

MC21 v.6.0 – A Continuous-Energy Monte Carlo Particle Transport Code with Integrated Reactor Feedback Capabilities

D. P. Griesheimer^{1*}, D. F. Gill¹, B. R. Nease¹, T. M. Sutton², M. H. Stedry², P. S. Dobreff², D. C. Carpenter¹,
T. H. Trumbull², E. Caro², H. Joo¹, D. L. Millman¹

¹Bechtel Marine Propulsion Corporation – Bettis Atomic Power Laboratory

²Bechtel Marine Propulsion Corporation – Knolls Atomic Power Laboratory

*Corresponding Author, E-mail: david.griesheimer.contractor@unnpp.gov

MC21 is a continuous-energy Monte Carlo radiation transport code for the calculation of the steady-state spatial distributions of reaction rates in three-dimensional models. The code supports neutron and photon transport in fixed source problems, as well as iterated-fission-source (eigenvalue) neutron transport problems. MC21 has been designed and optimized to support large-scale problems in reactor physics, shielding, and criticality analysis applications. The code also supports many in-line reactor feedback effects, including depletion, thermal feedback, xenon feedback, eigenvalue search, and neutron and photon heating. MC21 uses continuous-energy neutron/nucleus interaction physics over the range from 10^{-5} eV to 20 MeV. The code treats all common neutron scattering mechanisms, including fast-range elastic and non-elastic scattering, and thermal- and epithermal-range scattering from molecules and crystalline materials. For photon transport, MC21 uses continuous-energy interaction physics over the energy range from 1 keV to 100 GeV. The code treats all common photon interaction mechanisms, including Compton scattering, pair production, and photoelectric interactions. All of the nuclear data required by MC21 is provided by the NDEX system of codes, which extracts and processes data from EPDL-, ENDF-, and ACE-formatted source files. For geometry representation, MC21 employs a flexible constructive solid geometry system that allows users to create spatial cells from first- and second-order surfaces. The system also allows models to be built up as hierarchical collections of previously defined spatial cells, with interior detail provided by grids and template overlays. Results are collected by a generalized tally capability which allows users to edit integral flux and reaction rate information. Results can be collected over the entire problem or within specific regions of interest through the use of phase filters that control which particles are allowed to score each tally. The tally system has been optimized to maintain a high level of efficiency, even as the number of edit regions becomes very large.

KEYWORDS: MC21, Monte Carlo, reactor calculations

I. Introduction

MC21 is a Monte Carlo neutron and photon transport code under joint development by Bechtel Marine Propulsion Corporation at the Bettis Atomic Power Laboratory and the Knolls Atomic Power Laboratory. MC21 is the Monte Carlo transport kernel of a system of codes that provides an automated, computer-aided modeling and post-processing environment. MC21 is designed with reactor analysis calculations in mind, and is intended to push the Monte Carlo method beyond its traditional role as a benchmarking tool or “tool of last resort” into a mainstream design tool. This paper provides an overview of the code, emphasizing features that are new or have changed since the previous overview paper⁽¹⁾ was published in 2007.

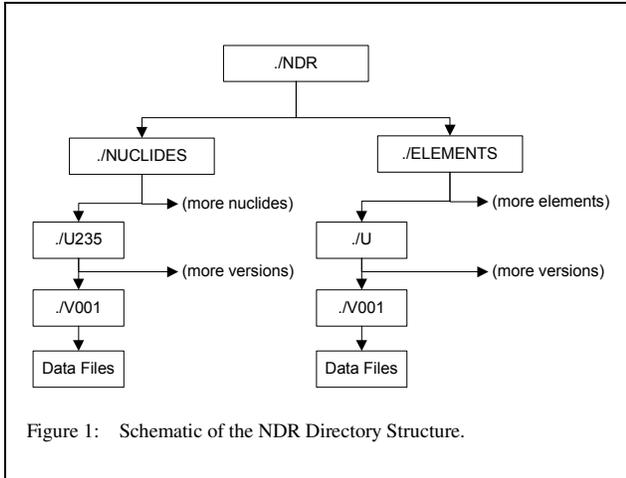
II. Nuclear and Photoatomic Data Processing System

The NDEX nuclear data system⁽¹⁾ supplies the data for neutron and photon transport calculations in MC21. The system is designed to extract data from a Nuclear Data Repository (NDR), which contains ENDF-formatted⁽²⁾ source data as well as NJOY⁽³⁾ output files. The data are

arranged in the NDR by nuclide names (for neutron interaction data) or element names (for photon interaction data). Each nuclide or element contains one or more versions. The different versions are used to account for different source data, i.e., ENDF/B, JEFF, JENDL, etc., or other factors that make the data unique. The data are arranged in a simple, UNIX file structure as shown in Fig. 1. NDEX assembles these data into either nuclear data (ND_LIBRARY) libraries or photoatomic data libraries (PD_LIBRARY) suitable for use in MC21 calculations.

II.1. Neutron-Nucleus Data

NJOY is used to populate the NDR with data required by NDEX. A typical NJOY job consists of running RECONR, BROADR, HEATR, and ACER. The output files from each module are stored in the NDR. Data are Doppler broadened to a reference temperature by NJOY (typically 293.6 K) and are later broadened to the requested temperature. Native routines in NDEX process data directly from the ENDF-formatted file to create probability tables in the unresolved resonance range, thermal scattering



data for moderators, fission spectrum information, and photon production data. NDEX will process nuclear data for all reactions with thresholds below 20.0 MeV; data for reactions with higher thresholds are not processed or saved.

Users generate ND_LIBRARY files for use in MC21 by executing NDEX with an associated input file that lists the requested nuclides, version numbers, 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 ND_LIBRARY file. Following each run, NDEX generates an output file summarizing the run and providing any warning and error messages. Extensive checks are performed of the data to ensure positivity of cross sections and energy distribution functions. If problems with the data are detected, NDEX will either correct the data and provide a warning message, or abort with an appropriate error message. Optionally, NDEX also generates files suitable for plotting cross sections.

II.2. Photoatomic Data

The NDEX/NDR system also processes photoatomic data to support photon tracking in MC21. NDEX extracts and processes photoatomic data required for MC21 primarily from the Lawrence Livermore National Laboratory (LLNL) Evaluated Photon Data Library (EPDL).⁽⁴⁾

NDEX processes and arranges the data in the ACE file to meet the specific requirements of MC21 and creates a PD_LIBRARY file containing basic photoatomic data for all elements, including: interaction cross sections, incoherent and coherent scattering form factors, physical properties of the atom for each electron orbital, binding energy, and Compton profiles. The PD_LIBRARY also includes data used for the production of bremsstrahlung photons, such as the scaled energy spectrum used for calculating the cross section for bremsstrahlung emission,⁽⁵⁾ 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.⁽⁶⁾

III. Interaction Physics

III.1. Neutron Interaction Physics

MC21 uses continuous-energy neutron/nucleus interaction physics over the energy range between 10^{-5} eV and 20 MeV. Each nuclide employs its own energy mesh on which the cross sections for all reactions for that nuclide are tabulated. The energy dependence of the cross sections between the tabulated points is assumed to be linear, and the mesh spacing is made sufficiently fine that linear interpolation leads to negligible error.

The unresolved resonance energy range (URR) is treated using a variation on the probability table method.⁽⁷⁾ In this method, the range of possible total cross section values is divided into a number of bands, and associated with each band is a probability giving the likelihood that the total cross section lies within the band. Resonance factors (ratios of band-average to dilute-average cross sections) for the total and various partial cross sections are also tabulated for each band. Whenever a neutron enters the URR of a nuclide, a band is sampled using the band probabilities for that nuclide and the corresponding resonance factors are applied to the continuous-energy infinite-dilute-average cross section values. These modified cross sections are then used for that neutron until it undergoes its next scattering event. The probability tables are statistically interpolated between energy mesh points.

For the purpose of scattering kinematics, nuclides are classified as either moderators or non-moderators. Those nuclides for which tabulated thermal-range scattering data are available in ENDF⁽²⁾ or similar nuclear data files are considered moderators, while all other nuclides are considered non-moderators. For both classes of nuclides the secondary energy and the scattering angle are sampled from distributions obtained from the nuclear data files.

For each moderator nuclide, an energy cutoff is specified which defines the top of the thermal energy range. Within this range mathematical models and corresponding tabulated parameters are used to determine the outcome of scattering events. The possible thermal-range scattering mechanisms are incoherent inelastic, incoherent elastic, and coherent elastic. A single moderator nuclide may employ one or more of these mechanisms. For incoherent inelastic scattering an improved version of Ballinger's method⁽⁸⁾ is used. For epithermal range scattering from moderators a version of the short-collision-time approximation that preserves detailed balance is employed.⁽⁹⁾ The epithermal range is considered to extend from the top of the thermal range to 400 kT, except for the case of ¹H bound in a molecule where it extends to 20 MeV. In the fast energy range the target nucleus is treated as if it were at rest in the laboratory reference frame.

For non-moderators, the target-at-rest model is used as for moderators for neutron energies above 400 kT. For lower neutron energies, the target nucleus is assumed to have a Maxwellian velocity distribution at the material temperature.

