

MC21 v.10 – coupled-radiation Monte Carlo transport solver with support for multiphysics simulations

David P. Griesheimer^{*}, Daniel F. Gill, Brian R. Nease, Andrew Pavlou, Mark H. Stedry, Jason T. Thompson, Paul E. Burke and Peter S. Dobreff

Naval Nuclear Laboratory, Pittsburgh, Pennsylvania, United States

Received: 15 June 2024 / Received in final form: 23 September 2024 / Accepted: 16 October 2024

Abstract. Over the past decade, the development of the MC21 Monte Carlo radiation transport solver has focused on extending the functionality of the code beyond static calculations of reactivity and reaction rate distributions, as well as improving accuracy, performance, and scalability. Notable improvements include enhanced interaction physics models, efficient model representation and tracking algorithms, support for coupled physics calculations using both in-line and externally coupled feedback modules, and the development of native visualization and results post-processing tools. As a result of these improvements, MC21 v.10 provides a comprehensive analysis framework that allows complicated engineering problems to be solved with minimal reliance on external tools.

1 Introduction

In 2003, Bettis and Knolls Laboratories, now known as the Naval Nuclear Laboratory, began the development of a new Common Monte Carlo Design Tool (CMCDT). The objective of this development effort was to enable the use of Monte Carlo transport for routine reactor analysis. Previously, Monte Carlo transport simulations were reserved for benchmarking studies and used as a “tool-of-last-resort” due to the large memory and runtime requirements for full core calculations. The CMCDT includes three major software components: the Physics Unified Modeling and Analysis (PUMA) tool for model creation and visualization, the MC21 Monte Carlo transport solver, and the Nuclear Data Extractor (NDEX) for processing of raw evaluated nuclear data into a form suitable for use by MC21.

The initial development of CMCDT focused on delivering traditional reactor analysis capabilities, with support for common feedback effects such as depletion, equilibrium Xenon, and thermal feedback. Emphasis was placed on performance and scalability, to achieve acceptable through-life analysis throughput rates for high-fidelity, full-core reactor models. Reference [1] describes the features and capabilities of MC21 and NDEX as of 2013. By this time, CMCDT was already established as a design tool and used for routine analysis of 3D, full-core models containing 10^6 – 10^8 depletable compositions.

* e-mail: david.griesheimer@unnpp.gov

Since 2013, CMCDT development has focused on extending analysis capabilities beyond the calculation of reactivity and power distributions. Support for calculating auxiliary design products such as indirect neutron and photon heating, decay heating, and kinetics parameters have been added. Portions of the MC21 solver were refactored to improve performance, incorporate more accurate physics models, enable coupling with external codes for multiphysics simulations, and generally provide quality-of-life improvements for the user community. With these improvements, the past decade has seen MC21 evolve from a capable Monte Carlo depletion solver into a complete, all-in-one reactor analysis suite. The remainder of this paper provides descriptions of new features and improvements made to CMCDT since 2013, which are available for use in MC21 v.10. For readability, the paper is divided into three sections, which describe enhancements to MC21 analysis capabilities, improvements to the Monte Carlo solver, and descriptions of recent performance benchmarking, respectively.

2 Analysis capability enhancements

This section describes recent improvements to the CMCDT that improve the usability of the code suite, application to additional types of reactor analysis calculations, and support for coupling MC21 with external solvers for multiphysics calculations.

2.1 User experience improvements

Notable MC21 user experience improvements during the past decade include the implementation of a native application program interface (API) for performing calculations and querying model information and results, and the development of stand-alone applications for manipulating, visualizing, and processing model and results information.

In 2018, a C API for accessing native MC21 functions was implemented. The MC21 API supports the execution of sequential control commands, using the same syntax as the traditional MC21 control input file (`mc21_input`). In addition, the API can be used to query model information, such as material, component, and grid definitions, and perform point location and surface intersection calculations for use by external visualization and domain mapping/homogenization tools. The MC21 API does not provide interfaces for model construction or modification beyond what is supported through the standard MC21 ASCII control input file. In addition to the C API, a Python version of the API, referred to as `mcpy`, is also available.

The MC21 Native Geometry Input Tool (MCNGIT) provides an intuitive interface into MC21 for analyzing and manipulating complex models. Multiple models can be loaded into memory at once and manipulated in a variety of ways. For instance, components can be moved or deleted within a model or even copied from one model to another. When a component is moved within a model, all relevant attributes are automatically renumbered, including all depletion information. MCNGIT provides a convenient Python interface, but the underlying routines are compiled directly against native MC21 functions. This allows MCNGIT to use, and extend, the same data structures and classes as MC21, resulting in a natural consistency between the codes.

