NUCLEAR SCIENCE AND ENGINEERING: **174**, 318–325 (2013)

Modeling of Nonuniform Density Distributions in the Serpent 2 Monte Carlo Code

Jaakko Leppänen*

VTT Technical Research Centre of Finland

Received June 29, 2012
Accepted November 16, 2012

Abstract – *This paper presents a methodology for applying continuously varying density distributions in Monte Carlo particle transport simulation. The capability is implemented in the Serpent 2 code, as part of an effort for developing a universal multiphysics interface for the coupling of Monte Carlo neutronics to thermal hydraulics and fuel performance codes. The method is based on rejection sampling of particle path lengths, but despite its close resemblance to the Woodcock delta-tracking method, the routine can be used with conventional surface tracking as well. The modified tracking routine is put to the test in a simple boiling water reactor pin-cell calculation with continuously changing void distribution in the coolant channel.*

I. INTRODUCTION

The Serpent Monte Carlo code[a] is currently being rewritten for the purpose of improving memory usage and parallelization routines and to extend the built-in burnup calculation capability from two-dimensional (2-D) lattice physics applications to three-dimensional (3-D) full-core problems.[1] The new code version, Serpent 2, also forms the platform for developing new features, such as photon transport capability and multiphysics coupling. The beta version of Serpent 2 was made available to licensed users in January 2012, and the public release is scheduled for 2013 or 2014.

This paper presents some of the preliminary work carried out for the development of a universal multiphysics interface that will eventually be used for coupling Serpent to other codes, such as computational fluid dynamics (CFD) codes, thermal hydraulics system codes, and fuel performance codes. The coupling is to be handled via a sequential exchange of input and output files, and material temperature and density distributions are passed into the transport simulation without any modifications in the actual geometry model. Temperature dependence involves the Doppler broadening of microscopic nuclidewise cross sections, and the capability to model nonuniform temperature distributions is a major research topic in Serpent development.[2,3] This study is focused on density effects and macroscopic cross sections, and the application considered in the context is the continuously varying distribution of coolant density or void fraction along the flow channel of a nuclear fuel assembly.

The developed methods are based on a rejection sampling routine, and the underlying theory is introduced in Sec. II. Practical implementation in the multiphysics interface is discussed in Sec. III. The final format of the interface itself is still under consideration, and the coupling lacks the capability to pass power distributions back to the thermal hydraulics solver. For this reason the introduced methodology is demonstrated in Sec. IV by a simple stand-alone test case involving a parabolic axial void distribution in a boiling water reactor (BWR) coolant channel. Section V is left for the conclusions.

II. THEORY

The possibility of handling nonuniform material compositions in Monte Carlo particle transport simulation has been studied at least by Carter et al.[4] and Brown and

*E-mail: Jaakko.Leppanen@vtt.fi

[a]For a complete and up-to-date description of the Serpent Monte Carlo code, see the Web site http://montecarlo.vtt.fi (current as of June 29, 2012).

Martin,[5] and it is generally recognized as one of the characteristic features of the delta-tracking method, developed by Woodcock et al. in the 1960s (Ref. 6). What is perhaps not as well understood, however, is the fact that this capability is not unique to the Woodcock method but rather results from two factors:

1. rejection sampling performed for the selection of particle path lengths
2. the use of the collision estimator of neutron flux (CFE) for calculating integral reaction rates.

Neither of these two factors necessarily implies the use of delta tracking, at least in the form introduced by Woodcock, even though the procedure is very similar to it.[b] The theoretical background for the method is discussed in the following.

II.A. Sampling Procedure

The total interaction probability P per traveled particle path length s is characterized by the material total cross section:

$$\frac{dP}{ds} = \Sigma_t(\mathbf{r}) \ . \tag{1}$$

From this definition it results that the probability that the particle encounters its first interaction at a distance s from an arbitrary starting position is written as[7]

$$f(s) = \Sigma_t e^{-s\Sigma_t} \ , \tag{2}$$

assuming that the material is homogeneous over interval $[0, s]$. Equation (2) is the probability density function (PDF) of the free particle path length, and the corresponding cumulative distribution function (CDF) is given by integration:

$$F(s) = \int_0^s \Sigma_t e^{-s'\Sigma_t} \, ds' = 1 - e^{-s\Sigma_t} \ . \tag{3}$$

The interpretation of the CDF is that it gives the probability that the particle reaches distance s without an interaction. This result can be used for sampling the distance to the next collision site in the random walk, by applying the inversion method[7]:

$$s = F^{-1}(\xi) \Rightarrow s = -\frac{1}{\Sigma_t} \ln(\xi) \ , \tag{4}$$

where ξ is a uniformly distributed random variable on the unit interval.