III.2. Photon Interaction Physics

For photons, MC21 uses continuous-energy photo-atomic interaction physics over the energy range from 1 keV to 100 GeV. Each element employs its own unique energy mesh on which the cross sections for all reactions for that element are tabulated. The energy dependence of the cross sections between the tabulated points is assumed to be logarithmic, except for cases where it is not mathematically possible and linear dependence is assumed. MC21 uses continuous energy transport of photons, and treats all common photon interaction mechanisms, including Compton scattering, pair production, and photoelectric interactions.

Inelastic scattering of a photon by a bound electron, referred to as incoherent scattering, is treated explicitly. Electron binding effects are approximated through the use of incoherent scattering functions, which modify the scattered photon energy, angular distribution, and total cross section of Compton (free electron) scattering, as given by the Klein-Nishina relationship.⁽¹⁰⁾ Effects of photon polarization and the pre-collision motion of electrons are neglected. Because MC21 does not account for electron transport, the kinetic energy of freed electrons is always deposited locally.

Photoelectric absorption is treated as a termination event. Creation of secondary electrons and fluorescence photons are neglected, and the energy of these secondary radiations is always deposited locally.

Pair-production reactions are treated as a scattering event, where the incident photon is replaced by a 0.511 MeV photon with twice the tally weight of the original photon. The direction of the post-reaction photon is sampled from an isotropic distribution. During pair-production, MC21 neglects transport effects for both the electron and positron and assumes that energy lost during deceleration of the charged particles is deposited locally.

Production of secondary bremsstrahlung photons due to charged particle deceleration is treated with a thick target approximation, which does not require explicit electron transport. In the thick-target approximation the electron mean range is subdivided into a series of sub-regions and the number of bremsstrahlung photons created in each sub-region are computed using the continuous slowing down approximation.⁽¹¹⁾ The energy and direction of the created photons are sampled at the end of each sub-region.^(5, 12,13)

IV. Geometry

MC21 includes a flexible three-dimensional (3-D) combinatorial geometry capability coupled with dedicated geometry kernels for common shapes.^(14,15) This combination allows MC21 to model complex three-dimensional systems and also ensures that the specialized geometries commonly encountered in reactor analysis models have a compact representation that supports efficient particle tracking. The four main geometric constructs of MC21 include surfaces, components, grids, and overlays. Examples of typical

commercial reactor geometries modeled with MC21 are illustrated in Figures 2, 3, and 7. An example of an MC21 model for a more complex reactor geometry, the Advanced Test Reactor (ATR), is shown in Figure 4.

Surfaces are the building blocks of components. Presently, MC21 supports surfaces up to second order in all three spatial dimensions (general quadratic surfaces). Reduced forms of the general quadratic equation (e.g., plane, cylinder, sphere, etc.) are available to reduce storage and take advantage of increased tracking efficiency relative to a general quadratic representation. Boundary conditions are assigned to surfaces, and MC21 supports reflective, periodic (rotational and translational), escape and normal boundary conditions.

Components are volumes defined by union and/or intersection sets of surface half-spaces. In addition to their half-space definitions, components can be further specified hierarchically with respect to other components. Through this hierarchical definition, components can be constrained to exist only within another parent component and may contain other child components.

Grids provide additional internal geometric detail to a component. Every component always contains a grid with at least one grid cell, and the component volume must be completely filled by the grid. Component boundaries and the boundaries of the contained grid do not have to be coincident—any portion of the grid that is outside of the component is effectively truncated. The three available grid types are: spherical (r), cylindrical (r,z) and Cartesian (x,y,z). The Cartesian grid is a non-uniform mesh consisting of lines parallel to the x -, y -, and z -axes. A generalization to the standard Cartesian grid allows an arbitrary angle between the x - and y -axes.

Within a grid cell, an additional level of geometric detail can be added by applying an overlay.⁽¹⁵⁾ An overlay is a collection of one or more simple volumetric objects that can be superimposed within the volume of a grid cell. Overlays are based on geometric templates, which provide a way to efficiently model and track over specialized geometries. Each template contains its own unique geometry kernel. Currently, MC21 supports 2-D lattice, 3-D lattice, and box template types. These templates are used to represent repeated groups of nested or disjoint objects, including extruded 2-D cylinders (2-D lattice), spheres (3-D lattice), and parallelepipeds (box). Once defined, template definitions can be reused throughout a model, which simplifies the creation of repeated structures such as lattices of fuel pins. In this case, multiple overlays can be created from a single template and each copy can be uniquely positioned and assigned properties.

The distinct objects that result from combining these geometric constructs into an actual model are called cells. Every cell has a set of properties, which are defined as a part of the model description. Available cell properties include

the material identifier, composition identifier, region identifier, overlay identifier, temperature region identifier, fluence region identifier, temperature, density factor, importance, and volume.

MC21 also supports models with movable geometry by allowing components to translate along or rotate around a user-defined vector.⁽¹⁶⁾ Movable groups can be constructed which contain one or more component motion definitions. A hierarchical group structure can be created where a group, known as a “supergroup,” contains one or more previously defined groups, known as “subgroups.” There is no limit on the number of levels which can be contained in the hierarchy. Once movable groups have been defined in a model they can be manually moved (individually or collectively) during the simulation through control input. The movable geometry capability is also used by the eigenvalue search capability described in Section VII.4.

In addition, MC21 allows user-defined “configurations” that identify the specific location of one or more movable groups within a model. Configurations are referenced through user-defined labels (e.g., “all_rods_in,” “all_rods_out,” etc.) and can be predefined in the problem geometry or saved “on-the-fly” during program execution. Once a configuration is defined it is possible to return all groups to their positions in that configuration through use of a single command.

V. Tallies

MC21 Tallies are based on the familiar collision and track-length estimators⁽¹⁷⁾ used in other Monte Carlo particle transport codes. These types of estimators are especially well-suited for calculating integral quantities of the form

$$X = \int d\mathbf{r} \int d\Omega \int dE f(\mathbf{r}, \Omega, E) \psi(\mathbf{r}, \Omega, E), \quad (1)$$

where f is an arbitrary scoring function and ψ is the angular neutron flux as a function of position, direction, and energy. By selecting an appropriate scoring function f and adjusting the limits of integration, Eq. (1) can produce many important quantities for nuclear design and analysis.

The tally system takes advantage of the generality of Eq. (1) to provide users with the maximum flexibility when defining tally edits. Individual edits are configured by defining two parameters: 1) a filter parameter that determines which particles are allowed to score to the edit, and 2) a score parameter that specifies the appropriate scoring function for the edit. The first parameter effectively controls the limits of integration for Eq. (1). This parameter is set by applying a series of one or more phase filters to the tally. The phase filters restrict which particles are allowed to score to a particular edit bin. The code allows both volume and surface type edits. For volume edits, MC21 allows filtering of particles based on cell properties such as spatial region, material, birth region, thermal region, and fluence region. By selecting appropriate combinations of phase filters, MC21 allows users to define edits over very specific phase

volumes. Advanced filters, such as birth region and post-collision energy, allow users to quickly and easily set up complex tallies to estimate fission and energy transfer matrices.

For surface edits, MC21 includes surface and angular direction filters, as well as a specially designed leakage filter. The leakage tally allows users to filter particles based on their phase state both before and after a surface crossing event. Leakage tallies are defined by specifying a “leaving” and “entering” condition for the particle. For example, a simple leakage filter might allow all particles leaving material 2 and entering material 3 to score. More complex tallies may involve groups of material or region identifiers as well as filters based on the type of surface boundary condition (e.g., reflecting, periodic, etc.) encountered by the particle. This tally format allows a convenient way to define many different types of leakage edits without requiring the specification of individual bounding surfaces.

In addition to the standard volume and surface filters, MC21 offers several capabilities for editing over volumes that are independent from the underlying problem geometry. Generalized spatial tallies allow edit volumes to be defined by intersections of quadratic surfaces in the global coordinate frame, similar to the way that components are defined. MC21 uses traditional collision estimators to collect edits within these user-defined edit regions. These spatial edit regions are not considered part of the model geometry and are not included during particle tracking. As a result, spatial edit regions can overlap cell boundaries in the true model geometry, allowing collection of edit results over an arbitrary region of space.

MC21 also supports mesh overlays for tallying spatially-varying results over a contiguous region of the model. MC21 presently supports Cartesian and bent-line mesh overlays for tallying. Both types of mesh are defined by a 3-D structured arrangement of mesh bins. In Cartesian mesh the bin edges are aligned with the x , y , and z axes of the global coordinate system, while the bent-line mesh supports non-orthogonal x - y bin edges on the interior of the mesh. As with spatial edit regions, the mesh overlays are not considered a part of the model geometry and may overlap with tracking surfaces in the underlying model. However, MC21 does compute distances traveled through each mesh bin during particle tracking, which enables the use of track length estimators for mesh tallies. Figures 2, 4, and 7 provide examples of MC21 mesh tally results for several different reactor cores.

The second tally parameter, the scoring function f , determines the scoring contribution from each particle at every event—such as a collision or surface crossing. By selecting different scoring functions, different physical quantities can be estimated. Scoring functions in MC21 are divided into two categories: ‘macroedits’ and ‘detailed’ edits. Macroedits allow users to quickly request common reaction types such as average flux, total reaction rate, average mean

free path, and total absorption, fission, and production rates in the current material. Detailed edits give the user more flexibility by allowing them to specify individual nuclides and reaction types to score. In general, MC21 allows users to request any nuclide in the problem and any ENDF reaction type. The code also supports grouping results from different nuclides or reaction types into a single edit bin. In addition, MC21 supports the application of user-defined response functions as a part of the tally scoring process. These response functions can be used to score quantities such as detector current or absorbed dose.

