Nuclear Science and Engineering

ISSN: 0029-5639 (Print) 1943-748X (Online) Journal homepage: https://www.tandfonline.com/loi/unse20

Development of the Assembly-Level Monte Carlo Neutron Transport Code M3C for Reactor Physics Calculations

Anek Kumar, Umasankari Kannan & S. Ganesan

To cite this article: Anek Kumar, Umasankari Kannan & S. Ganesan (2019): Development of the Assembly-Level Monte Carlo Neutron Transport Code M3C for Reactor Physics Calculations, Nuclear Science and Engineering, DOI: 10.1080/00295639.2019.1645502

To link to this article: https://doi.org/10.1080/00295639.2019.1645502

Published online: 12 Aug 2019.

Submit your article to this journal

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=unse20

NUCLEAR SCIENCE AND ENGINEERING
© 2019 American Nuclear Society
DOI: https://doi.org/10.1080/00295639.2019.1645502

Check for updates

Development of the Assembly-Level Monte Carlo Neutron Transport Code M3C for Reactor Physics Calculations

Anek Kumar,[a]* Umasankari Kannan,[a] and S. Ganesan[b]

[a]*Bhabha Atomic Research Centre, Reactor Physics Design Division, Mumbai-400085, India*
[b]*Bhabha Atomic Research Centre, Formerly Raja Ramanna Fellow of the DAE, Mumbai-400085, India*

Received April 24, 2019
Accepted for Publication July 16, 2019

Abstract — *The general geometry continuous-energy Monte Carlo code M3C is currently under development at the Bhabha Atomic Research Centre for reactor physics calculations. The development of the Monte Carlo code M3C for reactor design entails the use of continuous-energy nuclear data and Monte Carlo simulations for each of the neutron interaction processes. This paper describes the current status of the development of the code. The performance and accuracy of the code in application to a variety of problems have been investigated. The important features of this code are treatment of heterogeneous lattices by general geometry, use of point cross sections along with unionized energy grid approach, thermal scattering model for low-energy treatment, probability table treatment in unresolved resonance range, and capability of handling the microscopic fuel particles (TRISO) dispersed randomly, which is very useful in modeling high temperature gas-cooled reactor fuels. Apart from all of the important features in any Monte Carlo code available worldwide, the M3C code has an advanced capability to handle the geometry, which is not described by mathematical equations but only represented by the geometrical points. The code has been validated for its accuracy against a large number of sample problems covering a wide range from simple (like spherical) to complex geometry (like pressurized heavy water reactor lattice) and including randomly dispersed TRISO fuel particle systems. The code is presently restricted to assembly-level calculations.*

Keywords — *Monte Carlo, nuclear data, continuous energy.*

Note — *Some figures may be in color only in the electronic version.*

I. INTRODUCTION

Monte Carlo neutron transport codes in various countries[1–3] are widely used in various reactor physics applications, traditionally related to criticality safety analyses, radiation shielding problems, and validation of deterministic transport codes, due to their capability to model complex geometry and interaction physics. The basic nuclear data used in Monte Carlo calculations are those available in continuous-energy format with no discretization and condensation in any energy range. Therefore, in Monte Carlo method knowledge of detailed neutron interactions with matter through basic nuclear data is directly available for the neutron transport calculation. Approximations, such as the discretization used in multigroup methods in angle, space, and homogenization of the regions, are also avoided.

In the Bhabha Atomic Research Centre (BARC), advanced reactor designs are being pursued in India.[a] India is also pursuing a comprehensive high temperature reactor program.[4,5] In this context, the requirement of an indigenous Monte Carlo code for reactor physics design has been recognized as essential. In BARC, the MONALI (Ref. 6) code, based on Monte Carlo method that utilizes the multigroup cross-section data in WIMSD format[7] for neutron transport calculation, had been developed and used extensively. The

*E-mail: kanek@barc.gov.in

[a] In the Indian context, the nuclear power program is described in the official website: http://www.dae.gov.in

MONALI code is no more in use. In our present work, a fresh effort has been initiated toward the independent development of a continuous-energy general geometry Monte Carlo code starting from the basic point data derived in a compact ENDF (ACE) format obtained by processing evaluated nuclear data files, such as ENDF/B-VIII, JENDL-4 etc., in ENDF/B format[8] using the NJOY (Ref. 9) nuclear data processing code system. The development of our Monte Carlo code requires a full understanding of several topics such as neutron interactions, ENDF/B retrieval and processing technology, Monte Carlo methods, mathematics and computer programming, coordinate geometry, etc. The name of our code is M3C and it is derived from the Monte Carlo Criticality Calculation Code. The M3C Monte Carlo code has been under development at BARC for the past few years.

This paper is the first journal paper on the M3C code. The M3C code has all the capabilities for use in practical reactor design, such as unionized energy grid approach, explicit sampling of delayed-neutron spectrum, probability table treatment in unresolved resonance range (URR), and thermal treatment including $S(\alpha,\beta)$ in low-energy range including the handling of microscopic fuel particles (TRISO) dispersed randomly in systems. The capability of handling the randomly dispersed TRISO particles is useful for physics design of the high temperature reactors that are under development in India.[4,5] The modeling of randomly dispersed fuel participles is beyond the capabilities of deterministic reactor transport code. Due to the complexity in neutron tracking, the modeling of such geometry is complicated even in any Monte Carlo code. Extra efforts have been made to include this capability.

This code has been extensively validated for the calculation of the effective and infinite multiplication factor (with reflective boundary condition) of the fuel assembly in different nuclear reactors. The energy range of the neutrons is limited by the energy range of the neutron cross-section tables in our design of the code, which is 0 to 20 MeV. Every interaction of neutron with matter has been modeled and taken into account in the final reaction tally of the calculation (if reaction data are available in the nuclear data file). Presently, the code has been developed only for neutron transport. The code has been written for Windows and MS-DOS, as well as the Linux environment, using the FORTRAN 90 programming language. The M3C code can also be run in parallel in computer clusters and multicore workstations. Parallelization in the M3C code is implemented using the Message Passing Interface and a very simple procedure that divides the entire simulation into several parallel subtasks.

Already in the literature,[10,11] we have noted that a combination of a Monte Carlo technique and a conventional deterministic method is used for whole-core reactor calculation with burnup. A continuous-energy Monte Carlo technique is used to calculate multigroup constants of a fuel bundle (lattice) and is followed up by a three-dimensional core simulation performed using a conventional deterministic method. This combination is expected to enable very high-quality calculations while using feasible computing times and resources. In the future scope of work, one of the tasks that will be implemented in a future version of the M3C code is the inclusion of the option for generation of the deterministic group constants for whole-core calculation with burnup.