The geometries in Monte Carlo particle transport applications are generally not homogeneous, but consist of homogeneous arbitrarily shaped material regions. This implies that Σ_t is not a constant, but rather a piecewise constant function of the spatial coordinates. Consequently, the integration over particle path must be divided into parts, where the points of discontinuity are given by the tracking routine that calculates the distance to the nearest material boundary within the line of sight. In practice, however, particle path lengths are sampled from the single-material distribution functions according to Eq. (4), and if the collision point is located outside the material boundaries, the track is stopped at the boundary surface, and a new path length is sampled using the total cross section of the material that fills the next cell.

The fact that Σ_t is assumed to be not a piecewise constant, but actually an arbitrary continuous function of the spatial coordinates, changes the nature of the tracking routine completely. Equation (1) has no general solution, and distributions (2) and (3) become unknown functions, which depend on the spatial variation of the total interaction probability. Consequently, the inversion method, and the well-known formula (4), can no longer be applied for sampling the particle path lengths. It is possible to derive similar direct procedures for certain types of analytical distribution functions, as was done by Brown and Martin,[5] but a general case with arbitrary spatial dependence requires a different approach.

The method used in the Serpent tracking routine relies on rejection sampling,[7] which is commonly used for sampling random variables from complicated probability distributions. The method is based on the use of two distribution functions: the original PDF and a majorant function satisfying

$$f_{maj}(z) \geq f(z) \tag{5}$$

for all values of the random variable z, and formed in such a way that values from f_{maj} can be sampled using the inversion method. According to the theory, values sampled from the majorant function and accepted with probability

$$P = \frac{f(z)}{f_{maj}(z)} \tag{6}$$

follow the distribution of f. Rejected values are discarded, and the procedure is repeated until a successful sample is obtained from Eq. (6).

In this case the majorant function is the PDF corresponding to the maximum total cross section $\Sigma_{\max}$ in the material region,

$$\Sigma_{\max} \geq \Sigma_t(\mathbf{r}) \ , \tag{7}$$

[b] What exactly is considered "delta tracking" is a matter subject to some interpretation. In this context the term is reserved for the geometry-tracking routine introduced in the original paper by Woodcock et al.[6] The difference with the method developed for Serpent 2 is emphasized to point out the fact that the sampling routine can be used with conventional surface tracking as well. Comparison of the two methods is discussed in Sec. II.B.

and the rejection sampling is carried out by comparing the maximum and the actual cross section at the collision point. In practice, the path length is sampled using Eq. (4),

$$s = -\frac{1}{\Sigma_{\max}} \ln(\xi_1) \ , \tag{8}$$

and the collision point is accepted if a second uniformly distributed random variable ξ_2 satisfies condition

$$\xi_2 < \frac{\Sigma_t(\mathbf{r}')}{\Sigma_{\max}} \ , \tag{9}$$

where

$$\mathbf{r}' = \mathbf{r} + s\hat{\mathbf{\Omega}} \tag{10}$$

and where $\mathbf{r}$ and $\hat{\mathbf{\Omega}}$ are the initial position and the direction vector, respectively. If the point is rejected, the whole procedure restarts by sampling a new path length starting from the new position $\mathbf{r}'$.

The fact that the maximum cross section $\Sigma_{\max}$ in Eq. (8) is constant over the region implies that the total interaction probability can be an arbitrary function of the spatial coordinates. The actual value is needed only for performing the rejection sampling Eq. (9) at the tentative collision sites. This capability is put to practice in Sec. III.