A Python package named Miner has been developed to aid in the processing of MC21 output. The miner can read any of the top-level HDF5 output data groups and cast the data into either a Pandas DataFrame or Xarray Dataset object. This includes but is not limited to, tally data, nuclide loadings, particle sources, and temperature distributions. Data can be readily manipulated in Python via the Pandas and Xarray data representations. Miner also builds on these packages to streamline and automate many operations of interest. MC21 model attributes can be associated with output, which allows aggregation of results by one or more attributes. For example, detailed nuclide loading data can be collapsed using pin and assembly attributes to provide 2D mass loading tables indexed by pin/assembly. Miner also supports the interpretation and propagation of uncertainty data in several different ways. Population statistics (mean and variance) on any output from independent simulations using different random number seeds can be easily produced. In addition, Miner uses the ‘uncertainties’ Python package to propagate uncertainty information (including simple correlations) through most mathematical operations in Pandas and Xarray.

Finally, the MC21 model and results visualization are provided by a 3D ray tracing visualization code referred

to as MCVIZ. MCVIZ offers a Python-based GUI front-end (Fig. 1) that can connect to different ray tracing kernel backends, including the native MC21 particle tracking kernel via the MC21 API. Alternatively, MCVIZ can be run in batch mode to create pre-defined plots for model verification or post-processing as part of an automated analysis workflow. MCVIZ provides a convenient and flexible interface for selecting model features based on any combination of cell property and/or model attribute. Selected parts of the model can be hidden, isolated, or made partially transparent. MCVIZ can also interface with movable geometry via the MC21 API, allowing for the translation or rotation of any defined movable components or groups in the model. Results visualization in MCVIZ (shown in Fig. 1) leverages the MC21 API and Miner post-processing tool to display requested tally results on the ray-traced representation of the native geometry. Tally data is processed by Miner and passed, in memory, to MCVIZ as an Xarray object, which allows efficient access to the data during visualization.

2.2 Feedback and code coupling

MC21 v.6 included in-line calculation of reactor feedback effects such as depletion, thermal-hydraulic feedback, photon heating, control rod criticality search, equilibrium xenon, and calculation of peak xenon mass after shutdown. In 2016, a novel algorithm for calculating the peak reactivity worth of xenon following an arbitrary change in reactor power was added to the code [2].

In-line feedback calculations in MC21 utilize a fixed-point (Picard) iteration scheme between the neutron transport simulation and the corresponding feedback (e.g., thermal) solver. In the original MC21 implementation, feedback iterations were performed independently, requiring separate, fully converged, transport simulations for each effect. In 2020, an alternate nonlinear feedback strategy was tested in MC21, which allows simultaneous convergence of local properties (e.g., temperature) and the fission source distribution during a single k_{eff} transport calculation [3].

In 2018, the eigenvalue search capability in MC21 [1,4] was expanded to support criticality searches based on soluble boron concentration. Additionally, the eigenvalue search algorithm in MC21 was updated to improve both its robustness and performance and to simplify its use by removing most user-defined input parameters.

Additional improvements to in-line feedback effects include investigation, optimization, and standardization of running strategies for predictor-corrector constant-power depletion calculations [5,6], and implementation of an in-line residual fission product treatment to account for reactivity effects due to fission products not explicitly included in the model [6].

To enable seamless coupling with external solvers, an extensive and flexible system for exporting and importing data to/from a file was implemented in MC21. To this end, MC21 has adopted an output approach that standardizes all program output, other than terminal output, in HDF5 format. These HDF5 output files are composed of