Unlike integral tallies, which collect aggregate data over all particle histories, census tallies in MC21 allow users to save the state of individual particle histories that are undergoing a pre-defined type of event. This state information, which can include the particle position, energy, direction, and/or weight, is written to an HDF5 formatted census data file, which can be reused as a radiation source in subsequent MC21 calculations. The term census tally reflects the fact that the tally is collecting (and enumerating) individual members of the particle population. By collecting state information for particle histories undergoing a particular type of event (e.g., surface crossing), the census tally effectively generates a random sample of realizations from the true distribution for the event type. The resulting census data can then be treated as a randomly-sampled source of radiation particles in subsequent MC21 calculations. For example, a census source can be used as an initial source “guess” for eigenvalue calculations to reduce the number of necessary inactive batches, as well as for coupled MC21/MC21 splice calculations.

In addition to the user-defined tallies, MC21 offers several predefined tally sets and global edits. The predefined edits are specially designed to facilitate interaction with other MC21 feedback modules—such as the depletion and thermal-hydraulics modules. During a feedback calculation MC21 automatically defines internal tallies for the required reaction rates. The feedback modules then access and modify the reaction rates and other required data directly in memory, eliminating the need to write intermediate files to exchange data between modules.

For criticality calculations, MC21 always produces a small number of global edits that characterize the overall behavior of the problem. These global edits include the eigenvalue, the total particle and energy leakage from the problem space, the effective delayed neutron fraction (β_{eff}), edits that characterize the energy spectrum of the system (e.g., above thermal fission fraction, average energy of neutron causing fission, etc.), and a complete breakdown of the energy release and energy deposition rates.⁽¹⁸⁾ For eigenvalue estimates, three separate statistical estimators are used: collision, track-length, and absorption. During the simulation, batch-wise values of the collision estimator are sent to the output stream so that the user can monitor the progress of the simulation. At the end of the job, all three eigenvalue estimates are printed, along with minimum variance

combinations of the estimators.⁽¹⁹⁾ The final eigenvalue estimate provided by MC21 is the minimum variance combination of all three eigenvalue estimates.

In addition to flexibility, the MC21 tally system was also designed with performance in mind. The tally system has been specially optimized to maintain a high level of efficiency, even as the number of edit regions becomes very large. In general, Monte Carlo simulations containing many tallies (> 1000) can spend a significant amount of total run time searching for and scoring individual edit bins. This is a particular concern for large detailed depletion simulations, which may require hundreds of millions of individual edits. In problems containing many edits, the Monte Carlo code must determine which edit bins to score for every particle collision or surface crossing event. As the number of edits increases, this search process slows down accordingly. In MC21, the tally system uses a series of hashed arrays to increase the efficiency of the edit bin search. In tests this hashed search algorithm has proven to be efficient and robust, allowing simulations with over 300 million edits to be run routinely.

VI. Sources

MC21 offers a flexible input system for defining radiation source parameters (position, energy, and direction). Each of the parameters can either be sampled from a distribution or drawn from a census, which is a set of realizations stored in an HDF5 data file. MC21 supports both built-in (pre-defined) and user-defined distributions for each source parameter. The available built-in distributions include point, sphere, cylinder, and parallelepiped for position, mono-energetic, Watt fission spectrum, and Maxwellian for energy, and mono-directional, isotropic, and biased towards a fixed direction vector for direction. User-defined distributions offer more flexibility, but must be manually defined in a separate source input file. User-defined distributions may be input as a discrete probability density function, a piece-wise linear probability density function, or a histogram.

MC21 supports both simple and compound source definitions. A simple source definition has one distribution assigned to each of the three source parameters, and is usually defined in the control input file. A compound source definition is a combination of multiple simple source definitions that are weighted by the probability of sampling each simple source. Due to its complexity, a compound source is defined in a separate source input file. MC21 also allows for the sampling of these simple sources to be biased by providing an additional bias weight to each simple source. If the biased weights are provided, MC21 will sample a source definition from the biased probabilities and adjust the starting source weight by the ratio of the actual to the biased source probability.

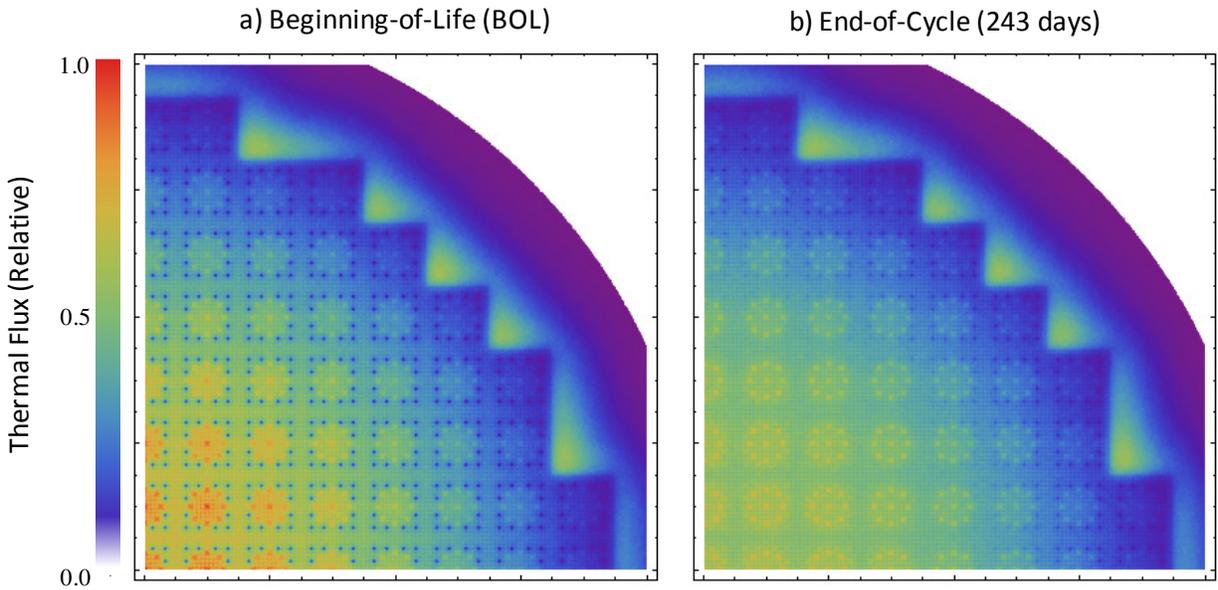


Figure 2: Quarter-core thermal flux distributions at core mid-plane for the HB Robinson commercial PWR⁽²²⁾ at (a) beginning-of-life and (b) at the end of the first cycle (5834 full-power hours). Constant-flux depletion analysis was performed with the in-line MC21 depletion solver, using 200,000 depletable compositions. Depletion sequence included 7 timesteps, with a simulation running strategy of 120 batches of 150,000 neutrons, with the first 20 batches discarded for source convergence for each timestep. Depletion power history and initial loading information was taken from Reference 22. Flux results were collected using track-length estimators on a 800×800×1 mesh tally. Depletion study ran in 1.93 hours on 192 cores (24 Intel Xeon Sandy Bridge 2.6GHz processors, 8 threads per processor).

MC21 also has the ability to filter source particles by birth location. Source particles may be accepted/rejected based on material type (e.g., fissionable), material number, component or region number. When source filtering is used, MC21 checks the state of each sampled particle against the applied source filters and rejects (and resamples) any source particles that do not meet the filter criteria. For example, if the source filter is set to restrict the source to fissionable materials, then only particles born in a fissionable material are accepted. Rejected particles are then resampled so that the total number of source particles is preserved.

VII. In-Line Feedback Effects

Perhaps the most distinguishing feature of MC21 is the integrated support for feedback effects relevant to reactor analysis applications. Presently, MC21 accounts for feedback due to depletion, thermal/hydraulic effects, xenon, photon heating, and control device motion (search to critical). In addition, MC21 provides the capability for executing external programs from within a simulation. This capability, along with the ability to import and export model conditions (e.g., temperature, density, etc.) provides nearly unlimited flexibility for coupling MC21 with external codes for support of additional feedback effects.

Each feedback effect in MC21 is driven by a separate solver kernel, each of which is tightly integrated into the overall code system. Communications between solver kernels and the main transport kernel is performed through a set of

well-defined application program interfaces (API). As a result, MC21 is able to exchange most data via local memory and avoid inefficient file passing techniques.

The feedback kernels account for the non-linear reactivity effects due to each of the above mentioned processes under (pseudo) steady-state conditions. This section provides a description of each of the feedback solver kernels individually. The next section will describe how the kernels work together during a simulation with feedback.

VII.1. Depletion

Depletion feedback capability in MC21 is handled by direct coupling with a dedicated depletion solver based on the Variable-coefficient Ordinary Differential Equation (VODE) solver package⁽²⁰⁾. MC21 supports both constant-flux and constant-power modes of depletion.⁽²¹⁾ Parallelism has been achieved by having multiple depletion code processes running simultaneously, one for each depletable composition. Example quarter-core thermal flux distributions from a one-cycle depletion analysis of the HB Robinson commercial PWR⁽²²⁾ are shown in Figure 2.