II. MONTE CARLO METHOD IN REACTOR PHYSICS

This section describes the general neutron transport equation, its relation to the eigenvalue equation, and the Monte Carlo solution of the eigenvalue equation. The methodology is described in many textbooks[12,13] and technical reports. Here, it is just a reproduction of the same for the sake of completeness for interested readers and kept as short as possible to save space.

In Monte Carlo method, since one does not solve any differential equation, there is no need for space, energy, and angle discretization. The principle is to follow individual particle histories (as many as possible). Then the particle average behavior is inferred, using the central limit theorem, from the simulated particles. Individual probabilistic events (such as interactions of particles with material) are simulated sequentially. The probability distributions governing these events are statistically sampled to describe the total phenomena.

The time-independent form of the neutron transport equation that is used to determine the criticality condition of the system is

$$\left[\hat{\Omega}.\vec{\nabla}+\Sigma_t(\vec{r},E)\right]\psi\left(\vec{r},\hat{\Omega},E\right)=\int dE'\int d\Omega'\Sigma_s\left(\vec{r},E'\rightarrow E,\hat{\Omega}'.\hat{\Omega}\right)\psi\left(\vec{r},\hat{\Omega}',E'\right)+\frac{1}{k_{eff}}\frac{\chi(E)}{4\pi}\int\int\upsilon\sum_f\left(\vec{r},E'\right)\psi\left(\vec{r},E',\vec{\Omega}'\right)d\vec{\Omega}'\,dE', \quad (1)$$

where the independent variables are defined as

$\vec{r}$ = positional coordinates

E = kinetic energy of neutrons

$\hat{\Omega}$ = neutron angular direction vector

and where the dependent variables are defined as

$\psi\left(\vec{r},\hat{\Omega},E\right)$ = neutron directional flux at $\vec{r}$ with direction $\hat{\Omega}$ and energy E

$\Sigma_s\left(\vec{r},E'\rightarrow E,\hat{\Omega}'.\hat{\Omega}\right)$ = macroscopic scattering cross section that describes the probability of a neutron with an initial energy E' and direction $\hat{\Omega}'$ that undergoes a scattering collision at position $\vec{r}$, which places it into a direction $\hat{\Omega}$ with energy E

$\nabla..\Omega\,\psi\left(\vec{r},\hat{\Omega},E\right)$ = net leakage per unit volume at $\vec{r}$ with direction $\hat{\Omega}$ and energy E

$\Sigma_t(\vec{r},E)$ = total macroscopic cross section at position $\vec{r}$ with energy E,

where

$\chi(E)$ = fission spectrum

ν = average number of neutrons produced per fission

$\Sigma_f\left(\vec{r},E\right)$ = fission cross section

k_{eff} = k-eigenvalue.

Here no external source is considered and an additional factor in fission multiplicity is introduced. This is understood as the hypothetical adjustment needed for a steady-state solution to exist.

Equation (1) can be written in operator form as

$$(L+T)\,\psi = S\psi + \frac{1}{k_{eff}}M\psi\ , \tag{2}$$

where

L = leakage

T = transport

S = scatter

M = fission operator.

Equation (2) can also be rearranged to

$$\psi = \frac{1}{k_{eff}}(L+T-S)^{-1}M\psi = \frac{1}{k_{eff}}F\psi\ , \tag{3}$$

whereby the time-dependent equation becomes an eigenvalue problem.

Equation (3) may be solved numerically using the standard power iteration method:

$$\psi^{(n+1)} = \frac{1}{k_{eff}{}^{(n)}}F\psi^{(n)},$$
$$n = 0,1,\ \ldots.,\ \text{given}\ k^{(0)}\ \text{and}\ \psi(0)\ . \tag{4}$$

Most Monte Carlo codes use the standard power method for solving k-eigenvalue problems. This is done by simulating successive neutron generations; each generation (cycle) contains a certain number m of neutrons. The neutron transport is simulated with the use of known probability distribution functions (pdf's) of various random variables (e.g., the distance between collisions, the type of isotope and reaction at the collision site, the number of new fission neutrons, scattering angles, etc.).

The effective multiplication factor in cycle n is defined as the ratio

$$k_{eff,n} = \frac{number\ of\ source\ neutrons\ in\ cycle\ n+1}{number\ of\ source\ neutrons\ in\ cycle\ n}\ . \tag{5}$$

The fission source is thus defined only in m sites in the system. The new fission neutrons sampled during the neutron transport are added into the so-called delay bank. At the end of each cycle, the delay bank is converted into the fission source for the subsequent cycle.

Another important parameter is the total number N of simulated cycles. The k_{eff} estimate is computed as the arithmetic average of the values obtained at each cycle in order to reduce the statistical error. The first I_C cycles in a criticality calculation are in inactive cycles, where the spatial source changes from the initial definition to the correct distribution for the problem. The quantities of interest, like k_{eff}, neutron flux, reaction rates, etc., must be combined over a number of active cycles with a converged fission source. The first I_C values of k_{eff} are therefore withdrawn from the arithmetic average. Estimates of the k_{eff} and corresponding standard deviation of the mean Δk_{eff} are written as

$$k_{eff} = \frac{1}{N-I_C}\sum_{n=I_C+1}^{N} k_{eff,n} \tag{6}$$

and

$$\Delta k_{eff} = \sqrt{\frac{1}{(N - I_C - 1)(N - I_C)} \sum_{n=I_C+1}^{N} \left(k_{eff,n} - k_{eff}\right)^2} \ . \tag{7}$$

III. TREATMENT OF POINTWISE NUCLEAR DATA IN THE M3C CODE

One of the advantages of Monte Carlo method is the utilization of continuous-energy nuclear data, which means fewer approximations are made in the calculation of energy-dependent cross sections, as compared with the deterministic method based on multigroup theory. The neutron-induced reaction cross-section data governing the various physical interaction of neutrons with various nuclei for continuous-energy problems have been represented using the ACE format. This format is described in Appendix-F in the MCNP (Ref. 1) manual. These nuclear data libraries are produced from the original ENDF by nuclear data processing codes, such as the NJOY code system,[9] and are given in the form of tabulated energy cross-section pairs. The use of a standard cross-section format allows for a direct comparison of the M3C with other codes since the same cross-section libraries can be used. At the current stage, the code is limited to using what is known as type-1 ACE-formatted files, which have data stored in sequential and in 80 characters per record in ASCII format.