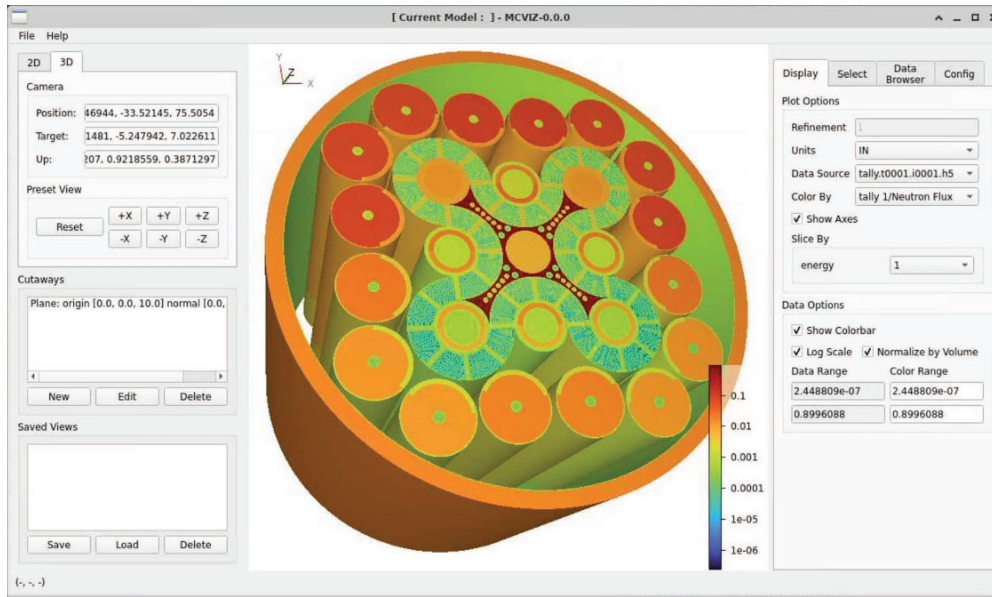


Fig. 1. Screenshot of the MCVIZ tool visualizing tally results (neutron flux in an energy group) across a model of the Advanced Test Reactor (ATR), shown with reflectors hidden.

one or more top-level HDF5 groups (folders). All possible data groups are exhaustively defined in the MC21 reference manual. Any nested data groups are fully defined, and all low-level datasets are described in terms of their types, dimensions, and contents. Any given data group may then appear as output in a variety of contexts. For example, the `/source` data group, which contains particle census data, may appear in the restart file of a k_{eff} calculation containing the converged fission source or as the output of a particle census tally that was defined to collect only particles crossing a particular surface within a particular energy range. This same data group is then recognized in all contexts where MC21 allows the loading of a source definition from an external file.

Using this standardized HDF5 format, fundamental model properties such as temperature, volume, and loading can be exported to file for analysis, manipulation, reimport, or exchange with external solvers. Property data can be exported by spatial cell or aggregated over any attribute (e.g., material) prior to export. Volume-integrated quantities, such as mass or atom loading, may be exported based on volumes provided as a part of the geometry input or stochastically calculated volumes determined by MC21 itself. Simulation “state” data, such as nuclide densities, temperature and xenon distributions, movable geometry positions, and fission sources, can also be exported to an HDF5 file. This data can then be partially or wholly re-imported into subsequent MC21 simulations, providing a simple mechanism for sharing data between simulations. In 2019, the data exchange format was used to couple MC21 with the CFX computational fluid dynamics code to investigate strategies for high-fidelity coupled nuclear/thermal calculations [7].

In addition to file-based data exchange, MC21 supports native integration with the PVMEXEC coupling framework [8]. PVMEXEC provides a generic program

interface for exchanging data between solver kernels for different physics processes, such as radiation transport, heat conduction, and fluid flow. Successful integration with PVMEXEC was demonstrated in 2014 by coupling MC21 to the subchannel thermal-hydraulics solver COBRA-IE through the framework [9].

To facilitate data exchange with mesh-based solvers, MC21 supports stochastic homogenization of the model geometry over an arbitrary mesh overlay. The stochastic homogenization capability uniformly samples a fixed, user-defined, number of points within each mesh cell to determine which MC21 spatial cells belong, in whole or in part, to the corresponding mesh cell. This cell-to-mesh mapping information is stored in an HDF5 “cell counts” file, which allows for subsequent conversion of MC21 attribute or property data to the mesh without needing to rerun the homogenization sampling process.

In 2017, the stochastic homogenization and mesh tally capabilities in MC21 were leveraged to provide support for the in-line generation of nodal coefficients for use in coarse-mesh neutron diffusion solvers. MC21 allows the superposition of a Cartesian mesh over any model and offers a calculation mode where nodal constants will be tallied in each mesh cell. Multigroup information calculated for each node includes diffusion coefficients, macroscopic cross-sections, surface fluxes, and partial currents. This data is suitable for use in the creation of nodal diffusion cross-sections, including the generation of discontinuity factors.

3 Solver methodology improvements

This section describes enhancements to the underlying algorithms and implementation details for the NDEX nuclear processing utility and MC21 transport solver

kernel, including improvements to nuclear data, interaction physics, particle tracking, and collection of tally results.