In constant-flux mode, MC21 solves the depletion equations assuming the reaction rates for each nuclide are held constant during the depletion step. The reaction rates are obtained from beginning-of-time step (BOT) transport calculations. In constant-power mode, a predictor-corrector methodology is used. During the predictor phase, constant-flux depletion is performed, followed by an

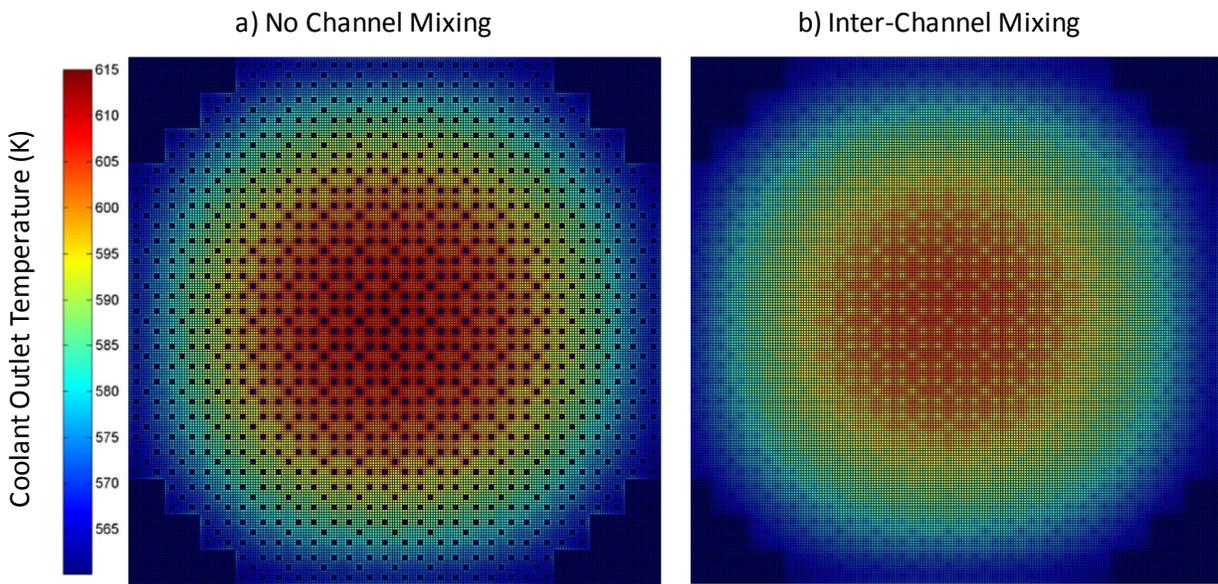


Figure 3: Full-core coolant outlet temperature for the Calvert Cliffs commercial PWR (a) without and (b) with coolant mixing between adjacent channels⁽²⁴⁾. Both temperature distributions were calculated with the in-line thermal feedback capability in MC21. Simulation included 11 thermal feedback iterations, each with a simulation running strategy of 220 batches of 500,000 neutrons, with the first 20 batches discarded for source convergence. Temperature results were collected for each of the 39,396 coolant channels. Total simulation time was 19 hours on 64 Intel Xeon Nehalem 2.4GHz processors.

end-of-time step (EOT) transport calculation. The changes between BOT and EOT nuclide reaction rates are approximated as linear in time, which allows slopes in the reaction rates to be computed. These slopes are then used in a corrector depletion wherein the coefficient matrix is incrementally changed during the multi-step solution of the VODE depletion solver. The continual update of the coefficient matrix simulates the changing of the reaction rates that would actually occur during a timestep. If desired, the corrector process may be repeated multiple times to improve convergence of the EOT number densities. However, experience to date has indicated that one corrector step is sufficient in most cases.

VII.2. Thermal Feedback

MC21 supports in-line steady-state thermal feedback calculations through the use of an integrated thermal model.⁽²³⁾ The thermal feedback model is based on a user-input description of heat transfer and coolant flow paths in the problem geometry.

In thermal feedback calculations each distinct spatial cell in the model is identified as a source, sink, or non-thermal region. Sources are the heat-producing regions in the core, sink regions contain coolant that removes heat from the core, and non-thermal regions are assumed to neither produce nor remove heat. Once each spatial cell is assigned a thermal region, the heat transfer mapping is applied, which describes the flow of heat from source to sink and flow of coolant from sink to sink. The code provides the flexibility to homogenize source and sink regions over arbitrary volumes of the core, which gives users the ability to control the level of spatial

detail used for the thermal feedback calculation. High spatial fidelity increases the memory requirements of the calculation and may also lead to increased uncertainty in local power edits due to the smaller volume, and hence reduced number of scores, in each source region. A look at the effects of spatial homogenization on MC21 thermal feedback is provided in Reference 24. Example results from this reference illustrating the capabilities of the in-line thermal feedback capability in MC21 for calculating outlet coolant temperatures at the top of the Calvert Cliffs commercial PWR are shown in Figure 3.

The thermal feedback module in MC21 tallies the power distribution by source region during the Monte Carlo transport simulation. The outputs of the thermal feedback module are new estimates of the temperature distribution in the source and sink regions and updated sink density values. The thermal feedback model in MC21 is based on the assumption of steady-state conservation of energy and mass under constant pressure. The only coolant currently supported by the thermal feedback model is water in the single-phase fluid, subcooled, and saturated fluid-vapor mixture regimes. To close the system of equations being solved, empirical correlations are used to determine the heat transfer coefficient and void fraction in each of the supported flow regimes. The conduction of heat through the fuel element is approximated by solving a one-dimensional conduction equation along a user-defined length (with input thermal conductivity) and the temperature rise across any non-thermal regions can be approximated through a user-input thermal resistivity value. Estimates of the critical heat flux at each heat transfer interface can be optionally enabled in the thermal feedback model.

The thermal feedback mechanisms accounted for by the thermal feedback model are the density change of water due to a change in temperature and the change in nuclear data due to the change in temperature. To properly account for the change in nuclear data (cross sections and thermal scattering data) it is necessary to use a nuclear data library containing evaluations at multiple temperatures. The nearest available temperature or interpolation can then be used to choose appropriate data from the library based on the results of the thermal feedback calculation. Due to the nonlinear nature of the thermal feedback process it is usually necessary to perform several iterations between the thermal feedback module and the Monte Carlo transport simulation before convergence is observed. MC21 does not automate this iterative process in any way; these iterations must be manually performed by the user.

VII.3. Xenon Feedback

MC21 supports in-line convergence to equilibrium xenon during criticality calculations as well as calculation of post-shutdown peak xenon conditions.⁽²⁵⁾ Convergence to the equilibrium xenon condition is achieved by periodically updating the xenon and iodine densities between neutron generations. During xenon feedback, MC21 collects tallies for regionwise xenon and iodine production rates along with the microscopic absorption rate for xenon in each region containing a depletable fissionable material. These reaction rates are used to compute the equilibrium xenon and iodine number densities in each region using a simple two nuclide xenon depletion chain.

Unlike the other MC21 tallies, xenon feedback tallies are not accumulated continuously over all neutron generations. Instead, a user-defined number of generations are grouped together into a xenon cycle, which is used as the basis for collecting xenon tallies. The reaction rates needed for the equilibrium xenon calculation can be tallied for each xenon cycle or accumulated over successive xenon cycles. The xenon and iodine number densities are updated at the end of each cycle. This process allows for the convergence of the xenon and iodine number densities to their equilibrium values, as well as the consistent convergence of the fission source distribution. In addition to the equilibrium xenon capability, MC21 can compute the maximum xenon mass in the core and the time at which it occurs following shutdown.

VII.4. Eigenvalue Search

MC21 includes the capability to search to a user-specified eigenvalue, within a specified confidence interval, by manipulation of the movable geometry.⁽¹⁶⁾ Eigenvalue searches can be performed in two different ways: through a movable search or through a search sequence. A movable search allows the user to define a single movable group to be moved during the search, along with the reference configuration for all search movement, the lower and upper bounds of the search space (with respect to the specified configuration), and, optionally, the starting point of the search and the reactivity worth of movement at the starting

point. During a movable search MC21 will move (translate or rotate) the search group through the range specified until the desired eigenvalue is found with the desired confidence interval or it is determined that there is no solution within the interval specified.

The second type of search, the search sequence, is a more complex search type which allows for a series of movable searches to be performed in a sequential manner until the desired eigenvalue is found with the desired confidence interval or the end of the sequence is reached without convergence. Sequences can be defined in a manner such that the specific series of movable searches to execute is dependent on the elapsed time in the simulation.

VII.5. Photon Heating

As a result of the need for different levels of accuracy in the energy deposition treatment, MC21 provides flexibility in how energy deposition for certain reactions is handled.⁽¹⁸⁾ By providing several energy deposition modes, each with different self-consistent (energy-preserving) approximations for neglecting the energy transport of certain secondary radiations, MC21 lets users choose between accuracy and computational efficiency.