The ACE format contains continuous-energy cross sections for the following types of reactions: elastic scattering, fission (or first-chance fission, second-chance fission, etc.), inelastic scattering, (*n,xn*), (*n*,γ), and various other absorption reactions. For those reactions with one or more neutrons in the exit channel, secondary angle and energy distributions may be provided. In addition, fissionable nuclides have total, prompt, and/or delayed nubar ν as a function of energy and neutron precursor distributions. Many nuclides also have probability tables to be used for accurate treatment of self-shielding in the URR. For bound scatterers, separate tables with $S(\alpha,\beta)$ scattering law data can be used. Most of the ACE-formatted files for each isotope used for the simulation using this code have been downloaded from the website of the International Atomic Energy Agency (IAEA) Nuclear Data Services.[b] These IAEA ACE libraries are generated for Monte Carlo transport codes for use in the analysis of accelerator-driven systems.[14]

[b] See https://www-nds.indcentre.org/ads/

III.A. Unionized Energy Grid Approach

In ACE-formatted nuclear data files, the cross-section data are given as tables of energy cross-section pairs. In Monte Carlo transport simulation, energy grid search and linear interpolation are carried out in order to obtain the specific cross sections (cross sections in the URR always need special treatment rather than direct interpolation from tabulated data) at each energy point E:

$$\sigma(E) = \frac{E - E_{j-1}}{E_j - E_{j-1}} \left(\sigma_j - \sigma_{j-1}\right) + \sigma_{j-1} \ . \tag{8}$$

To reproduce the accuracy of cross sections with the original ENDF, each nuclide has a unique energy grid in pointwise nuclear data, and hence the procedure of grid search and interpolation has to be repeated for every nuclide and every time when the neutron energy is changed or the neutron crosses over the boundary of materials. This process is very time consuming.

To reduce the number of energy grid search operations and cross-section interpolations, a method using unionized energy grid was first implemented in the code Serpent,[15] a reactor physics code developed at the VTT Technical Research Centre of Finland. The same method is implemented in the M3C code[16] also. The key idea of this method is to merge the grids of all nuclides into a unionized energy grid upon which all cross sections are reconstructed, including the material macroscopic total cross sections for rapid calculation of neutron mean free path. In this way, the grid search and interpolation factor calculation are performed only once at each energy. More importantly, the accuracy of the original nuclear data cannot be compromised during the process. To test the implementation of the unionized energy grid approach in the code and its accuracy, a test problem is considered. A unionized energy grid is generated combining two resonating nuclides, ^{239}Pu and ^{233}U. Cross sections are reconstructed upon the unionized energy grid and compared with cross-section data in an original ACE-formatted nuclear data file. Therefore, the extracted data from nuclide-specific original ACE-formatted library and reconstructed cross section upon the unionized energy grid are again put up in ENDF/B format and comparison studies are performed in such a way that they fulfill all international standards. In Fig. 1, the total cross-section data of ^{239}Pu in the original ACE-formatted data library are compared with the total cross section reconstructed upon the unionized energy grid. The difference between the two data sets varies from −0.711% to 0.710%. Similarly, in Fig. 2, this is done in the case of ^{233}U where

the difference varies from −0.397% to 0.811%. These types of results give full confidence while using the unionized energy grid approach in neutron transport calculation in realistic problems.

In both Figs. 1 and 2, the comparison graphs have been generated using the well-known software COMPLOT, which is a part of the PREPRO code system.[17] The unionized data are cast into ENDF/B format as PREPRO is compatible to the ENDF/B format.

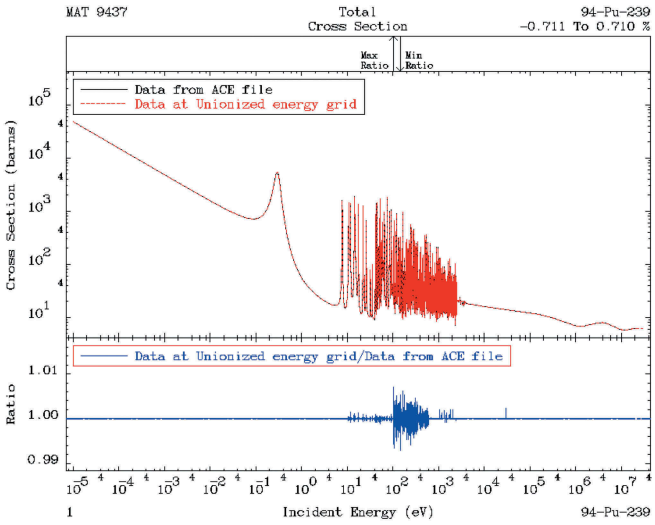

Fig. 1. Comparison of total cross-section data of ^{239}Pu between original data presented in ACE file and data reconstructed from unionized energy grids in case of two isotopes, ^{239}Pu and ^{233}U.

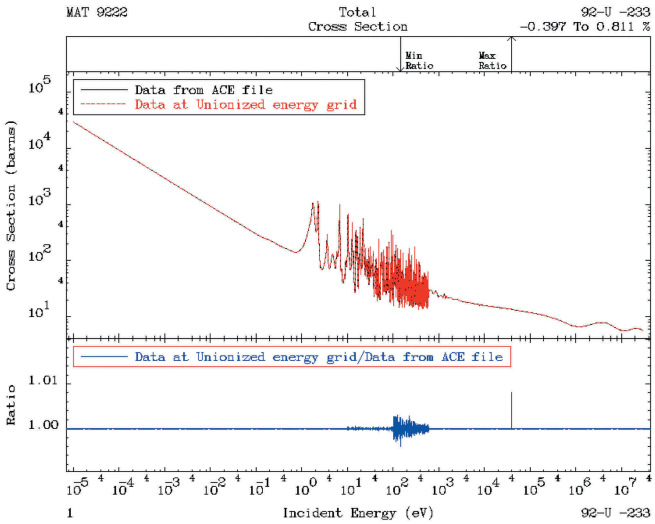

Fig. 2. Comparison of total cross section of ^{233}U data between original data presented in ACE file and data reconstructed from unionized energy grids in case of two isotopes, ^{239}Pu and ^{233}U.

