. Bolding, F.B. Brown, J.S. Bull, T.P. Burke, A.R. Clark, D.A. Dixon, R.A. Forster, III, J.F. Giron et al., *Tech. Rep. LA-UR-22-33103*, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2023, <https://www.osti.gov/biblio/1909545>
 47. F.B. Brown, C.J. Josey, S. Henderson, W.R. Martin, *Tech. Rep. LA-UR-19-23887*, Los Alamos National Laboratory, Los Alamos, NM, USA, 2019, https://mcnp.lanl.gov/pdf_files/TechReport_2019_LANL_LA-UR-19-23887_BrownJoseyEtAl.pdf

48. C. Josey, Comparing the performance of the new tally architectures in the MCNP code, in *Proceedings of the International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering (M&C 2023)* (American Nuclear Society, Niagara Falls, Ontario, CA, August 13–17, 2023)
49. The HDF Group, *Hierarchical Data Format, Version 5*, Website (2020), <http://www.hdfgroup.org/HDF5/>
50. J.A. Clarke, E.R. Mark, Enhancements to the eXtensible Data Model and Format (XDMF), in *HPCMP User's Group Conference 2007. High Performance Computing Modernization Program: A Bridge to Future Defense* (Pittsburgh, PA, USA, June 18–21, 2007), pp. 322–327
51. *XDMF Model and Format*, Website (2020), http://xdmf.org/index.php/XDMF_Model_and_Format
52. U. Ayachit, *The ParaView Guide*, community edn. (Kitware Inc., 2018)
53. J.P. Ahrens, B. Geveci, C. Law, *Visualization Handbook* (Elsevier Inc., Burlington, MA, USA, 2005), chap. ParaView: An End-user Tool for Large-data Visualization, pp. 717–731, <https://www.sciencedirect.com/book/9780123875822/visualization-handbook>
54. D.A. Brown, M.B. Chadwick, R. Capote, A. Kahler, A. Trkov, M. Herman, A. Sonzogni, Y. Danon, A. Carlson, M. Dunn et al., *Nuclear Data Sheets* **148**, 1 (2018)
55. J.L. Conlin, W. Haack, D.K. Parsons, D. Neudecker, M.C. White, Tech. Rep. LA-UR-18-24034, Los Alamos National Laboratory, Los Alamos, NM, USA, 2018
56. D.K. Parsons, C.A. Toccoli, Tech. Rep. LA-UR-20-24456, Los Alamos National Laboratory, Los Alamos, NM, USA, 2020
57. D.K. Parsons, C. Toccoli, Tech. Rep. LA-UR-21-23357, Los Alamos National Laboratory, Los Alamos, NM, USA, 2020, <https://www.osti.gov/biblio/1776727/>
58. A. Kahler, R. MacFarlane, R. Mosteller, B. Kiedrowski, S. Frankle, M. Chadwick, R. McKnight, R. Lell, G. Palmiotti, H. Hiruta et al., *Nuclear Data Sheets* **112**, 2997 (2011)
59. C.J. Josey, A.R. Clark, J.A. Kulesza, E.J. Pearson, M.E. Rising, Tech. Rep. LA-UR-22-32951, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022, <https://www.osti.gov/biblio/1907750>
60. J.A. Kulesza, Presentation LA-UR-23-32743, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2024, <https://www.osti.gov/biblio/2290286>
61. M.A. Lively, D. Perez, B.P. Uberuaga, X. Tang, *Rad. Phys. Chem.* **216**, 111483 (2024)
62. J.E. Sweezy, Tech. Rep. LA-UR-11-01825, Los Alamos National Laboratory, Los Alamos, NM, USA, 2011, https://mcnp.lanl.gov/pdf_files/TechReport_2011_LANL_LA-UR-11-01825_Sweezy.pdf
63. G.W. McKinney, M.R. James, J.H. Armstrong, J.M. Clem, Tech. Rep. LA-UR-12-00196, Los Alamos National Laboratory, Los Alamos, NM, USA, 2012, https://mcnp.lanl.gov/pdf_files/TechReport_2012_LANL_LA-UR-12-00196_McKinneyJamesEtAl.pdf
64. J. Palomares, G.W. McKinney, Tech. Rep. LA-UR-13-20254, Los Alamos National Laboratory, Los Alamos, NM, USA, 2013, https://mcnp.lanl.gov/pdf_files/TechReport_2013_LANL_LA-UR-13-20254_PalomaresMcKinney.pdf
65. T.A. Wilcox, G.W. McKinney, T. Kawano, Tech. Rep. LA-UR-14-21300, Los Alamos National Laboratory, Los Alamos, NM, USA, 2014, https://mcnp.lanl.gov/pdf_files/TechReport_2014_LANL_LA-UR-14-21300_WilcoxMcKinneyEtAl.pdf
66. R.B. Wilkerson, G.W. McKinney, M.E. Rising, J.A. Kulesza, Tech. Rep. LA-UR-20-27819, Los Alamos National Laboratory, Los Alamos, NM, USA, 2020
67. A.J. Zukaitis, R.A. Forster, III, R.R. Picard, Presentation LA-UR-21-26639, Los Alamos National Laboratory, Los Alamos, NM, USA, 2021
68. J.A. Kulesza, Memorandum XCP-3:22-13/LA-UR-22-32444, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022, Up-to-date details are available on the MCNP website.
69. J.C. Armstrong, M.R. Dzur, K.C. Kelley, C.A. D'Angelo, Presentation LA-UR-22-30643, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022
70. J.A. Kulesza, Tech. Rep. LA-UR-20-27150, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022, <https://www.osti.gov/biblio/1888190>
71. J.A. Kulesza, R.L. Martz, *Nucl. Technol.* **195**, 55 (2016)
72. J.A. Kulesza, Tech. Rep. LA-UR-22-30007, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022, <https://www.osti.gov/biblio/1889958>
73. E. Gonzalez, J.C. Armstrong, J.R. Tutt, Tech. Rep. LA-UR-22-33091, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022, <https://www.osti.gov/biblio/1907753>
74. M.R. Dzur, J.C. Armstrong, C.A. D'Angelo, MCNP6.3 unstructured mesh verification for OKTAVIAN geometries, in *Transactions of the American Nuclear Society* (2023), Vol. 128, pp. 858–861, <https://www.ans.org/pubs/transactions/article-53483>
75. B.J. Gladden, J.C. Armstrong, K.C. Kelley, Presentation LA-UR-23-30406, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2023, <https://www.osti.gov/biblio/2005769>
76. J.A. Kulesza, R.L. Martz, *Nucl. Technol.* **195**, 44 (2016)
77. J.A. Kulesza, R.L. Martz, Evaluation of the pool critical assembly benchmark with explicitly modeled geometry using MCNP6's unstructured mesh capabilities, in *Proceedings of Sixteenth International Symposium on Reactor Dosimetry*, edited by M.H. Sparks, K.R. DePriest, D.W. Vehar, ASTM International (ASTM International, Santa Fe, NM, USA, 2017), Vol. STP 1608, pp. 437–445, <https://www.astm.org/stp160820170041.html>
78. J.L. Alwin, J.B. Spencer, Presentation LA-UR-19-26393, Los Alamos National Laboratory, Los Alamos, NM, USA, 2019
79. J.L. Alwin, J.B. Spencer, Presentation LA-UR-21-25913, Los Alamos National Laboratory, Los Alamos, NM, USA, 2021
80. H. Nifenecker, C. Signarbieux, M. Ribrag, J. Poitou, J. Matuszek, *Nucl. Phys. A* **189**, 285 (1972)
81. S.A. Pozzi, B. Wieger, A. Enqvist, S.D. Clarke, M. Flaska, M. Marcath, E. Larsen, R.C. Haight, E. Padovani, *Nucl. Sci. Eng.* **178**, 250 (2014)
82. M.E. Rising, P. Talou, I. Stetcu, P. Jaffke, A.E. Lovell, T. Kawano, Presentation LA-UR-21-26275, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2021, <https://www.osti.gov/biblio/1813823-open-source-release-cgmf-integration-mcnp6-code>
83. F.B. Brown, C.J. Josey, Tech. Rep. LA-UR-18-27656, Los Alamos National Laboratory, Los Alamos, NM, USA, 2018