During transport, MC21 allows energy transfer to take place at fission sites, neutron collision sites, and/or photon interaction sites, as appropriate. The energy deposition mode for the calculation determines both how energy release should be categorized and where (i.e., at what reactions) the energy will be deposited. Presently, MC21 offers 4 different energy deposition modes, which range from full transport of all neutral secondary radiations to a basic treatment that assumes that all energy is deposited locally.

For criticality calculations that allow energy transport via neutrons, MC21 preserves energy between generations by applying an energy rebalance factor, which is defined as the ratio of the total neutron energy released in fission during the previous generation to the total neutron energy in the normalized source bank for the current generation. This energy rebalance factor serves as a weighting factor for energy deposition and leakage tallies, which preserves total fission neutron energy across generations.

The most-detailed energy deposition mode includes explicit energy transport for both neutrons and photons. In this mode, MC21 first runs a neutron transport calculation and samples capture and fission photons for each neutron collision event. The sampled photon source sites are written to an HDF5 census data file. Following the neutron transport calculation, MC21 performs a photon transport calculation using the source data from the photon source data census file. During photon transport, MC21 computes the energy deposition for each cell in the problem using the local photon KERMA value. Following the calculation, the code normalizes energy deposition values by the total photon source energy released to compute the relative photon energy deposition fraction for every cell in the problem.

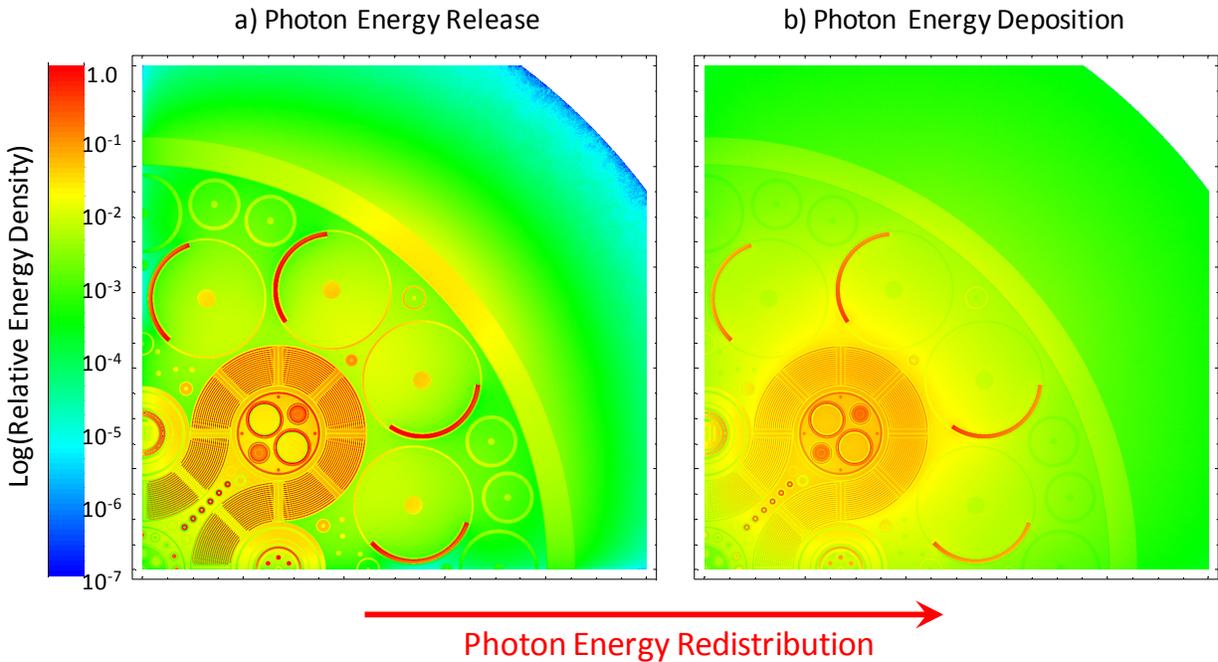


Figure 4: Quarter-core photon energy redistribution in the Advanced Test Reactor (ATR), showing relative spatial density of (a) photon energy release and (b) photon energy deposition. Results were generated with the in-line photon heating capability in MC21, with fully-coupled neutron/photon transport. Heating calculation included separate neutron and photon simulations, each with 5,000 active batches of 10,000 particles. Flux results were collected using track-length estimators on a $1000 \times 1000 \times 1$ mesh tally. Total simulation time was 3 hours on 64 Intel Xeon Nehalem 2.4GHz processors.

This cell-based energy deposition fraction data forms a discretized representation of the photon redistribution function, which describes how photons will redistribute energy throughout the problem via transport. Once calculated, the photon redistribution function is multiplied by the total photon energy released during the neutron calculation and the resulting product is summed over all cells to give an estimate of the total photon energy deposition. The photon redistribution function is saved so that the local photon energy deposition can be quickly recalculated based only on the total photon energy released in a subsequent neutron calculation. Thus, photon transport calculations only need to be run when changes in the problem (due to depletion, thermal feedback, etc.) have caused a change in the overall shape of the photon redistribution function. An example of the redistribution of photon energy due to radiative transport in the Advanced Test Reactor is shown in Figure 4.

VIII. Running a Simulation

This section provides an overview of the mechanics of setting up and running an MC21 simulation and the interactions among the various feedback modules.

VIII.1. Input/Output Formats

MC21 recognizes two different file formats: ASCII and the binary self-describing Hierarchical Data Format 5 (HDF5).⁽²⁶⁾ ASCII is a well known standard for encoding character data and can be processed by any text editor or command line

utility. The major MC21 input files that describe the problem and the simulation options are all ASCII text files. The ASCII character set has the advantage of being universally supported and human-readable, but can be inefficient for large data sets. As a result, the majority of MC21 output, aside from the standard output stream, is encoded in the binary HDF5 format.

VIII.2. Job Control

When MC21 is executed, the program immediately searches for a control input file, which contains a list of directives and user-defined options for the job. The MC21 control file is a text file which contains a user-defined sequence of control cards. Control cards are used to define all aspects of the MC21 job, including running strategy, physics options, and feedback effects. Control cards use a case-sensitive keyword based input system, with one card per line of the input file. Every card has the form:

keyword *options*

where *keyword* is the name of the control card, and *options* are a list of user-selected input values required for the card. An example control input file is shown in Figure 5.

All MC21 simulations include one or more timesteps, which can be used to model the behavior of the problem over some period of time. As a result, the control input file is divided into independent timestep input sections using the

```

job_type                k-eff
histories               5000
batches                 100
discard                 10
seed                   587
num_timestep            4
rated_power             100.0 MW

<TIMESTEP> 1
case_title              'Source Convergence Step'
source_coordinates      point 0.0 0.0 0.0
source_energy           mono 2.0e6
{EXECUTE} spatial

<TIMESTEP> 2
case_title              'Depletion: 1000h @ 10MW'
source_definition       last
timestep_length         1000.0 hours
depletion_power_fraction 0.1
{EXECUTE} depletion

<TIMESTEP> 3
case_title              'Depletion: 1000h @ 100MW'
histories               10000
depletion_power_fraction 1.0
{EXECUTE} depletion

<TIMESTEP> 4
case_title              'Shutdown Depletion: 3y'
depletion_power_fraction 0.0
timestep_length         3 years
{EXECUTE} depletion

```

Figure 5: Example MC21 control input file for a multiple timestep depletion calculation.

<TIMESTEP> control card. The timestep cards in the control input file serve as case separators, dividing the control cards based on the timestep that they apply to.

Within each timestep, execution cards (`{EXECUTE}`) are used to instruct MC21 to perform a given type of calculation or simulation. When an execution card is encountered during the processing of control cards within a timestep, MC21 suspends reading the control input file and runs the requested calculation(s). Upon completion, MC21 resumes processing any additional control cards for the timestep. This process continues until all control and execution cards for the timestep have been processed.

Each execution card defines a single feedback iteration for the timestep. The use of multiple execution cards within a timestep allows users to control the number (and sequence) of iterations used to converge feedback effects (thermal, equilibrium xenon, depletion, etc.) for the timestep. Additional control cards may be included between execution cards in order to change MC21 options between iterations. For example, a user may wish to change the number of neutron histories or batches between successive iterations as a part of their desired convergence strategy. Most MC21 feedback iterations require at least one Monte Carlo transport simulation per call. However, there are a few execution options (e.g., setting maximum xenon condition) that do not require a transport simulation. MC21 also allows multiple execution options to be listed on a single execution card. In this case MC21 will attempt to run all of the

requested calculations simultaneously. In many cases, MC21 can collect information for multiple calculations (e.g., thermal, equilibrium xenon, depletion, etc.) during a single transport simulation, so it is often advantageous to combine execution options on a single card.

VIII.3. Feedback Sequence and Convergence

There is an important relationship between timesteps and feedback iterations in MC21. A timestep in MC21 refers to a finite period of calendar time, defined by the user-input timestep length. In contrast, feedback iterations involve static calculations performed at a particular instant in time. As a general rule, calendar time is not advanced during a feedback iteration. The only exception to this rule involves feedback iterations that include depletion calculations, as described below. Because calendar time is not advanced during a feedback iteration, it is possible to use multiple iterations to converge a solution (including all feedback effects) at any particular instant in time. For example, in a problem including both temperature and equilibrium xenon/samarium feedback effects, it may be necessary to run several iterations to ensure that the temperature distribution and equilibrium xenon/samarium concentrations have reached a converged solution. The resulting (converged) temperature, xenon/samarium concentration, and neutron flux distributions are estimates for the steady-state solution at a particular point in time.