III.B. Thermal Scattering Treatment

In a Monte Carlo code, one must account for the effect of the thermal motion on both the integrated cross section as well as secondary angle and energy distributions. For integrated cross sections, it is possible to calculate thermally averaged cross sections by applying a kernel Doppler broadening algorithm to data at 0 K (or some temperature lower than the desired temperature). Cross-section libraries processed by NJOY (Ref. 9) already include Doppler broadening of elastic, capture, fission, and other low-threshold absorption cross sections. Other than this, at thermal neutron energies, individual thermal motions of the scattering nucleus become important for the calculation of neutron scattering cross sections and the energy and angular distributions of secondary neutrons. There are two thermal treatments in the Monte Carlo code. One of these is the free-gas model in which, for elastic collisions, nuclei of the medium are assumed free from any molecular or metallic bindings and behave as a free gas. That the cross section varies slowly with energy is also assumed, which is very much valid in the case of elastic scattering. Atoms of the medium are assumed to have a Maxwellian distribution with ambient thermal temperature. Although all nuclides present in the reactor must be treated with this model, this effect is more effective in the case of light nuclides. The free-gas treatment is entirely responsible for neutron upscattering, the process by which a neutron can increase its energy in a nuclear reactor.

Second, the $S(\alpha,\beta)$ scattering model is available, which accounts for chemical binding and crystalline effects at very low energies. For bound scatterers, this information is stored as thermal scattering law data in the evaluated nuclear data library in ENDF/B files for a few materials, such as light water, heavy water, and graphite, and converted into discrete data in secondary energy and direction by nuclear data processing codes and appended to ACE files that are used by the M3C Monte Carlo code for the simulation. Coherent elastic cross sections are converted to a cumulative "stair-step" form and stored in ACE format. The angular representation for incoherent elastic scattering is stored directly. Angular distribution of outgoing neutrons is not needed for coherent elastic scattering reactions. The incoherent inelastic scattering energy distributions are converted into probability bins with the equally probable angles left unchanged. The bins can have equal probabilities or variable probabilities. In the latter case, outlying bins with smaller probabilities are provided to extend the sampling to rare events. All this information is stored in an ACE-formatted thermal data

library that has been used by Monte Carlo code for the simulation. Typically, at room temperature, the free-gas model is used from around 10 to 4 eV, and then the $S(\alpha,\beta)$ scattering model is used below that. In case of nonavailability of the $S(\alpha,\beta)$ scattering model, the free-gas model again is used below 4 eV. The unionized data are cast into ENDF/B format as PREPRO is compatible to the ENDF/B format.

III.C. Unresolved Resonance Treatment

One might think that by using continuous-energy cross sections rather than multigroup cross sections the problem of self-shielding can be completely avoided. Unfortunately, this is not the case because one must still deal with self-shielding in the unresolved resonance energy range. In the URR, the resonances are so narrow and closely spaced that is not possible to experimentally resolve the details of all resonances. Therefore, in the unresolved resonance region we do not know the cross section as a continuous function of energy; we only know the probability distribution of the cross sections, and these distributions can result in significant self-shielding effects. The averaged over probability distribution function expected value is what is commonly referred to as an infinite-dilute cross section; for a more detailed discussion of this subject, see Ref. 18.

In order to more faithfully account for resonance structure and the resulting self-shielding effect in the URR, the probability table method was proposed. The cross-section probability tables are appropriately distributed throughout the unresolved energy range of a given isotope. These tables consist of a probability distribution of cross sections to be used in an URR. These probability tables are generated with a cross-section processing code, like NJOY (Ref. 9), by using the evaluated statistical information about the resonance to calculate the cumulative pdf for the microscopic cross section. The elastic, the fission, and the radiative capture cross sections are also tabulated as the average values of each of these partials is conditional upon the value of the total. In general, a different set of tables is required at every temperature for each nuclide with a URR in a simulation. The unresolved resonance probability tables affect only integrated cross sections, not the secondary angle-energy distribution. The procedure in this work for implementation of the probability table in URR is adopted from the OpenMC (Ref. 19) code developed by members of the Computational Reactor Physics Group at the Massachusetts Institute of Technology.

In the absence of probability table in URR, infinite-dilute cross sections (single average values) are used in Monte Carlo neutron transport simulations. The use of unresolved resonance probability tables can be switched on and off using as an option in the input of the problem. Since the probability table sampling has to be carried out during tracking, the transport cycle tends to slow down significantly. All cross sections are sampled every time whenever the neutron scatters to a new energy in the unresolved energy range.

IV. GEOMETRY CAPABILITY OF THE CODE

In order to model arbitrarily complex geometric objects, the M3C code uses a constructive solid geometry representation. In such a representation, any closed volume can be represented as the union, intersection, and/or difference of multiple spaces. Almost all geometries of interest in particle transport can be modeled with first- and second-order surfaces, including some fusion geometries where a fourth-order torus is required. As in MCNP and Serpent,[19] repeated structures are handled through the use of universes. The user can specify that a cell is to be filled with something called a universe, like in the case of MCNP. Similar to MCNP, in this code a universe also can be either a lattice or an arbitrary collection of cells and a cell can be filled with this universe. Vacuum or reflective boundary conditions can be applied to any surface giving the user full flexibility in the treatment of boundaries.

There are two ways to specify surface parameters, either by supplying the appropriate coefficients needed to satisfy the surface equation or by specifying known geometrical points on a surface. In a conventional geometry simulated by Monte Carlo code, the surfaces are specified by supplying the appropriate coefficient needed to satisfy the surface equation. In some cases, it becomes more viable to represent the surface by geometrical point rather than by an equation.

The geometry module in the M3C code can handle the geometry specified by the surfaces represented by geometrical points. This is an important capability of the M3C code from a geometry point of view.

The most crucial part of the geometry module in this case is to get the intersection points between the neutron path and underlying surfaces that are defined by geometric points. In the case of surfaces defined by geometrical points, this is very challenging as compared to the case of surfaces defined by a surface equation. Here, one has to deal with the intersection problem between a ray (constructed by neutron

path) and a surface constructed by a set of points near the corresponding ray. The basic idea is to identify those geometry points that are very close to the ray, and the ray can intersect the surfaces constructed by these points, called inclusion points. To speed up this calculation, the whole problem is divided into small lattices and the lattice index corresponding to every geometry point in which it lies is found. After that, a quasi-Monte Carlo method[20] is used to correct the inclusion points around the ray. In Ref. 20, a clustering technique and an algorithm (algorithm 2 in Ref. 21) for computing the cluster is discussed. The same algorithm is implemented in this code. After getting the resultant cluster, a set of quadratic surfaces are constructed passing through the geometrical point in each cluster. Then, the ray-tracing algorithm is applied as done earlier. A typical picture of an object constructed by randomly distributed geometrical points on the surface is presented in Fig. 3.