3.1 Nuclear data processing

Within CMCDDT, the NDEX code is responsible for processing raw evaluated nuclear data and producing an HDF5 continuous energy data library (CE_LIBRARY) file that is read by MC21. Originally, NDEX relied on NJOY routines such as ACER and HEATR to preprocess ENDF-formatted data into a form that could be easily reformatted for use by MC21 [1]. Many of these preprocessing routines have been replaced by native versions implemented directly in NDEX. Implementation of native routines reduces the dependency on NJOY, modernizes the code base, and streamlines the preprocessing step.

In addition, NDEX routines for Doppler broadening of the resolved resonance region via SIGMA1 [10] were separated out into a stand-alone library, which is used by both NDEX and MC21. This enables in-line (pre-transport) generation of temperature-dependent cross-sections from an isothermal data library within MC21 itself.

CE_LIBRARY files are generated by running NDEX with an input file that specifies the requested nuclides, along with the corresponding evaluation libraries, and temperatures. In cases where multiple temperatures for a nuclide are requested, NDEX writes all the temperature-dependent data for each temperature requested but only writes the temperature-independent data once, thus reducing the size of the final CE_LIBRARY file.

Processing of thermal scattering law (TSL) data is performed by transforming thermal scattering cross-sections into probability distributions of two variables, α and β , which can be sampled directly [11,12]. This method retains the detail present in the TSL evaluation, while processing it into a form well-suited to MC transport simulations. In 2020, the TSL data processing algorithm was updated to improve the adaptive energy mesh used when generating scattering cross sections from the raw $S(\alpha,\beta)$ TSL data, in order to better capture fine detail present in the evaluated data [13].

Photoatomic data and atomic structure data for photon transport are also included in the CE_LIBRARY. Raw photon interaction data is taken from available evaluations such as the Evaluated Photon Data Library (EPDL) [14] and the Evaluated Atomic Data Library (EADL) [15]. X-ray fluorescence spectra are generated from electron binding energies and atomic relaxation data [16]. Subshell-specific photon incoherent scattering cross-sections are generated based on published electron momentum distributions [17,18]. These subshell-specific incoherent scattering reactions eliminate the need for scattering form factors, naturally incorporate the effects of photon Doppler broadening, and simplify sampling of fluorescence X-rays.

The CE_LIBRARY also includes data used for the production of bremsstrahlung photons [19] and parameters needed for the computation of the electron collision-stopping power. These data are derived from the synthesis

of different theoretical approaches and available numerical data [20].

3.2 Interaction physics

MC21 v.6 included support for traditional neutron-nucleus and photoatomic reaction types, as described in reference [1]. MC21 uses continuous-energy interaction physics models that preserve the accuracy and fidelity of the underlying evaluated nuclear data. Recent developments in physics modeling have focused on neutron resonance elastic scattering and high-fidelity photon physics.

For resonance elastic scattering of neutrons, MC21 supports the optional use of Doppler broadening rejection correction (DBRC) sampling. In 2019, the DBRC implementation was updated to directly sample the relative velocity between the neutron and target and perform rejection based on the cosine of the scattering angle, using the method described in reference [21]. Testing performed with low-enriched Uranium (LEU) and mixed-oxide (MOX) pin cell models shows a 10–20% speedup in neutron tracking rate using the improved implementation.

In-line sampling and tracking of fluorescence photons was added to MC21 in 2019 [16]. The sampling algorithm uses collapsed X-ray spectra to eliminate the complexity and computational cost of sampling a full relaxation cascade realization for each photoelectric and Compton scattering event. The collapsed spectra are pre-calculated from EADL transition probabilities for each subshell in every element and included in the CE_LIBRARY file along with other photoatomic data. During Monte Carlo transport the collapsed spectrum is used to sample secondary fluorescence photons just like any other secondary particle emission, yielding excellent agreement with traditional analog methods that directly sample the atomic relaxation cascades.

Detailed treatment of polarization effects for incoherent and coherent photon scattering and pair production was also added to MC21 in 2019 [22]. The implementation uses the relativistic impulse approximation to account for the momentum distribution of target electrons during Compton scattering. As a result, MC21 can explicitly represent photon Doppler broadening effects by including information about the precollision motion of atomic electrons.

Finally, in 2022, the photon incoherent scattering model in MC21 was updated to use tabulated subshell-specific reaction probabilities instead of form factors when sampling the outgoing photon energy and direction. The improved model ensures that the calculated photon energy deposition rates account for subshell-dependent reactions such as X-ray fluorescence and photon Doppler broadening.

3.3 Geometry

MC21 uses a hierarchical geometry representation that involves three distinct types of geometric objects: components, grids, and overlays [1]. Components are traditional

combinatorial solid geometry (CSG) volumes formed from intersections and unions of quadric surface half-spaces. Grids allow for subdivision of components into cells that can be assigned unique properties (e.g., materials). Overlays, which are based on standardized templates (e.g., an array of cylindrical rods), allow for efficient representation and particle tracking through repeating structures [1,23].