Because timesteps in MC21 correspond to a fixed period of calendar time, it is natural to talk about the beginning-of-timestep (BOT) and end-of-timestep (EOT) time points. Feedback iterations in MC21 always take place at either the BOT or EOT times. There is no way to perform a calculation or obtain results from MC21 for intermediary times within a timestep. For the first timestep in a beginning of life (non-restart) job the initial BOT time is set to 0.0 by default. For all subsequent timesteps, the beginning of timestep calendar time is set to the cumulative time elapsed during all previous timesteps (i.e., the sum of the timestep lengths from all previous timesteps). During a timestep, calendar time is advanced from the BOT time to the EOT time immediately after the first depletion feedback iteration in the timestep (the first execution card that includes a depletion mode keyword). This means that any feedback iterations prior to (and including) the first depletion calculation in a timestep will be performed at the BOT time. Any feedback iterations after the first depletion calculation in a timestep (including any subsequent depletion calculations) will be performed at the EOT time. This allows MC21 to provide converged solutions at both BOT and EOT time points for each timestep. MC21 does not require that the converged BOT solution for a timestep must match the converged EOT solution from the previous timestep. This capability is particularly important for cases where step changes to the reactor configuration or operating conditions are made between timesteps. For example, consider a problem where a step change to a control rod position is made at the BOT time of a particular timestep. Following this change, BOT feedback iterations can be used

to produce a converged solution that reflects the new control rod position. It follows that this converged BOT solution will not match the EOT solution from the previous timestep, due to the change in control rod position. Because the calendar time has not changed between the previous EOT and the current BOT, MC21 effectively treats the step change (and the response of the reactor to the change) as an instantaneous event. Finally, in timesteps with no depletion calculations, all feedback iterations are performed at the BOT time.

MC21 uses a predictor-corrector methodology to solve for the time-dependent feedback behavior of a problem over the length of a timestep. This methodology assumes that reaction rates throughout the problem vary linearly between BOT and EOT values, as described in Reference 21. As noted above, any changes to the problem made at BOT are assumed to occur instantaneously, and are reflected in the final BOT reaction rates. Once the first depletion calculation has been performed in a timestep the BOT reaction rates are set and cannot be affected by subsequent iterations. Any additional iterations are used to converge the EOT solutions, which, in turn, determines the slope of reaction rate changes over the timestep. As a result, any changes to the reactor configuration or operating conditions (either manual changes or natural changes) will be effectively averaged over the entire length of the timestep. For example, MC21 will treat an adjustment to the control rod position during the BOT feedback iterations (prior to or during the first depletion calculation for the timestep) as though it occurred instantaneously at the BOT time. However, MC21 will treat an adjustment to the control rod position during EOT feedback iterations (after the first depletion calculation for the timestep) as a continual withdrawal over the timestep length.

VIII.4. In-Line Branch Calculations

MC21 offers an in-line branch capability. After all feedback iterations for the main calculation have been performed, a %BRANCH% control card can be used to create one or more stand-alone branch calculations at the end of a timestep. Within a branch, any MC21 input options (with the exception of depletion) can be used to perform additional calculations at the EOT state. All branches begin at the converged EOT conditions, which were achieved immediately prior to the first branch control card within a timestep. Calculations performed as a part of a branch are completely independent from other branches as well as the main-line (trunk) calculation. In particular, branch calculations do not affect the results from the trunk calculation, for either the current or any subsequent timesteps. At the end of each branch the problem conditions are reset to the converged EOT state for the timestep. The branch capability allows users the ability to perform in-line sensitivity studies (e.g., xenon worth, temperature reactivity, etc.) at predetermined points in the calculation, without using the timestep restart capability offered by the code.

VIII.5. Radiation Transport Modes

MC21 supports two different types of radiation transport calculations: fixed- and iterated-fission-source. In fixed-source calculations the spatial distribution of source neutrons is provided as input, and remains unchanged throughout the calculation. In iterated-fission-source calculations an initial 'guess' of a source distribution is provided as input, and then the code converges on the true fission source distribution.

For fixed-source calculations the neutron histories are grouped into batches. At the beginning of a batch, the tally accumulators for the batch are set to zero. As neutrons score a particular tally, the corresponding accumulator is incremented by the appropriate amount. At the conclusion of the batch the batch accumulators are used to increment the corresponding mean and mean-squared accumulators for the calculation as a whole. At the end of the calculation, these latter accumulators are used to construct 95% confidence intervals using the usual formula for independent, normally-distributed random variables.

The reason that MC21 does not use the scores for individual histories for the statistical analysis is that in general these will not be normally distributed. While the central limit theorem states that—regardless of the distribution of the random variables—their sample mean will be normally distributed, it does not follow that the t -statistic used to compute the confidence interval will be distributed with a Student's t -distribution⁽²⁷⁾. For sufficiently large batches, however, the batch scores will approach a normal distribution (due to the central limit theorem) and the quantiles of the t -distribution can then be used to compute valid confidence intervals.

Like many similar Monte Carlo codes, MC21 uses the method of successive generations⁽²⁸⁾ to perform iterated-fission-source calculations. As has been shown previously,⁽²⁹⁾ this method causes successive fission generations to be correlated. The effect of this correlation must either be eliminated or accounted for to compute valid confidence intervals. MC21 uses a method due to Gelbard and Prael⁽³⁰⁾ that reduces the correlation to negligible levels so that the standard statistical procedures for uncorrelated random variables may be used. This method consists of dividing the fission generations into batches, each consisting of a number of consecutive generations. Since the correlation between generations falls off with the lag between generations, correlation between two consecutive batches is due primarily to the generations at the end of the first and the beginning of the second. By including a sufficiently large number of generations in a batch, the correlation due to these generations can be made negligibly small.⁽³¹⁾ Confidence intervals are then computed using the batch results as the random variables as for fixed-source calculations.

IX. Variance Reduction

MC21 uses implicit capture to reduce the statistical uncertainty of the results for a given amount of computational effort. In this method, each neutron is given an initial starting statistical ‘weight’ of unity. At each collision, this weight is multiplied by the non-absorption probability. The difference between the pre- and post-collision weights is attributed to absorption. Eventually, the weight may become so small that it is not worth the computational effort to continue tracking the neutron. This situation is handled using the Russian roulette technique, in which one of two possibilities is chosen randomly. One possibility is that the weight is set to zero and the neutron is no longer tracked. The other is that the weight is increased. The probabilities associated with these two possibilities, as well as the new weight should the second possibility be selected, are chosen to conserve weight on the average.

MC21 also has the capability of using weight-window variance reduction based on importance values that vary by geometric cell or by mesh cell. Mesh cell importances can vary in space or energy, whereas geometric cell importances can only vary in space. When a neutron changes energy or crosses a spatial boundary (geometric or mesh), it may be split, terminated by the Russian roulette process, or left unchanged, depending on the new importance value. To improve efficiency, MC21 does not split and roulette every time a particle moves to a mesh bin (or energy group) with a different importance value. Instead there exists a window of allowable particle weights around the desired weight value of each mesh cell and energy group. Any particle whose weight is inside of the weight window is not modified, and any particle whose weight is outside the weight-window undergoes splitting or rouletting, as appropriate, to bring the weight of the particle into the desired range. The size of the weight-window in an MC21 calculation is controlled by a pair of user-adjustable parameters. These parameters are global multipliers that define the window width for each cell or mesh bin/energy group as a fraction of the desired weight (i.e., inversely proportional to the importance) for the cell or mesh bin/energy group.

The mesh-based weight-window capability in MC21 has been designed to seamlessly work within the Consistent Adjoint Driven Importance Sampling (CADIS) and Forward-Weighted (FW-)CADIS frameworks.^(32,33) Both of these techniques provide a way to accelerate fixed-source Monte Carlo calculations using importance values taken from an adjoint solution taken from a deterministic transport (or diffusion) calculation. The CADIS and FW-CADIS methods make use of the fact that the adjoint flux solution provides a measure of the relative importance of a particle in any given region of phase space with respect to a given objective tally or set of tallies. Thus, the adjoint flux can be used to generate space- and energy-dependent importance values that are optimized for a particular problem. The CADIS method is used for calculating responses at single detectors and energy spectra, whereas the FW-CADIS

method is used for calculating global distributions, such as flux or dose rate distributions, as well as responses at multiple detectors and energy spectra.

Low-flux regions of a reactor core tend to produce reaction rate tallies with higher relative statistical uncertainties than high-flux regions due to the fewer number of scoring events in the former relative to the latter. In some circumstances a more-nearly uniform distribution of uncertainties is desirable. One way to achieve this is through the use of spatially-dependent importance functions, as described previously. While this can be a very successful approach, methodologies that implement it, such as FW-CADIS, require auxiliary deterministic transport calculations. MC21 has the option to use the Uniform Fission Site (UFS) method⁽³⁴⁾ to achieve a flattening of the relative uncertainty distribution without the need for auxiliary calculations. This method adjusts the starting scoring weights of the neutron histories in such a way as to flatten the distribution of fission source sites while preserving the correct fission source distribution.