II.B. Comparison to Woodcock's Original Method

The introduced sampling scheme is easily confused with the Woodcock delta-tracking method,[6] which is also based on rejection sampling. To understand the difference, it is important to realize that the procedure still involves sampling collision points inside bounded material regions. If a material boundary is crossed, the sampled path length is no longer statistically valid. The particle track must be stopped at the boundary surface and a new path length sampled corresponding to the new interaction probability, as is done in conventional surface tracking.

In Woodcock's method, on the other hand, the rejection sampling is applied for the purpose of extending the sampled path lengths over one or several material boundaries, without calculating the distances to the boundary surfaces. The geometry is assumed to consist of arbitrarily shaped homogeneous material regions, and the method can essentially be interpreted as a rejection technique used for sampling path lengths from a piecewise continuous PDF of interaction probability, in which the points of discontinuity are not known.

In practice, the difference is seen in the way the maximum cross section $\Sigma_{\max}$ is formed. In Woodcock's original delta-tracking scheme, this cross section is given by the maximum of all materials in the geometry, while in the introduced method it is the maximum within a single material region. What makes things a bit more complicated is the fact that these two methods, as well as conventional surface tracking, can also be combined, as is done in the tracking routine implemented in Serpent 2 (see Sec. III).

II.C. Limitations

In addition to rejection sampling, delta tracking and the introduced method share certain limitations regarding the use of the track-length estimate (TLE) of neutron flux for calculating reaction rate integrals. The delta-tracking routine cannot tell where or when a boundary surface has been crossed, which means that the track lengths are not available for calculating cell flux tallies. When used with surface tracking, the introduced method does have access to track lengths inside cells, but the estimator itself cannot be used with cross sections that are not constant over the sampled path. The result is that the TLE can be used for calculating integral flux, but not local reaction rates inside the cell.

The alternative method for calculating flux and reaction rate integrals is the CFE, which is scored when the sampled path length terminates in an interaction. Since the number of collisions is always less than or equal to the number of tracks, the CFE generally results in inferior statistics. This is especially the case in small or optically thin cells and regions of low collision density. Practical applications with the Serpent code[8] have shown, however, that the difference in efficiency between the two estimators is not that significant in reactor physics applications, such as burnup calculation and the generation of homogenized multigroup constants. This is because of two reasons:

1. Reaction rate integrals are typically calculated in the same part of the geometry as where neutrons are born, which ensures a large number of scores in the region of interest.

2. Collision density is high in both the fuel and the moderator, and the number of collisions is comparable to the number of tracks.

For the same reasons the use of the CFE is not considered a major limitation for multiphysics applications either, where the main parameter of interest is the power distribution in the fuel.

Another well-known drawback of delta tracking is the fact that the efficiency of the rejection sampling routine becomes poor when the maximum cross section exceeds the material total by several orders of magnitude. This happens, for example, in the presence of localized heavy absorbers, such as control rods or burnable absorber pins. In the geometry-tracking routine, the problem is easily avoided by switching to surface tracking when the efficiency of delta tracking becomes poor.[8] A similar approach cannot be applied when the rejection

method is used for modeling nonuniform material compositions, and whether or not this becomes a problem depends on how much the material total cross section varies inside the region.

III. IMPLEMENTATION IN SERPENT 2

The method was implemented in Serpent 2.1.3. The tracking routine was already based on the combination of surface and delta tracking,[8] with reaction rate integrals calculated using the CFE. Consequently, the only modification that was required for implementing the methodology was an additional rejection test, performed according to Eq. (9) before sampling the reaction mode. The implementation to Monte Carlo codes based exclusively on surface tracking and TLE may require more extensive modifications, which are not discussed in this paper. The remainder of Sec. III is instead focused on how the spatially dependent total cross section is determined at the collision sites.

III.A. Dependence on Material Density

In Sec. II it was assumed that the material total cross section is an arbitrary function of the spatial coordinates. From here on this study is focused on changes caused by variation in material density, assuming that the isotope vector remains constant, which is the case when the changes are caused by heat expansion or boiling—two factors relevant for multiphysics applications.[c] Temperature dependence of microscopic cross sections is another topic, covered in Refs. 2 and 3.