The smallest uniquely addressable volume of an MC21 model is referred to as a spatial cell. Originally, all model properties (e.g., material, region, temperature) were assigned on a per-cell basis. In 2019, the MC21 model representation was refactored to introduce the concept of a part, which is a collection of one or more spatial cells. Every cell in a model is assigned to a corresponding part. In the revised model, inherent cell characteristics such as volume, temperature, and local energy deposition are stored on a per-cell basis, while all other properties, referred to as “attributes”, are assigned by part. Native MC21 part attributes include material, composition, region, temperature region, and fluence region. As mentioned in Section 2.1, additional attributes (e.g., fuel assembly or pin identifiers) may be assigned to aid with collecting and/or post-processing results from the transport simulation. Assigning such attributes to parts rather than cells can significantly reduce memory for models with more cells than unique combinations of attributes.

The geometry representation in MC21 is based on parent-child hierarchy rather than the universe/fill approach used by many other codes. MC21 cell volumes are implicitly defined by the intersection of the boundaries of parent/ancestor (i.e., grandparent, etc.) cells and the complement of any contained (child) cells. As a result, child component volumes are automatically excluded from the parent component. In contrast, universe/fill approaches typically require the creation of non-overlapping cell volumes, either manually or through the explicit use of a complement operator.

The MC21 geometry representation is especially convenient for movable geometry, as the parent volume will automatically backfill when a child component is moved. However, the possibility of implicit boundary surfaces in component definitions prevents the use of traditional surface-neighbor-list algorithms for identifying adjacent components. As a result, MC21 constructs and optimizes component neighbor lists during particle tracking to improve performance. In addition, the particle tracking kernel includes logic to ensure consistency between distance to surface and surface sense calculations when crossing (nearly-) coincident surfaces. This logic can detect, and resolve, situations where a particle has an incorrect, or inconsistent, sense with respect to surfaces in the component hierarchy following a surface crossing event – often caused by finite precision arithmetic. Related tracking logic is used to determine if sibling components (i.e., components at the same level of hierarchy) are overlapping by more than the allowed tolerance.

To further improve MC21 tracking performance, support for delta tracking in model regions containing repeated overlay cells was added. Delta tracking is available for all overlay template types and is enabled as a part of the template definition. When enabled, MC21 automati-

cally calculates the majorant cross-section among materials in the overlay cells and parent component/grid cells. The use of delta tracking disables track-length tally estimators in the simulation but partially compensates by scoring collision tallies at both real and pseudo-collision events during particle tracking.

The MC21 geometry kernel also supports the use of mesh overlays for collecting tally results over spatial domains that are independent of the underlying model representation. Mesh overlays are commonly used to collect tallies on finite-element mesh for exchange with deterministic solvers, apply (FW-)CADIS variance reduction, and calculate Shannon entropy and dominance ratio for k_{eff} calculations. In previous releases, MC21 grids, which are tied to the underlying model representation, and mesh, which are independent of the model representation, used separate particle tracking algorithms. In 2019, mesh and grids were refactored to use a common underlying implementation, referred to as subdivision tracking. Presently, MC21 supports tracking over Cartesian, cylindrical, and spherical subdivision types, which can be used for either grid or mesh definitions.

3.4 Materials

MC21 v.6 provided basic functionality for defining materials, with support for void and single-phase materials, which are specified as a list of nuclide names and corresponding number densities. In 2016, support was added for multi-phase, mass-mixture material definitions. Mass-mixture materials include a background (base) phase along with an arbitrary number of additive phases, which are homogenized with the base phase. Homogenization of additive phases in the material is based on a user-specified mass concentration, which can be changed via user input between transport calculations within a single job execution. Mass-mixture materials are especially useful for modeling soluble poison letdown through life in pressurized water reactors (PWRs). MC21 also supports in-line criticality search by phase concentration in mass-mixture materials, as described in Section 2.2.