In the Indian high temperature reactor program,[4,5] the high temperature reactor fuels are composed of thousands of TRISO particles randomly dispersed in a graphite matrix. The modeling of such geometry is complicated, especially using continuous-energy Monte Carlo codes, which are not able to apply any deterministic corrections in the calculation. A geometry routine is developed for modeling randomly dispersed particle fuel in the Monte Carlo code. In this code, an improved "fast" random sequential addition (RSA) method[22] is used to generate random arrangements for TRISO particles. In the basic RSA method,[23] a position *XYZ* for a spherical fuel particle within the given volume (such that *XYZ* is at least a distance *R* from the container boundaries where *R* is the radius of the TRISO particle) is randomly selected. The center of the spherical fuel particle is considered at position *XYZ*. Simultaneously care is also taken that this sphere centered at *XYZ* does not overlap any previous randomly placed spheres. Because of this step, checking all previous spheres for no overlap, the basic RSA method is time consuming and takes very large amounts of computing time. In the improved fast RSA, the test for overlapping is performed only on the nearest neighboring spheres rather than an exhaustive check of all previous spheres. This is achieved by defining a lattice with spacing $h \leq 2R/\sqrt{3}$, where R is the radius of the TRISO particle. This choice of lattice spacing guarantees that each lattice box can contain at the most one sphere center, and only $m = [R/h] + 1$ neighboring boxes need to be checked for overlapping spheres in the $\pm X$, $\pm Y$, and $\pm Z$ directions. Lists are maintained to keep track of the spheres contained in each lattice box as well as the empty lattice boxes. The random particle (TRISO particle) distribution generator works by prompting the user to provide information on the volume type and dimensions, particle data, and number of random particles. The code then generates a distribution inside the desired volume without overlapping any particles. A typical picture of TRISO particle distribution in a graphite matrix in presented in Fig. 4 for demonstration purpose.

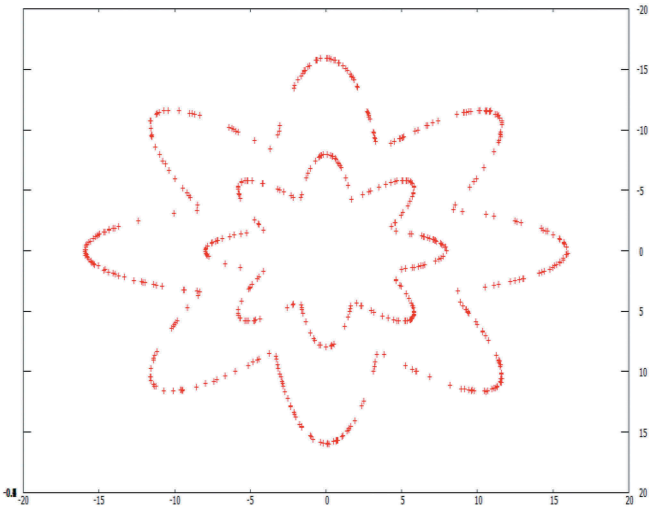

Fig. 3. Example of a typical picture of an object (cross-section view across a plane) constructed by randomly distributed geometrical points on the surface.

V. RESULTS AND VALIDATION OF THE CODE REGION

V.A. Validating the $S(\alpha,\beta)$ Treatment in the Code

This code is being developed for the first time in India. Each capability of this code is tested extensively. A theoretical problem that highlights the effect of the bound-atom scattering treatment in multiplication factor

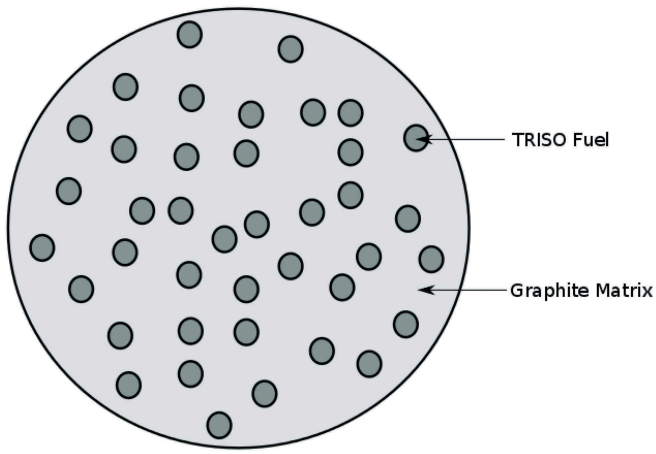

Fig. 4. TRISO particles distributed in a graphite matrix.

is a very simple pin cell problem proposed by Cullen et al.[24] This problem consists of an infinite lattice of fuel pins with a 2-in. pitch and varying fuel pin radii. The fuel consists of only two nuclides, ^{235}U and ^{238}U, and furthermore there is no fuel cladding or gap. The ratio of water to fuel has been defined to maximize the effect of thermal scattering since at present implementation of this effect is being tested in the code. From the geometry and material points of view, this problem is so simple that any differences in answers can be almost solely attributed to the bound scattering treatment. In the report by Cullen et al.,[24] six cases are considered corresponding to each combination of three different fuel pin radii and two different scattering treatments. The fuel pin radii specified were 1/2, 1/4, and 1/8 of an inch, and each of these models was to be run with and without the $S(\alpha,\beta)$ scattering law data.

The results in the M3C code with and without the $S(\alpha,\beta)$ treatment are compared with the published results in Ref. 25 using OpenMC (Ref. 25) and MCNP5 (Ref. 26) and the same ENDF/B-VII.0 cross-section libraries. With the same cross-section libraries and similar physics treatments, it would be expected that the results between the M3C and those published earlier from OpenMC and MCNP5 should be very close. As discussed in Ref. 25, each MCNP run had 50 inactive batches and 1000 active batches, each with 100 000 particles. Due to the nonavailability of the combined estimator for the *k*-effective like MCNP does, OpenMC runs had 50 inactive batches and 4000 active batches, each with 100 000 particles. In the M3C code also, only single estimator (collision) is used for the *k*-effective like the OpenMC code. Therefore, for an exact comparison, the M3C code runs also had 50 inactive batches and 4000 active batches, each with 100 000 particles. The results are presented in Table I.

V.B. Validation of Implementation of the Probability Table Method

For testing the implementation of the probability table method in URR in the M3C code, two test problems (consisting of a theoretical and a practical test problem) are considered.

V.B.1. Infinite, Homogeneous Medium Test Problem

This test problem is a simple infinite, homogeneous medium system.[27] It is composed of a 10:1 ratio of ^{238}U and ^{235}U nuclei that is brought to critical with the addition of ^{10}B. The test problem is constructed to have an intermediate/fast neutron energy spectrum so that it is sensitive to the handling of the URR. In this study, the eigenvalue (k_{eff}) of the system is simulated at 293.6 K temperature with and without use of probability table and compared with the published results of OpenMC using the same ENDF/B-VII.1 cross-section libraries; results are presented in Table II.