Since macroscopic cross sections are proportional to atomic density, the spatial dependence can be written in the form

$$\Sigma_t(\mathbf{r}) = g(\mathbf{r})\Sigma_{\max} \quad , \tag{11}$$

where g is a spatially dependent density factor:

$$g(\mathbf{r}) = \frac{n(\mathbf{r})}{n_{\max}} = \frac{\rho(\mathbf{r})}{\rho_{\max}} \quad . \tag{12}$$

The fact that the ratio of atomic densities n is equal to the ratio of mass densities ρ results from the assumption of constant isotopic composition. The maximum density $\rho_{\max}$, which is also used for determining the maximum cross section $\Sigma_{\max}$, is taken from the geometry definition in the Serpent input file. The spatial density distribution is defined in the multiphysics interface, which currently provides three options:

1. piecewise constant distribution on a regular mesh
2. weighted average of pointwise values
3. user-specified functional dependence.

Each distribution operates on a single material, and multiple distributions can be applied simultaneously, which enables, for example, different density distributions for coolant and moderator channels in a BWR assembly calculation. As mentioned in Sec. I, the distributions are given in a separate interface file, without any modifications in the actual geometry model.

III.B. Interface Type 1: Piecewise Constant Distribution on a Regular Mesh

The difference between the three methods is in the way the density factor is determined based on the spatial coordinates. In the first method, the distribution is given on a Cartesian one-dimensional (1-D), 2-D, or 3-D mesh. Each mesh cell is assigned a density, which is uniform over the cell volume. Apart from a simplified user interface, this option offers nothing new to Monte Carlo calculation, as the same result can be achieved by explicit definition of homogeneous material regions at different densities. It is also the only distribution type that can be used for validating the methodology, since comparable reference results cannot be produced for continuous distributions.

III.C. Interface Type 2: Weighted Average of Pointwise Values

The second method enables the definition of nonuniform and truly continuous distributions of material density. The interface file contains a set of pointwise values g_i, which are used to calculate weighted averages at an arbitrary position $\mathbf{r}$:

$$g(\mathbf{r}) = \frac{\sum_i g_i(\mathbf{r} - \mathbf{r}_i)^n}{\sum_i (\mathbf{r} - \mathbf{r}_i)^n} \quad , \tag{13}$$

where exponent n is a user-defined order of the weighting factors. It should be noted that the same methodology can be applied for several types of kernel functions, and Eq. (13) is only one example. The comparison of different options was intentionally left for future studies, involving coupled neutronics and thermal hydraulics calculations and more realistic 3-D density distributions.

The distribution may involve thousands of data points, and averaging over all of them at every collision point requires considerable CPU time. The computational effort can be significantly reduced by excluding points beyond a specified radius r_{ex} from the sum at each

[c] Nonuniform distributions of isotopic composition could be caused by mixing and dilution of materials in liquid form, and boron dilution transients, for example, play a significant role in pressurized water reactor safety analyses. Because of the complex nature of the process, it was decided to leave the topic for future studies and first focus on changes in material density.

collision point. This exclusion radius is one of the input parameters in the interface file. The calculation of local average involves searching through the data to find the points located inside the exclusion radius. To speed up the procedure, the coordinates are stored on a pregenerated search mesh. Instead of looping through all points, the search algorithm needs to calculate the distance only to points located in mesh cells that intersect the exclusion radius.

III.D. Interface Type 3: User-Specified Functional Dependence

The third format involves a user-specified dependence on the spatial coordinates. The interface file provides a set of input parameters that are passed into a blank subroutine, along with the coordinates at each collision point. This interface type can be considered the most versatile option for representing the density distribution, as the user can implement any kind of spatial distribution for the density factor.

IV. TEST CASE

The main application for the developed method is the modeling of a continuous coolant density or void distribution in the flow channel of a nuclear fuel assembly. The multiphysics interface in Serpent 2 is still under development, and the calculation system lacks the capability to perform calculations that are coupled in both directions. The new methodology is therefore tested without actual coupling, by supplying a given coolant density distribution to Serpent and comparing the resulting thermal flux and power distributions to reference results. The format that is to be used for passing the results from the Monte Carlo simulation back to the thermal hydraulics code is not considered here.