Additional improvements related to MC21 materials include support for compound material definitions and merged nuclides. Compound materials offer simplified input for creating new materials based on combinations of previously defined materials, which is convenient for defining a variety of homogenized metal-water materials such as support grids and assembly inlet/outlet nozzles in reactor models. Merged (composite) nuclides are combinations of low-reactivity nuclides that are treated as a single nuclide during particle transport [24]. The creation of merged nuclides is performed by NDEX, which maps cross-sections from all constituent nuclides in the merge onto a unionized energy mesh. During transport, the merged cross-section is used to calculate macroscopic cross-sections and relative reaction probabilities for the parent material. When a reaction with a merged nuclide is sampled, MC21 immediately resamples a specific target nuclide from among the merged constituents and uses the detailed reaction data from the sampled constituent for

the reaction. The use of merged nuclides to replace large numbers of low-reactivity nuclides, such as alloying elements in stainless steel, Inconel, or Zircaloy, can provide significant performance improvements for particle tracking. Additional performance improvements are achieved by using a lethargy-based hashed lookup scheme for microscopic cross-sections and caching microscopic cross-section data for reuse following surface crossing events, as described in reference [25].

3.5 Tallies

MC21 has always provided a flexible system for defining what information is to be collected and reported during a simulation. To this end, MC21 v.6 included two distinct types of tallies: integral tallies and census tallies [1]. Integral tallies use traditional collision and track length estimators to score integrated flux moment information using a user-defined volume or surface domain. Integral tallies are defined by a set of phase-space filters and scoring functions. The filters determine the tally domain by controlling which particles are allowed to score, while the scoring functions determine what type of flux moment (i.e., reaction rate) information is collected at each scoring event. Census tallies record particle state information at user-defined event types, creating a collection of particle information that can be post-processed or used directly as source particles in a subsequent MC21 simulation.

Over the past 5 years, the MC21 integral tally system was refactored into four distinct tally types: conventional, scatter, point detector, and property tallies. Each type offers a variety of related estimators and scoring functions related to the tally objective.

Conventional tallies include standard volume and surface reaction rate tallies available in previous versions of MC21. This tally type supports the use of track length, collision, and particle birth estimators for volumetric quantities, and a surface crossing estimator for scoring integrated flux and/or current over a surface domain.

Scattering tallies score at particle scattering events and allow the collection of information regarding the change in particle state during the collision, such as energy transfer. Scatter tallies always use a collision estimator; however, these tallies are unique in that they score the post-collision particle weight rather than the pre-collision particle weight like all other tally types. Scattering tallies are especially useful for estimating energy-transfer matrices and Legendre scattering moments for use with low-order, multi-group transport or diffusion solvers.

Point detector tallies allow reduced variance estimates for flux, reaction rate, and/or dose quantities at specific point locations in the model geometry. MC21 point tallies support both a traditional next-collision estimator and a kernel density estimator (KDE). The next-event estimator determines the probability that a particle will scatter towards the point detector at each collision event, and then creates a virtual pseudoparticle, which is transported towards the detector. The MC21 implementation includes several improvements over traditional next-collision estimator algorithms. Specific improvements include a pre-

collision estimator for photons that simultaneously considers detector scoring contributions for coherent, incoherent, and pair production scattering types, and rouletteing of pseudoparticles based on the space/energy variance reduction importance map applied to the problem. These improvements mitigate known issues with next-event estimators and can provide significant improvements in pseudoparticle tracking/scoring rates without affecting tally accuracy. The KDE point detector tally uses one of several available statistical kernels (uniform, triangular, Epanechnikov, or Gaussian) to calculate the expected detector contribution from any particle that has an interaction within a user-defined scoring domain about the detector point [26].

Property tallies are used to collect intrinsic information, such as volume, mass loading, and decay activity, from the model. This tally type is unique in that it collects information by iterating over all cells defined in the model rather than scoring particle events during a transport simulation. The property tally allows the aggregation of model information by attribute, which is useful for exchanging information with external codes.

Additional improvements to MC21 tallies include support for the use of phase-state filtering with census tallies, and the development of improved in-line estimators for effective neutron generation time (Λ_{eff}) and effective delayed neutron fraction (β_{eff}) using correlated sampling, as described in references [27–29].

4 Performance and benchmarking

Like most large-scale Monte Carlo transport solvers, MC21 supports both message-passing parallelism between compute nodes and thread parallelism within nodes. To maximize performance in parallel calculations, MC21 uses specialized routines for random number generation, fission source renormalization, and fixed-source sampling, which have been optimized for hybrid message-passing/threaded parallelism [30]. For fixed source calculations, MC21 supports asynchronous parallelism, often referred to as ‘flurry’ parallelism, where independent transport calculations are run on separate compute nodes, using thread parallelism only, and then the results are combined in post-processing. Flurry calculations offer high throughput relative to traditional multi-node parallelism by avoiding long queuing times when requesting large numbers of nodes and eliminating load imbalances between nodes, which can be significant in simulations with aggressive space/energy importance-based variance reduction.