V.B.2. Big Ten Critical Assembly

The second benchmark problem that has been chosen as a means of validating the implementation of the probability table method in the M3C code is the Big Ten critical assembly.[c] This real system with an intermediate/fast spectrum is chosen from the International Criticality Safety Benchmark Evaluation Project (ICSBEP) handbook.[28] This assembly is a large, mixed uranium metal cylindrical core of 10% enrichment surrounded by a ^{238}U reflector. In Ref. 25, the results from MCNP5 and OpenMC codes using the ENDF/VII.0 cross-section libraries are presented. In the M3C run, the same number of particle histories is taken as in the case of the run in OpenMC in Ref. 25.

TABLE I

Effective Multiplication Factor k_{eff} for Different Cases

Case	M3C Code	MCNP5 Code[a]	OpenMC Code[a]
1/2-in. pin, no $S(\alpha,\beta)$	1.01655 ± 0.00009	1.01649 ± 0.00004	1.01656 ± 0.00006
1/2-in. pin, $S(\alpha,\beta)$	0.96914 ± 0.00010	0.96912 ± 0.00004	0.96814 ± 0.00006
1/4-in. pin, no $S(\alpha,\beta)$	1.01326 ± 0.00010	1.01330 ± 0.00005	1.01328 ± 0.00006
1/4-in. pin, $S(\alpha,\beta)$	0.92231 ± 0.00010	0.92226 ± 0.00005	0.92219 ± 0.00006
1/8-in. pin, no $S(\alpha,\beta)$	1.01323 ± 0.00010	1.01327 ± 0.00007	1.01327 ± 0.00006
1/8-in. pin, $S(\alpha,\beta)$	0.90917 ± 0.00011	0.90921 ± 0.00007	0.90921 ± 0.00006

[a]As published in Ref. 25.

[c] See IEU-MET-FAST-007 (Ref. 28).

TABLE II

Comparison of Effective Multiplication Factor k_{eff}

Case	M3C Code	OpenMC Code[a]
Probability table off	1.00010 ± 0.00012	1.00001 ± 0.00008
Probability table on	1.00382 ± 0.00014	1.00390 ± 0.00009

[a]As published in Ref. 27.

In Table III, the results obtained from the developed M3C code are compared with results from the other two well-known Monte Carlo codes, MCNP and OpenMC. The results from both of the test problems (in Tables II and III) clearly show that the unresolved resonance probability table treatment in the M3C code has been implemented correctly.

V.C. Some Other Benchmarks

V.C.1. JEZEBEL and GODIVA Benchmarks

The specifications adopted for these benchmark calculations are those given in the ICSBEP[d] fast reactor benchmarks. JEZEBEL is a bare sphere of plutonium metal and GODIVA is a bare sphere of highly enriched uranium. Both the benchmarks are single-region, simple geometry, and uniform composition, which conveniently facilitates calculation testing.

The JEZEBEL spherical homogeneous model has a core radius of 6.385 cm and the composition shown in Table IV. The measured eigenvalue with vacuum boundary condition is 1.000 ± 0.002 and the k_{eff} calculated using the developed code is 1.00056 ± 0.00079.

The GODIVA spherical homogeneous model has a core radius of 8.741 cm and the composition shown in Table V. The measured eigenvalue with vacuum boundary condition is 1.000 ± 0.001. The value of k_{eff} is found to be 1.00034 (± 0.00062) using the developed code, which is in good agreement with the experimental data.

V.C.2. Criticality Benchmark for Water-Reflected Metal Assembly

This water-reflected metal assembly benchmark is a simpler geometry of the highly enriched uranium (HEU) sphere in a cylindrical tank of water[29] (see Fig. 5). The radius of sphere considered is 6.5537 cm. The cylindrical tank height is 70 cm and the diameter is 60 cm. Isotopic number densities for the sphere and the reflector are given in Table VI.

The benchmark value of the effective multiplication factor of this assembly is 1.0003 ± 0.005. The calculated value of the effective multiplication factor from the M3C code is found to be 0.9989 ± 0.0005, which can be considered as in good agreement with the benchmark value of the k_{eff}.

V.C.3. PHWR Fuel Cluster

The pressurized heavy water reactor[30] (PHWR) is a pressure tube–type reactor with heavy water as moderator and coolant, and natural uranium dioxide as fuel. The fuel bundle consists of a 19-rod cluster. Using the developed Monte Carlo code, k-infinity (k_{inf}) is calculated for a 220-MW(electric) PHWR rod fuel cluster in the cold condition assuming that fuel, coolant, and moderator temperatures are all the at same temperature, 25°C. The specification of the 19-rod fuel cluster is presented in Table VII. The details of this

TABLE III

Comparison of Effective Multiplication Factor k_{eff}

Case	M3C Code	MCNP Code[a]	OpenMC Code[a]
Probability table off	1.00079 ± 0.00011	1.00085 ± 0.00005	1.00095 ± 0.00007
Probability table on	1.00472 ± 0.00012	1.00480 ± 0.00005	1.00485 ± 0.00007

[a]As published in Ref. 25.

[d] See http://icsbep.inl.gov.

TABLE IV

JEZEBEL Spherical Homogeneous Model

Isotope	Density (atoms/b · cm)
^{239}Pu	0.03705
^{240}Pu	0.001751
^{241}Pu	0.000117