Two additional features available in MC21 are the use of the Shannon entropy diagnostic⁽³⁵⁾ to help ascertain source convergence, and the ability to estimate the dominance ratio using the Noise Propagation Matrix method,⁽³⁶⁾ which is based on and shares many properties with the Course-Mesh Projection Method.⁽³⁷⁾

X. Performance and Scaling

From its initial conception, MC21 was designed for massively parallel computing environments. As of version 6.0, MC21 supports both message passing parallelism between compute nodes using MPI, as well as shared memory parallelism (threading) within nodes using OpenMP. In addition, MC21 uses the parallel I/O capability of HDF5 to read and write large data files in parallel. Testing has shown that the code has excellent scaling up to 10,000 processors, which is the largest number of processors available for testing the code. Scaling results for the latest version of the code up to 4096 processors are shown in Figure 6. The code is routinely run with hundreds to thousands of processors for production level design and analysis work.

Additional performance testing of MC21 has been conducted using several recently published PWR benchmark problems, including the MIT PWR (BEAVRS) benchmark⁽³⁸⁾ and the Nuclear Energy Agency (NEA) Monte Carlo performance benchmark problem⁽³⁹⁾. These benchmark problems were created to provide code developers and users a common model to use to assess the performance improvement of Monte Carlo codes as algorithms and computational platforms evolve.⁽³⁹⁾ The MC21 analyses of the BEAVRS and NEA benchmark problems are documented in References 40 and 31, respectively.

MC21 v.6.0 Parallel Performance (Weak Scaling)

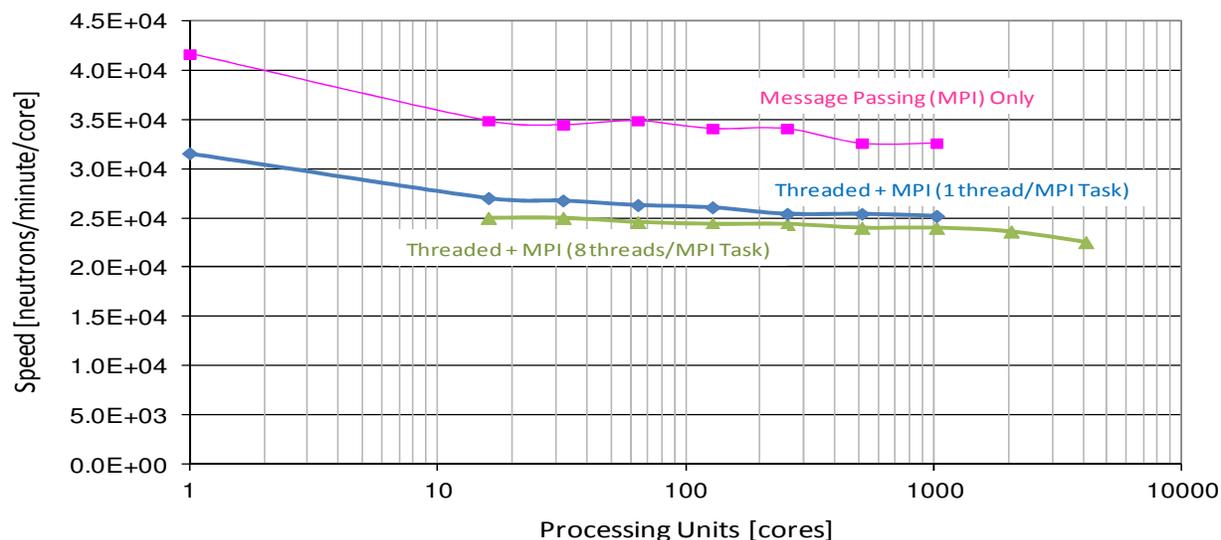


Figure 6: Weak parallel scaling performance of MC21 up to 4096 processing units (cores). Scaling results shown for the MIT BEAVRS PWR benchmark model,⁽³⁸⁾ which requires 3.94 Gb of memory per MPI task. Results shown for MC21 runs using message-passing (MPI) parallelism only and hybrid shared memory (threaded) and message passing parallelism for both 1 thread and 8 threads per MPI task. Computational work was kept constant at 5,000 histories per processing unit for all tests. All simulations were run on Intel Xeon Sandy Bridge 2.6GHz processors.

The MC21 representation of the MIT BEAVRS benchmark problem included 209 components, containing 40.896 million grid cells and 4.635 million grid cell overlays for a total of 45.532 million cells in the problem. Tally results for the problem were collected over 5.044 million mesh regions. Reference calculations reported in Reference 40 used 30,520 generations (250 discarded) of 4 million neutrons per generation for a total of 120 billion active histories. Calculations required 2.5 days on 1000 Intel Xeon processors (combination Westmere and Sandy Bridge), with an effective throughput of 33.8 million neutrons/minute. Example flux results for a 2-D quarter-core slice of the MIT BEAVRS benchmark are shown in Figure 7.

As reported in Reference 31, the MC21 representation of the NEA benchmark model contained approximately 7.3 million tally regions. Flux, absorption rate, and power were computed for each region, for a total of approximately 21.8 million tallies. The calculation used 3 discard and 250 active batches, each consisting of 200 fission generations. Each generation consisted of 4 million histories, for a total of 202.4 billion histories. The calculation took 75.6 hours on 750 Intel Xeon Westmere X5660 cores, resulting in a rate of 44.62 million neutrons/minute.

Acknowledgements

The authors wish to thank C. M. Rodenbush, B. R. Hanna, and S. J. Spychala from the Bettis Atomic Power Laboratory and D. J. Kelly and B. N. Aviles from the Knolls Atomic Power Laboratory for providing the reactor models shown in this paper.

References

- 1) T. M. Sutton, T. J. Donovan, T. H. Trumbull, P. S. Dobreff, E. Caro, D. P. Griesheimer, L. J. Tyburski, D. C. Carpenter and H. Joo, "The MC21 Monte Carlo Transport Code," *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in the Nuclear Industry*, (CD ROM) Monterey, California, April 15–19, 2007.
- 2) M. Herman and A. Trkov, "ENDF-6 Formats Manual," ENDF-102/BNL-90365-2009, June 2009.
- 3) R. E. MacFarlane, D. W. Muir, "The NJOY Nuclear Data Processing System, Version 91," Los Alamos National Laboratory, LA-12740-M (1994).
- 4) D. E. Cullen, "EPDL97: The Evaluated Photon Data Library, '97 Version," UCRL-50400, Volume 6, Rev. 5 (1997).
- 5) S. M. Seltzer and M. J. Berger, "Bremsstrahlung Spectra from Electron Interactions with Screened Atomic Nuclei and Orbital Electrons," *Nuclear Instruments and Methods in Physics Research*, **B12**, pp. 95–134 (1985).
- 6) R. M. Sternheimer, S. M. Seltzer and M. J. Berger, "Density Effect for the Ionization Loss of Charged Particles in Various Substances," *Atomic Data and Nuclear Data Tables* **30**, pp. 261-271 (1984).
- 7) L. B. Levitt, "The Probability Table Method for Treating Unresolved Neutron Resonances in Monte Carlo Calculations," *Nucl. Sci. Eng.*, **49**, pp. 450–457 (1972).
- 8) C. T. Ballinger, "The Direct $S(\alpha, \beta)$ Method for Thermal Neutron Scattering," *Proceedings of the International Conference on Mathematics and Computation, Reactor Physics, and Environmental Analysis*, Vol. 1, pp. 134-143, Portland, Oregon, April 30–May 4, 1995.
- 9) T. M. Sutton, T. H. Trumbull and C. R. Lubitz, "Comparison of Some Monte Carlo Models for Bound Hydrogen Scattering," *Proc. Int. Con. Math., Comp. Meth. And Reactor Phys.*, (on CD ROM) Saratoga Springs, New York, May 3–7, 2009.