Results from performance and scalability testing of MC21 on massively parallel high-performance computing systems are provided in references [30,31]. These tests confirm that MC21 can run petaflop-scale simulations and model sizes approaching 12 Tb. In general, MC21 shows excellent parallel scalability (both strong and weak scaling) up through 1000 nodes. Additional MC21 performance and benchmarking data were obtained via simulations of the BEAVRS benchmark [32] and the Consortium for Advanced Simulation of Light Water Reactors (CASL) core physics challenge problems 6 and 7, as reported in ref-

erences [33,34]. In 2017, MC21 was the first code that was able to successfully analyze the challenge problem configurations and produce results with the specified level of accuracy.

5 Conclusions

Since 2013, improvements to MC21 and NDEX have provided additional analysis capabilities and facilitated increased use of the CMCDDT code suite for routine, high-fidelity reactor calculations. Support for detailed interaction physics, in-line sampling of intrinsic radiation sources, coupled radiation transport, and coupling with external physics solvers, along with performance, scalability, and usability improvements, have enabled the use of MC21 for a variety of analyses beyond traditional reaction rate and eigenvalue calculations.

Acknowledgments

The authors wish to acknowledge the contributions and support of the many MC21 developers and users from the last 20 years, including E. Caro, D. Carpenter, H. Joo, S. Marin, D. Millman, K. Remley, P. Romano, and T. Sutton (NNL), and A. Bird and C. Murphy (Jacobs) for development contributions; S. Douglass, A. Klinvex, J. Holmes, and M. Zerkle (NNL) for nuclear data and numerical methods support; and B. Aviles and D. Kelly (NNL) for code qualification and testing.

Funding

This manuscript has been authored by Fluor Marine Propulsion under contract No. DOE-89233018CNR000004 with the U.S. Department of Energy. The United States government retains and the publisher, by accepting this article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, and worldwide license to publish, distribute, translate, duplicate, exhibit, and perform the published form of this manuscript, or allow others to do so, for United States Government purposes.

This research made use of Idaho National Laboratory's High-Performance Computing systems located at the Collaborative Computing Center and supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

Conflicts of interest

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

Data availability statement

No data is associated with this article.

Author contribution statement

All authors contributed to the development and implementation of the technical work and the writing of the final manuscript.

References

1. D.P. Griesheimer et al., *Ann. Nucl. Energy* **82**, 29 (2015)
2. D.P. Griesheimer, *Trans. Am. Nucl. Soc.* **115**, 578 (2016)
3. D.F. Gill, Inline thermal and xenon feedback iterations in Monte Carlo reactor calculations, in *Proceedings of Physor 2020, 29 March – April 2* (Publisher: EPJ Web of Conferences, Cambridge, UK, 2020) <https://doi.org/10.1051/epjconf/202124704019>
4. D.F. Gill, B.R. Nease, D.P. Griesheimer, Movable geometry and eigenvalue search capability in the MC21 Monte Carlo Code, in *Proceedings of M&C 2013, 5-9 May* (Publisher: American Nuclear Society, Sun Valley, Idaho, 2013)
5. S.J. Spsychala, D.P. Griesheimer, Effects of stochastic noise on a three-dimensional Monte Carlo depletion analysis of the H.B. Robinson reactor, in *Proceedings of PHYSOR 2014, 28 September – 3 October* (Publisher: Japan Atomic Energy Agency, Kyoto, Japan, 2014)
6. D.P. Griesheimer, D.C. Carpenter, M.H. Stedry, *Prog. Nucl. Energy* **101**, 409 (2017)
7. K.E. Remley et al., Investigations into coupling the MC21 Monte Carlo transport code to the CFX computational fluid dynamics code, in *Proceedings of M&C 2019, 25-29 August* (Publisher: American Nuclear Society, Portland, Oregon, 2019)
8. D.L. Aumiller et al., *Nucl. Tech.* **193**, 183 (2016)
9. D.F. Gill, D.L. Aumiller, D.P. Griesheimer, Monte Carlo and thermal-hydraulic coupling via PVMEXEC, in *Proceedings of PHYSOR 2014, 28 September – 3 October* (Publisher: American Nuclear Society, Kyoto, Japan, 2014)
10. D.E. Cullen, C.R. Weisbin, *Nucl. Sci. Eng.* **60**, 199 (1976)
11. C.T. Ballinger, The direct $S(\alpha, \beta)$ method for thermal neutron scattering, in *Proceedings of M&C 1995, 30 April – 4 May* (Publisher: American Nuclear Society, Portland, Oregon, 1995)
12. T.H. Trumbull, Computational methods used to process thermal neutron scattering data for use in continuous energy Monte Carlo codes, in *Proceedings of PHYSOR 2016, 1-5 May* (Publisher: American Nuclear Society, Sun Valley, Idaho, 2016)
13. J.L. Wormald, J.T. Thompson, T.H. Trumbull, *Ann. Nucl. En.* **149**, 107773 (2020)
14. D.E. Cullen, J.H. Hubbell, L. Kissel, URCL-LR-50400, Vol 6, Rev 5, 1997
15. D.E. Cullen et al., UCRL-50400, Vol. 30, 1991
16. J.T. Thompson, D.P. Griesheimer, E. Caro, Pre-collapsed fluorescence methodology for Monte Carlo radiation transport simulations, in *Proceedings of M&C 2019, 25-29 August* (Publisher: American Nuclear Society, Portland, Oregon, 2019)
17. F. Biggs, L.B. Mendelsohn, J.B. Mann, *At. Data Nucl. Data Tables* **16**, 201 (1975)
18. M.C. White, LA-UR-03-1019, 2002
19. S.M. Seltzer, M.J. Berger, *Nucl. Instrum. Methods Phys. Res. B* **12**, 95 (1985)
20. R.M. Sternheimer, M.J. Berger, S.M. Seltzer, *At. Data Nucl. Data Tables* **30**, 261 (1984)
21. P.K. Romano, J.A. Walsh, *Ann. Nucl. Energy* **114**, 318 (2018)
22. E. Caro, Photon polarization and doppler broadening in low energy Monte Carlo transport, in *Proceedings of M&C 2019, 25-29 August* (Publisher: American Nuclear Society, Portland, Oregon, 2019)
23. B.R. Nease et al., Geometric templates for improved tracking performance in Monte Carlo codes, in *Proceedings of*