TABLE V

GODIVA Spherical Homogeneous Model

Isotope	Density (atoms/b · cm)
^{235}U	0.04500
^{238}U	0.002498
^{234}U	0.000492

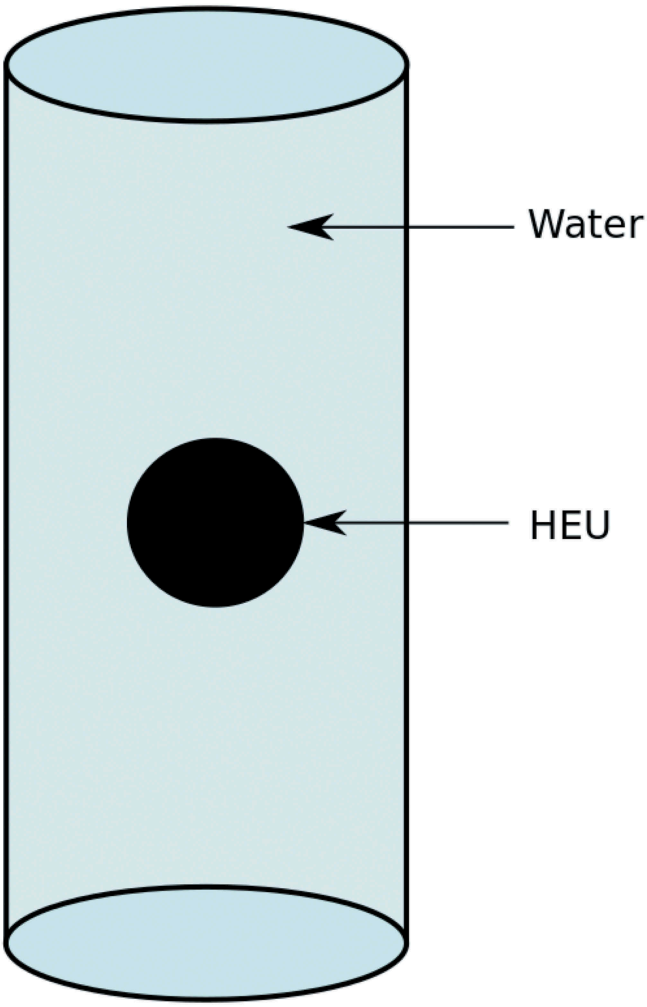

Fig. 5. Water-reflected metal assembly.

TABLE VI

Isotopic Densities for Sphere and Reflector

Region	Isotope	Number Density (atoms/barns · cm)
Sphere	^{234}U	5.3×10^{-4}
	^{235}U	4.703×10^{-2}
	^{236}U	1.0×10^{-4}
	^{238}U	4.9×10^{-4}
Reflector	Hydrogen	6.679×10^{-2}
	Oxygen	3.340×10^{-2}

TABLE VII

Specification of 19-Element 220-MW(electric) PHWR Fuel Cluster

Fuel Material	Natural UO_2
Number of fuel rods in a fuel bundle	19
Ratio ^{238}N/^{235}U	137.8
Fuel density	9.91 gm/cm^3
Moderator material	D_2O
Sheath material	Zr-4
Sheath density	8.91 gm/cc
Fuel bundle length	49.53 cm
Diameter of fuel rod	1.437 cm
Inner diameter of the sheath	1.445 cm
Outer diameter of the sheath	1.521 cm
Pressure tube material	Zr-2
Calandria tube material	Zr-2
Inside diameter of pressure tube	8.26 cm
Outside diameter of pressure tube	9.10 cm
Inside diameter of calandria tube	10.80 cm
Outside diameter of calandria tube	11.06 cm

cluster can be found in Ref. 30. The 19-rod fuel cluster is shown in Fig. 6. We have simulated the same cluster also by the deterministic neutron transport theory code CLUB (Refs. 31 and 32) using the 172-energy group WIMSD nuclear data library based on ENDF/B-VII.0 point data for the k_{inf} with reflective boundary condition. The value of k_{inf} for this cluster using CLUB is 1.14125 (Ref. 25). The same test problem is also simulated using the developed M3C code using a pointwise nuclear data set based on ENDF/B-VII.0. The value of k_{inf} is found to be 1.13971 ± 0.00086, which can be considered in good agreement with the previous value of k_{inf} calculated from the deterministic code CLUB. The comparison with CLUB is done for curiosity and historical purposes as the WIMSD convention–based CLUB code has several

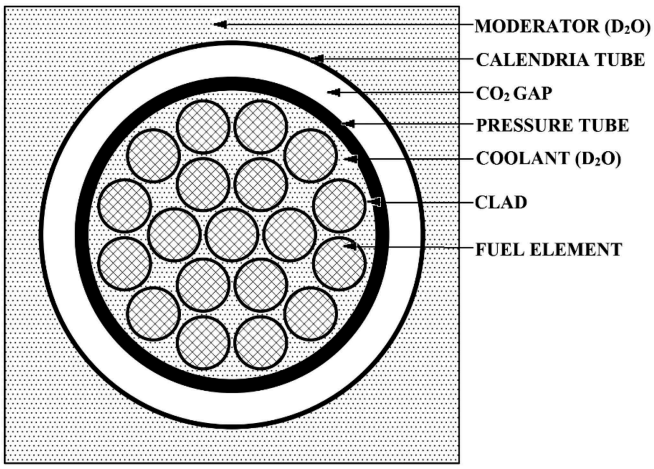

Fig. 6. Unit cell for 19-rod fuel element region.

physics approximations as compared to a Monte Carlo code starting with a basic evaluated nuclear data file. These approximations in WIMSD physics include multigroup treatment, treatment of (*n,xn*) cross sections as negative absorption, truncation of anisotropy in the treatment of cross sections, an upper energy limit of 10 keV for resonance self-shielding, and intermediate resonance approximation, etc. Therefore, the results by a multigroup approximation should not be used to validate a Monte Carlo code. The blind comparison is presented, in the Indian context, merely for historical purposes and these sample comparison results should not be generalized.

VI. CONCLUSIONS

The most important contribution of this work is the indigenous development of a new calculation code for neutron transport by Monte Carlo method using continuous-energy cross-section data. Code development involved modeling every aspect of neutron transport. The development activity had several challenging tasks, such as developing a general geometry module for neutron tracking, understanding and modeling the physics behind each process, comprehensive use of point cross sections in specific ACE format, understanding the Monte Carlo method, and advanced computer programming. From the nuclear data point of view, this code uses standardized ENDF-format reaction data read from ACE-formatted data libraries. This also helps in the benchmarking of this code with other international Monte Carlo codes.

The important features of this code are treatment of heterogeneous lattices by general geometry, use of point cross sections along with a unionized energy grid approach, thermal scattering model for low-energy treatment, use of a probability table for accurate treatment of self-shielding in the URR, and the capability of handling TRISO particles dispersed randomly. The M3C code can be run in parallel in computer clusters and multicore workstations. The characteristic feature of the M3C code is that it can be used for reactor physics calculation at the fuel assembly level.

The results of the simulations of several benchmarks are in good agreement with published results in literature, giving us confidence about the performance of this code for criticality calculations in reactor physics design.