84. M.E. Rising, A.R. Clark, J.L. Alwin, Tech. Rep. LA-UR-23-25883, Los Alamos National Laboratory, Los Alamos, NM, USA, 2023, <https://www.osti.gov/biblio/2204814>
85. J.A. Kulesza, C.J. Josey, Presentation LA-UR-22-31281, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022, <https://www.osti.gov/biblio/1894826>
86. P. Pébay, T.B. Terriberry, H. Kolla, J. Bennett, Comput. Stat. **31**, 1305 (2016)
87. J.A. Kulesza, Tech. Rep. LA-UR-23-33941, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2023, <https://www.osti.gov/biblio/2323510>
88. C.J. Josey, Presentation LA-UR-22-30536, Los Alamos National Laboratory, Los Alamos, NM, USA, 2022
89. C.J. Josey, Tech. Rep. LA-UR-23-25111, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA, 2023, <https://www.osti.gov/biblio/1998091>

Cite this article as: Michael E. Rising, Jerawan C. Armstrong, Simon R. Bolding, Jeffrey S. Bull, Laura Casswell, Alexander R. Clark, R. Arthur Forster, Cole S. Frederick, Jesse F. Giron, Fred B. Jones, Colin J. Josey, Timothy M. Kelley, Joel A. Kulesza, Michael A. Lively, Robert C. Little, Sriram Swaminarayan, Jeremy E. Sweezy, Pablo A. Vaquer, Colin A. Weaver, Anthony J. Zukaitis. The MCNP[®]6 code: A decade of progress, EPJ Nuclear Sci. Technol. **11**, 9 (2025). <https://doi.org/10.1051/epjn/2025003>.