The geometry in the test case is a 370-cm-high axially finite and radially infinite BWR pin-cell model, with axial coolant void distribution characterized by a density factor with functional dependence:

$$g(z) = \begin{cases} 1.0 & \text{for } z < 50 \\ 1 - 0.9\left(\dfrac{z-50}{320}\right)^2 & \text{for } z \geq 50 \end{cases} . \qquad (14)$$

Reflectors and all other structures above and below the active fuel height are omitted for simplicity. The functional dependence is applied directly with the type 3 interface described in Sec. III. The pointwise distribution (type 2) is given using 11 discrete values, uniformly spaced over the active height. The interpolation order is set to 1, and the exclusion radius to match the distance between two points. The overall result is a 1-D piecewise linear function for the density factor. The values for the Cartesian-mesh distribution (type 1) are calculated by averaging Eq. (14) over ten cells with equal height. The dependence seen by the Serpent tracking routine is plotted in Fig. 1 for each distribution type. The main geometry and material parameters are summarized in Table I.

It should be noted that the simplified density distribution used in the geometry model is not based on any actual heat transfer correlation, and it is not intended to yield a realistic description of boiling in a BWR coolant channel, but to provide a test case for the developed methodology. Since it is not possible to generate exact results for the continuous distributions, the validation was done for the piecewise homogeneous mesh-based distribution. The results were compared to a reference calculation, in which the coolant channel was explicitly divided into an equal number of axial cells, each assigned a different density. The results for the other two

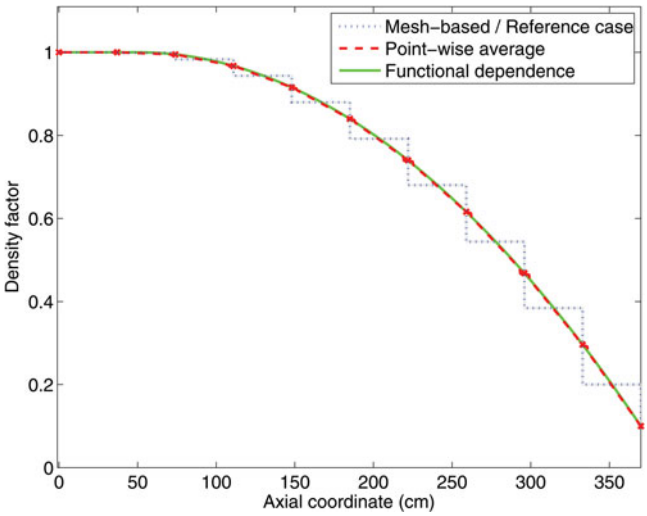

Fig. 1. Density factor as a function of axial coordinate.

TABLE I

Geometry and Material Parameters in the Test Case*

Parameter	Value
Fuel	UO_2 enriched to 3.4 wt% ^{235}U
Pellet radius	0.4335 cm
Cladding outer radius	0.5025 cm
Pin pitch	1.3 cm
Active height	370 cm
Homogenized coolant density	0.7396, 0.7388, 0.7271, 0.6978, 0.6506, 0.5857, 0.5030, 0.4024, 0.2841 and 0.1480 g/cm^3

*The coolant channel is divided into ten axial zones with different densities.

distributions are included, but they are expected to differ significantly from the reference calculation.

Thermal flux in the coolant and fission power in the fuel were calculated using standard detectors (tallies) in the Serpent code. The geometry model was divided into 200 axial bins, and the boundary between thermal and fast flux was set to 0.625 eV. The calculation was run using JEFF-3.1–based cross-section libraries, with 200 inactive and 10 000 active cycles of 200 000 source neutrons, totaling 2 billion active neutron histories. The results were collected in 50 batches of 200 cycles each, in an effort to reduce the impact of cycle-to-cycle correlations on statistical errors. Source convergence was confirmed by monitoring fission source entropy, which in Serpent can be calculated separately in the *x*-, *y*-, and *z*-directions. All calculations were run in full OpenMP parallel mode on a 12-core, 3.47-GHz Intel Xeon workstation.