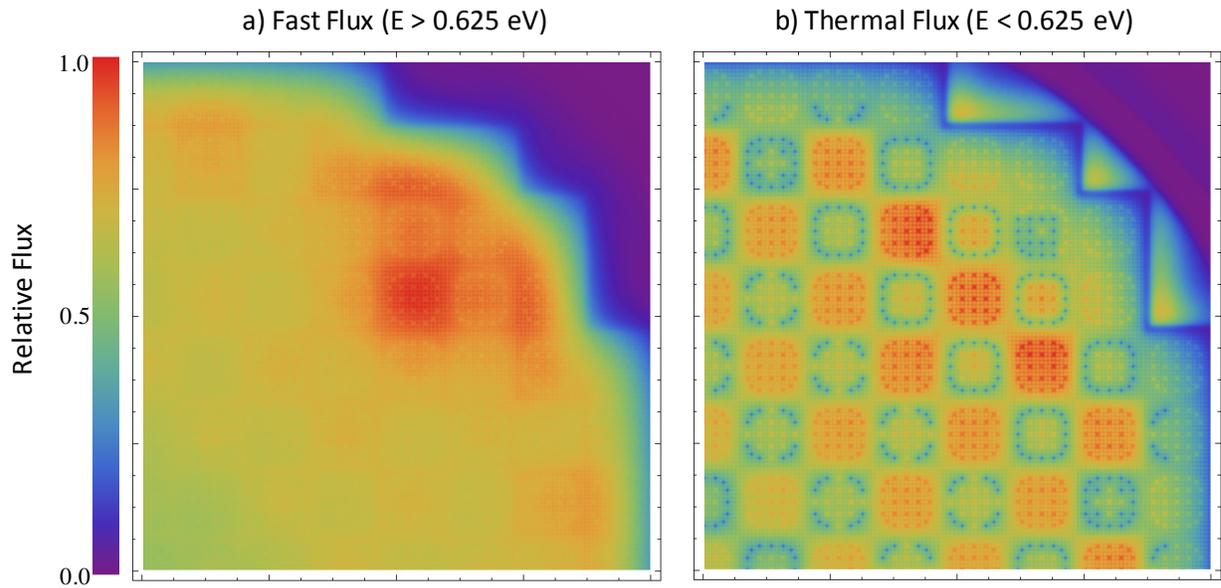


Figure 7: Quarter-core (a) fast (> 0.625 eV) and (b) thermal flux distributions at core mid-plane ($z = 200$ cm) for the MIT PWR (BEAVRS) benchmark⁽³⁸⁾. Results were collected using track-length estimators on a $800 \times 800 \times 1$ mesh tally with a simulation running strategy of 120 batches of 1 million neutrons, with the first 20 batches discarded for source convergence. Simulation ran in 5.41 minutes on 128 Intel Xeon Sandy Bridge 2.6GHz processors.

- 10) W. Heitler, *The Quantum Theory of Radiation*, Oxford University Press, London, 1954.
- 11) M. J. Berger, "Monte Carlo Calculation of the Penetration and Diffusion of Fast Charged Particles," *Methods in Computational Physics*, 1, edited by B. Alder, S. Fernbach and M. Rotenberg, Academic Press, New York, 1963, pp. 135–215.
- 12) W. C. Dickinson and E.M. Lent, "Calculation of Forward Bremsstrahlung Spectra from Thick Targets," UCRL-50442, January 1968.
- 13) W. W. Scott, "Angular Distribution of Thick-Target Bremsstrahlung which Includes Multiple Electron Scatterings," NASA Technical Note TN D-4063, August 1967.
- 14) T. J. Donovan, L. J. Tyburski, "Geometric Representations in the Developmental Monte Carlo Transport Code MC21," *American Nuclear Society's Topical Meeting on Reactor Physics Organized and hosted by the Canadian Nuclear Society (PHYSOR 2006)*, (CD ROM) Vancouver, BC, Canada, September 10–14, 2006.
- 15) B. R. Nease, D. P. Griesheimer, D. L. Millman, and D. F. Gill, "Geometric Templates for Improved Tracking Performance in Monte Carlo Codes," *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC 2013)*, Paris, France, October 27–31 (2013).
- 16) D. F. Gill, B. R. Nease, and D. P. Griesheimer, "Movable Geometry and Eigenvalue Search Capability in the MC21 Monte Carlo Code", *Int. Conf. Math. Comp. Meth. Appl. Nucl. Sci. Eng.*, (CD ROM) Sun Valley, Idaho, USA, May 5–9, 2013.
- 17) J. Spanier, "Two Pairs of Families of Estimators for Transport Problems," *SIAM J. Appl. Math.*, **14**, pp. 702–713 (1966).
- 18) D. P. Griesheimer and M. H. Stedry, "A Generalized Framework for In-Line Energy Deposition during Steady-State Monte Carlo Radiation Transport," *Int. Conf. Math. Comp. Meth. Appl. Nucl. Sci. Eng.*, (CD ROM) Sun Valley, Idaho, USA, May 5–9, 2013.
- 19) T. J. Urbatsch, *et. al.*, "Estimation and Interpretation of keff Confidence Intervals in MCNP," LA-12658, Los Alamos National Laboratory, Los Alamos, NM (1995).
- 20) P. N. Brown, G. D. Byrne and A. C. Hindmarsh, "VODE: A Variable-Coefficient ODE Solver," *SIAM J. Sci. Stat. Comput.*, **10**, pp. 1038–1051 (1989).
- 21) D. C. Carpenter, "A Comparison of Constant Power Depletion Algorithms," *Proc. Int. Con. Math., Comp. Meth. And Reactor Phys.*, (CD ROM) Saratoga Springs, New York, May 3–7, 2009.
- 22) O. W. Hermann, S. M. Bowman, M. C. Brady, and C. V. Parks, "Validation of the SCALE System for PWR Spent Fuel Isotopic Composition Analyses," Oak Ridge National Laboratory, ORNL/TM-12667 (1995).
- 23) D. P. Griesheimer, D. F. Gill, J. W. Lane, D. L. Aumiller, "An Integrated Thermal Hydraulic Feedback Method for Monte Carlo Reactor Calculations," *Proc. PHYSOR'08 – Int. Conf. Phys. Reactors*, Interlaken, Switzerland, Sept. 14–19, Paul Scherrer Institute, Villigen, Switzerland (CD ROM) (2008).
- 24) B. R. Hanna, D. F. Gill and D. P. Griesheimer, "Spatial Homogenization of Thermal Feedback Regions in Monte Carlo Reactor Calculations," *Proc. PHYSOR'12 – Adv. In Reactor Physics*, April 15–20, Am. Nucl. Soc., LaGrange Park, IL (CD ROM) (2012).
- 25) D. P. Griesheimer, "In-Line Xenon Convergence Algorithm for Monte Carlo Reactor Calculations," *Proc. PHYSOR 2010 – Adv. In Reactor Physics to Power the Nuclear Renaissance*, May 9–14, Am. Nucl. Soc., LaGrange Park, IL (CD ROM) (2010).
- 26) The HDF Group, <http://www.HDFGroup.org>, visited May 5, 2012.
- 27) L. Bondesson, "When is the t -Statistic t -Distributed," *Sankhyā: The Indian Journal of Statistics*, **45**, Series A, Pt. 3, 338 (1983).
- 28) I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Boca Raton, Florida, U. S. A. (1991).

- 29) R. J. Brissenden and A. R. Garlick, "Biases in the Estimation of K_{eff} and its Error by Monte Carlo Methods," *Ann. nucl. Energy*, **13**, pp. 63–83 (1986).
- 30) E. M. Gelbard and R. Prael, "Computation of Standard Deviations in Eigenvalue Calculations," *Progress in Nuclear Energy*, **24**, pp. 237–241 (1990).
- 31) D. J. Kelly, T. M. Sutton, and S. C. Wilson, "MC21 Analysis of the Nuclear Energy Agency Monte Carlo Performance Benchmark Problem," *PHYSOR 2012 – Advances in Reactor Physics*, (CD ROM) Knoxville, Tennessee, USA, April 15–20, 2012.
- 32) J. C. Wagner, A. Haghghat, "Automated Variance Reduction of Monte Carlo Shielding Calculations Using the Discrete Ordinates Adjoint Function," *Nucl. Sci. Eng.*, **128**, pp. 186 (1998).
- 33) J. C. Wagner, E. D. Blakeman, and D. E. Peplow, "Forward-Weighted CADIS Method for Global Variance Reduction," *Trans. Am. Nucl. Soc.*, **97**, pp. 595–597 (2007).
- 34) J. L. Hunter and T. M. Sutton, "A Method for Reducing the Largest Relative Errors in Monte Carlo Iterated-Fission-Source Calculations," *Int. Conf. Math. Comp. Meth. Appl. Nucl. Sci. Eng.*, (CD ROM) Sun Valley, Idaho, USA, May 5–9, 2013.
- 35) T. Ueki and F. B. Brown, "Stationarity Modeling and Informatics-Based Diagnostics in Monte Carlo Criticality Calculations," *Nucl. Sci. Eng.*, **149**, pp. 38–50 (2005).
- 36) T. M. Sutton, P. K. Romano and B. R. Nease, "On-the-Fly Dominance Ratio Calculation Using the Noise Propagation Matrix Method," *Prog. Nucl. Sci. Tech.*, **2**, pp. 749–756 (2011).
- 37) B. R. Nease and T. Ueki, "Time Series Analysis of Monte Carlo Fission Sources—III: Coarse-Mesh Projection," *Nucl. Sci. Eng.*, **157**, pp. 51–64 (2007).
- 38) N. E. Horelik, B. R. Herman, B. Forget, and K. S. Smith "Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS)", *Int. Conf. Math. Comp. Meth. Appl. Nucl. Sci. Eng.*, (CD ROM) Sun Valley, Idaho, USA, May 5–9, 2013.
- 39) J. E. Hoogenboom, W. R. Martin and B. Petrovic, "Monte Carlo Performance Benchmark for Detailed Power Density Calculation in a Full Size Reactor Core," Benchmark Specifications Revision 1.2, July 2011, <http://www.nea.fr/dbprog/MonteCarloPerformanceBenchmark.htm> (2011).
- 40) D. J. Kelly III, B. N. Aviles, and B. R. Herman, "MC21 Analysis of the MIT PWR Benchmark: Hot Zero Power Results", *Int. Conf. Math. Comp. Meth. Appl. Nucl. Sci. Eng.*, (CD ROM) Sun Valley, Idaho, USA, May 5–9, 2013.