References

1. "MCNP—A General Monte Carlo N-Particle Transport Code," LA-CP-03-0284, Los Alamos National Laboratory (2003).
2. J. LEPPÄNEN et al., "The Serpent Monte Carlo Code: Status, Development and Applications in 2013," *Ann. Nucl. Energy*, **82**, 142 (2015); https://doi.org/10.1016/j.anucene.2014.08.024.
3. V. S. W. SHERRIFFS, "MONK A General Purpose Monte Carlo Neutronics Program," Safety and Reliability Directorate Report SRD R 86, UKAEA (Jan. 1978).
4. I. V. DULERA and R. K. SINHA, "High Temperature Reactors," *J. Nucl. Mater.*, **383**, 183 (2008); https://doi.org/10.1016/j.jnucmat.2008.08.056.
5. R. K. SINHA, "Advanced Nuclear Reactor Systems—An Indian Perspective," *Energy Procedia*, **7**, 34 (2011); https://doi.org/10.1016/j.egypro.2011.06.005.
6. H. C. GUPTA, "MONALI—Rev. 1 A Monte Carlo Code for Analyzing Fuel Assemblies of Nuclear Reactors" BARC-1543, Bhabha Atomic Research Centre (1991).
7. C. J. TAUBMAN, "The WIMS 69-Group Library Tape 166259," AEEW-M1324, United Kingdom Atomic Energy Authority, Winfrith, United Kingdom (1975).
8. "ENDF-6 Formats Manual, Data Formats and Procedures for the Evaluated Nuclear Data File ENDF/B-VI and ENDF/B-VII," M. HERMAN and A. TRKOV, Eds., BNL-90365-2009 (ENDF-102), Brookhaven National Laboratory (June 2009).
9. R. E. MacFARLANE and D. W. MUIR, "The NJOY Nuclear Data Processing System," LA-12740, Los Alamos National Laboratory (1994); http:/t2.lanl.gov/codes/codes.html (current as of Apr. 24, 2019).
10. G. ILAS and F. RAHNEMA, "A Monte Carlo Based Nodal Diffusion Model for Criticality Analysis of Spent Fuel

Storage Lattices," *Ann. Nucl. Energy*, **30**, 1089 (2003); https://doi.org/10.1016/S0306-4549(03)00037-9.

11. H. J. SHIM et al., "Generation of Few Group Diffusion Theory Constants by Monte Carlo Code," *Trans. Am. Nucl. Soc.*, **99**, 343 (2008).
12. J. J. DUDERSTADT and L. J. HAMILTON, *Nuclear Reactor Analysis*, John Wiley & Sons, Inc. (1976).
13. A. HEBERT, *Applied Reactor Physics*, Presses Internationales Polytechnique (2009).
14. D. LOPEZ ALDAMA and A. L. NICHOLS, "ADS-2.0: A Test Library for Accelerator Driven Systems and New Reactor Design," INDC(NDS)-545, International Atomic Energy Agency, Vienna, Austria (2008).
15. J. LEPPANEN, "Two Practical Methods for Unionized Energy Grid Construction in Continuous-Energy Monte Carlo Neutron Transport Calculation," *Ann. Nucl. Energy*, **36**, 878 (2009); https://doi.org/10.1016/j.anucene.2009.03.019.
16. A. KUMAR, "Implementation of Unionized Energy Grid Method in Continuous-Energy 3-D Monte Carlo Neutron Transport Code," presented at ARP 2013—Advances in Reactor Physics: Simulation Techniques and Analysis Methodologies, Anushakti Nagar, Mumbai, India, October 23–25, 2013.
17. D. E. CULLEN, "PREPRO 2007: 2007 ENDF/B Pre-Processing Codes," IAEA-NDS-39, Rev. 13, Nuclear Data Section, International Atomic Energy Agency, Vienna, Austria (Mar. 17, 2007).
18. L. LEVITT, "The Probability Table Method for Treating Unresolved Neutron Resonances in Monte Carlo Calculations," *Nucl. Sci. Eng.*, **49**, 450 (1972); https://doi.org/10.13182/NSE72-3.
19. J. LEPPÄNEN, "Development of a New Monte Carlo Reactor Physics Code. P640," VTT Technical Research Centre of Finland (2007).
20. Y.-S. LIU et al., "A Quasi-Monte Carlo Method for Computing Areas of Point-Sampled Surfaces," *Comput.-Aided Des.*, **38**, 55 (2006); https://doi.org/10.1016/j.cad.2005.07.002.
21. Y.-S. LIU et al., "Surface Area Estimation of Digitized 3D Objects Using Quasi-Monte Carlo Methods," *Pattern Recognit.*, **43**, 3900 (2010); https://doi.org/10.1016/j.patcog.2010.06.002.
22. F. B. BROWN, "A Fast RSA Method for Fuel Particle Packing," *Trans. Am. Nucl. Soc.*, **94**, 571 (2006).
23. B. WIDOM, "Random Sequential Addition of Hard Spheres to a Volume," *J. Chem. Phys.*, **44**, 3888 (1996); https://doi.org/10.1063/1.1726548.
24. D. E. CULLEN et al., "How Accurately Can We Calculate Thermal Systems?" INDC(USA)-107, International Nuclear Data Committee (2004).
25. P. K. ROMANO and B. FORGET, "The OpenMC Monte Carlo Particle Transport Code," *Ann. Nucl. Energy*, **51**, 274 (2013); https://doi.org/10.1016/j.anucene.2012.06.040.
26. X-5 MONTE CARLO TEAM, "MCNP—A General Monte Carlo N-Particle Transport Code, Version 5," LA-UR-03-1987, Los Alamos National Laboratory (2008a).
27. J. A. WALSH et al., "Direct, On-The-Fly Calculation of Unresolved Resonance Region Cross Sections in Monte Carlo Simulations," presented at the ANS MC2015—Joint Int. Conf. Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA), and the Monte Carlo (MC) Method, Nashville, Tennessee, April 19–23, 2015.
28. "International Handbook of Evaluated Criticality Safety Benchmark Experiments," NEA/NSC/DOC(95)03, (2009), Organisation for Economic Co-operation and Development/Nuclear Energy Agency Nuclear Science Committee.
29. C. C. BYERS et al., "Critical Measurements of a Water-Reflected Enriched Uranium Sphere," *Trans. Am. Nucl. Soc.*, **27**, 412 (1977).
30. "NAPP Physics Design Manual," Nuclear Power Corporation of India Limited (NPCIL) (1987).
31. P. D. KRISHNANI, "CLUB—A Multigroup Integral Transport Theory Code for Analysis of Cluster Lattices," *Ann. Nucl. Energy*, **9**, 255 (1982); https://doi.org/10.1016/0306-4549(82)90029-9.
32. P. D. KRISHNANI, "Input Preparation for Computer Code ITRAN," Technical Report BARC/RPDD/RP/011, Bhabha Atomic Research Centre (June 23, 2010).