IV.A. Results

The results are plotted in Figs. 2 through 5.[d] As expected, the thermal flux distribution peaks at the lower half of the coolant channel, where water density is at maximum. Fission power follows closely the distribution of thermal flux. Comparison of different models shows that the Cartesian-mesh-type interface reproduces the reference results to within statistics.[e] The other two interface types, based on continuous distributions, result in clear differences compared to the piecewise homogeneous models. The oscillatory behavior is caused by the fact that water density, compared to the reference calculation, is first overestimated, then underestimated, inside each homogeneous region when moving upward in the coolant channel (see Fig. 1).

Effective multiplication factors are given in Table II. As expected, the result of the mesh-based distribution model matches the reference calculation. What is a bit surprising, however, is that the other two models yield very consistent values as well. This can be considered a reminder of the fact that an integral parameter like k_{eff} does not necessarily reflect the local differences in flux and reaction rates, which in this case increase to several

[d]The distributions are plotted as smooth curves since the 1.85-cm tally segments are too short to produce a clear stair-step plot.

[e]It is not possible to attain perfect reproducibility because the application of the density factor involves rejection sampling, and the random-number sequence is not preserved.

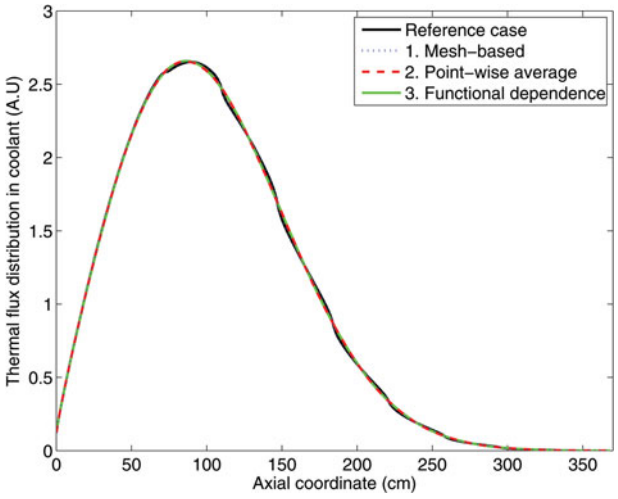

Fig. 2. Thermal flux distribution in the coolant channel as a function of axial coordinate.

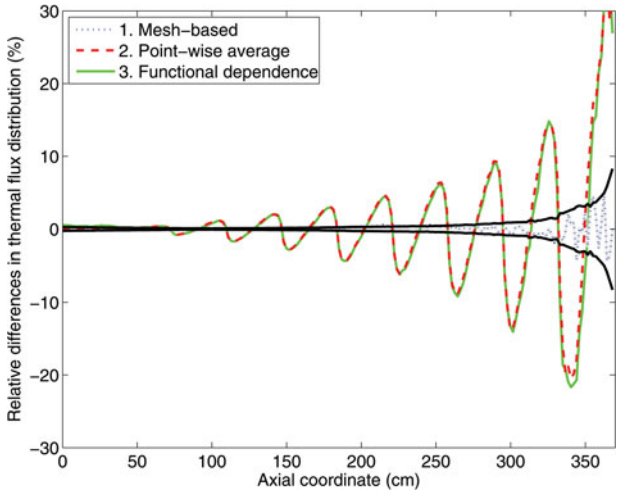

Fig. 3. Relative differences in thermal flux distribution compared to the reference case. The solid black lines show the 95% confidence interval of the reference result.

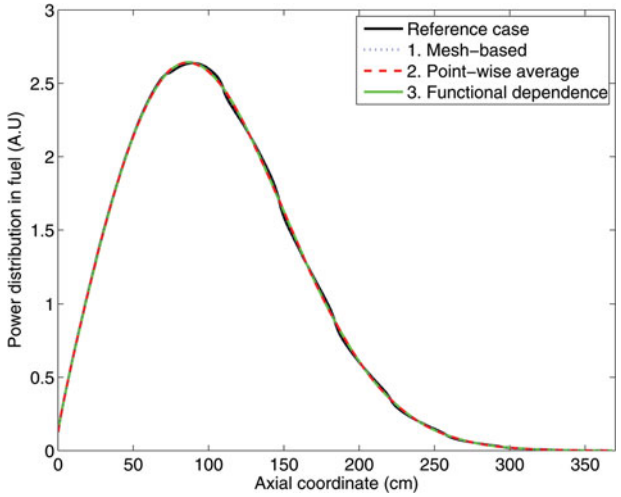

Fig. 4. Power distribution in the fuel as a function of axial coordinate.