- SNA+MC 2013, 27-31 October* (Publisher: EDP Sciences, Paris, France, 2013)
24. D.P. Griesheimer, R.R. Gouw, *Trans. Am. Nucl. Soc.* **111**, 708 (2014)
 25. D.P. Griesheimer, *Trans. Am. Nucl. Soc.* **111**, 703 (2014)
 26. K. Banerjee, W.R. Martin, *Nucl. Sci. Energy* **174**, 30 (2013)
 27. D.P. Griesheimer, T.P. Goter, Estimating the effective neutron generation time with Monte Carlo correlated sampling, in *Proceedings of M&C+SNA+MC 2015, 19-23 April* (Publisher: American Nuclear Society, Nashville, Tennessee, 2015)
 28. D.P. Griesheimer, N.A. Gibson, Simplified method for estimating the effective delayed neutron fraction with Monte Carlo correlated sampling, in *Proceedings of M&C 2019, 25-29 August* (Publisher: American Nuclear Society, Portland, Oregon, 2019)
 29. D. Heinrichs, β_{eff} Benchmarks, LLNL-PRES-796197, in *Presentation to the Cross Section Evaluation Working Group, 4 November* (Publisher: National Nuclear Data Center, Brookhaven National Laboratory, Brookhaven, New York, 2019)
 30. D.P. Griesheimer et al., Strategies and algorithms for hybrid shared-memory/message-passing parallelism in Monte Carlo radiation transport codes, in *Proceedings of M&C+SNA+MC 2015, 19-23 April* (Publisher: American Nuclear Society, Nashville, Tennessee, 2015)
 31. D.P. Griesheimer, P.S. Dobreff, R. Gomez, Monte Carlo scalability on extreme memory computing nodes, in *Proceedings of M&C 2021, 3-7 October* (Publisher: American Nuclear Society, Raleigh, North Carolina, 2021)
 32. D.J. Kelly III et al., Analysis of select BEAVRS PWR benchmark cycle 1 results using MC21 and OpenMC, in *Proceedings of PHYSOR 2014, 28 September – 3 October* (Publisher: Japan Atomic Energy Agency, Kyoto, Japan, 2014)
 33. B.N. Aviles et al., *Prog. Nucl. Energy* **101**, 338 (2017)
 34. D.J. Kelly III et al., *Nucl. Eng. Tech.* **49**, 1328 (2017)

Cite this article as: David P. Griesheimer, Daniel F. Gill, Brian R. Nease, Andrew Pavlou, Mark H. Stedry, Jason T. Thompson, Paul E. Burke, Peter S. Dobreff. MC21 v.10 – coupled-radiation Monte Carlo transport solver with support for multiphysics simulations, *EPJ Nuclear Sci. Technol.* **10**, 23 (2024)