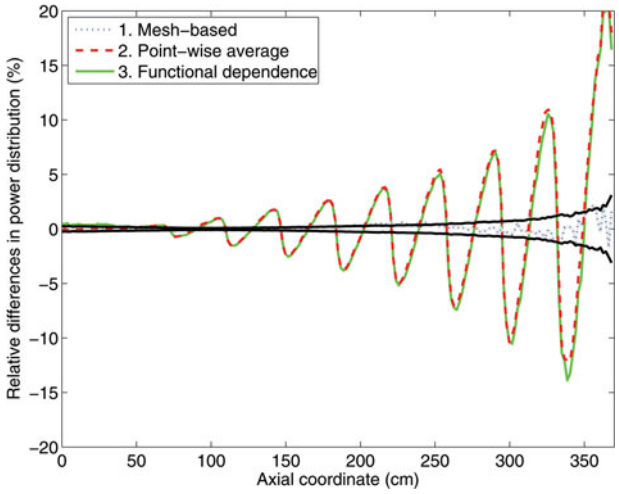

Fig. 5. Relative differences in power distribution compared to the reference case. The solid black lines show the 95% confidence interval of the reference result.

TABLE II

Criticality Eigenvalues and Running Times for Each Calculation Case

Case/Model	k_{eff}	Time (min)
Reference case	1.30657 ± 0.00001	335.4
1. Mesh-based	1.30657 ± 0.00001	367.7
2. Pointwise average	1.30656 ± 0.00001	438.0
3. Functional dependence	1.30695 ± 0.00001	357.5

percent, even in the region where the absolute values are high.

The comparison of running times in Table II shows that the modified transport routine does not result in a significant change in calculation time, except for distribution type 2, in which additional work is required for calculating the distances between each collision point and the data points inside the exclusion radius. To some extent this result can be contributed to the use of delta tracking, which in the reference calculation performs the same type of rejection sampling for the material regions at reduced density as the new model does for the density factor. The overall count of accepted and rejected collisions is therefore roughly the same.[f]

The efficiency of the rejection sampling routine was 71% in all cases. According to previous studies,[8] there is no significant deterioration in the performance of the Serpent tracking routine until the efficiency falls below ~30%. As mentioned earlier, the methodology can also be used with conventional surface tracking, in which case the rejected collisions may increase the calculation time to some extent. In the present test case, such behavior was not observed.

[f]The small overhead for the mesh-based distribution compared to the reference case results from the fact that the rejection sampling is performed when sampling the target nuclide, while in delta tracking it is done before calling the collision routine.

IV.B. Discussion

The main purpose of this study was to demonstrate that nonuniform density distributions can be accurately modeled in continuous-energy Monte Carlo particle transport simulation. This capability, however, is only one part of the solution. An equally important task is the generation of these distributions, taking into account the physical boundary conditions and source terms. There exist elaborate calculation tools, based on, for example, CFD, that can accomplish this task at high fidelity. The challenge that remains for the Monte Carlo code is the interpretation of the thermal hydraulics data.

Communication between Serpent 2 and the externally coupled thermal hydraulics solver is to be handled via a universal multiphysics interface. The work is by no means complete, and the three preliminary methods introduced in this paper are all subject to limitations. The Cartesian-mesh distribution does not fit particularly well in complicated 3-D geometries, and other regular mesh types (e.g., cylindrical) are not doing a much better job, either. The user-defined distribution function has much potential for specific problems where the user has extensive knowledge of the physics of coolant flow, but it cannot really be considered a universal solution.

Of the introduced methods the pointwise approach can be considered the most general way of passing 3-D density distributions from thermal hydraulics codes to the Monte Carlo tracking routine. The format is not limited to any particular solution method, and the resolution can easily be adjusted by varying the density of the data points. There are some issues, however, related to, for example, the selection of the exclusion radius. If the value is set too small, there may be parts of the geometry where not a single point falls inside the radius, and consequently, there is no information on the local density. If the radius is too large, an excessive number of points are included in the average, which can significantly increase the running time without major contribution in the results. It should also be noted that the distribution seen by the tracking routine is only a representation of the actual thermal hydraulics solution, so the Monte Carlo code is essentially working with a slightly different problem. Whether these issues are major or minor limitations is left for future studies.

When considering the coupling to CFD codes, the most obvious solution would be to pass the density distribution to the Monte Carlo calculation using the same

unstructured mesh that was used for modeling fluid flow. Mesh generation for CFD codes is a complicated procedure, but the mesh structure itself is relatively simple, and it should be directly applicable for the task. The main advantages of this approach are that both the neutronics solution and the thermal hydraulics solution are based on the same setup, the distribution can be modeled in high resolution where needed, and the same mesh structure can be used for passing results in both directions.

V. SUMMARY AND CONCLUSIONS

This paper presented some preliminary work, carried out for the development of a universal multiphysics interface, to be used for coupling the Serpent 2 Monte Carlo code to thermal hydraulics and fuel performance codes. The developed methodology enables the modeling of coolant density distributions separate from the actual geometry input, which considerably simplifies the external coupling of the two solutions. The implementation is based on a rejection sampling routine, which enables the modeling of continuously varying distributions. Similar methodology is currently being developed for material temperatures.[2,3]

The development work is an ongoing process, and the currently available options for passing the density data into the tracking routine include Cartesian-mesh-based, pointwise, and user-defined functional distributions. The methodology was validated using a simple BWR pin-cell model with axially varying coolant void distribution. The test case can be considered a proof of concept, but the practical applicability of the interface still needs to be verified on a realistic 3-D geometry. Other topics for future work include the implementation of an unstructured-mesh-based interface for coupling with CFD codes as well as methods for passing the power distribution from the Monte Carlo simulation back to the thermal hydraulics solver.

ACKNOWLEDGMENTS

This work has been funded by the European Union High Performance Monte Carlo Reactor Core Analysis (HPMC) project and the KÄÄRME project under the Finnish National Research Programme on Nuclear Power Plant Safety, SAFIR-2014.

REFERENCES

1. J. LEPPÄNEN and A. ISOTALO, "Burnup Calculation Methodology in the Serpent 2 Monte Carlo Code," *Proc. PHYSOR 2012*, Knoxville, Tennessee, April 15–20, 2012, American Nuclear Society (2012).

2. T. VIITANEN and J. LEPPÄNEN, "Explicit Treatment of Thermal Motion in Continuous-Energy Monte Carlo Tracking Routines," *Nucl. Sci. Eng.*, **171**, 165 (2012).

3. T. VIITANEN and J. LEPPÄNEN, "Explicit Temperature Treatment in Monte Carlo Neutron Tracking Routines—First Results," *Proc. PHYSOR 2012*, Knoxville, Tennessee, April 15–20, 2012, American Nuclear Society (2012).

4. L. L. CARTER, E. D. CASHWELL, and W. M. TAYLOR, "Monte Carlo Sampling with Continuously Varying Cross Sections Along Flight Paths," *Nucl. Sci. Eng.*, **48**, 403 (1972).

5. F. B. BROWN and W. R. MARTIN, "Direct Sampling of Monte Carlo Flight Paths in Media with Continuously Varying Cross Sections," LA-UR-02-6430, Los Alamos National Laboratory (2002).

6. E. R. WOODCOCK et al., "Techniques Used in the GEM Code for Monte Carlo Neutronics Calculations in Reactors and Other Systems of Complex Geometry," ANL-7050, Argonne National Laboratory (1965).

7. I. LUX and L. KOBLINGER, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Inc. (1991).

8. J. LEPPÄNEN, "Performance of Woodcock Delta-Tracking in Lattice Physics Applications Using the Serpent Monte Carlo Reactor Physics Burnup Calculation Code," *Ann. Nucl. Energy*, **37**, 715 (2009).