Application of the Full 3-D Collision Probability Method to Randomly Distributed Spherical Fuel Elements

M-A. Lajoie, G. Marleau
École Polytechnique de Montréal, Montréal, Québec, Canada

Abstract

The analysis of VHTR fuel tends to be difficult when using deterministic methods currently employed in lattice codes notably because of limitations on geometry representation and the stochastic positioning of spherical elements. The method proposed here and implemented in the lattice code DRAGON is to generate the positions of multi-layered spheres using random sequential addition, and to analyze the resulting geometry using a full three-dimensional spherical collision probability method. The preliminary validation runs are consistent with results obtained using a Monte-Carlo method, for both regularly and randomly positioned pins.

1. Introduction

Fourth generation very high temperature reactors (VHTRs) present characteristics that tend to be harder than conventional geometries to analyse using methods available in traditional deterministic neutronics computer codes. Worthy of interest in the scope of this document are the random character of fuel elements in a uniform matrix, and the multi-layered, spherical geometry of these fuel elements.

The goal of this project has been to integrate the capacity to analyse such properties in the transport lattice code DRAGON [1]. First, a geometrical analysis of spherical volumes has allowed us to develop analytical volumes calculation capacities. Then, integration line tracking for spherical geometries has been implemented. Routines enabling this exact geometrical analysis of multi-layered, randomly disposed spherical particles have been successfully incorporated and tested in this code. A thorough analysis of combinations of spherical and Cartesian meshes has also been performed, to ensure the validity of the lengths and volumes calculated by the new routines.

Finally, preliminary validation runs have been carried out, comparing results obtained using the new module on simple, three dimensional, spherical geometry elements, using both explicitly positioned pins and randomly distributed fuel elements. Results have been compared and are consistent with those established using the Monte-Carlo code SERPENT [2]. Increasing angular and spatial tracking refinement leads to better precision, however at the expense of a rapidly increasing computational burden.

This paper will begin by presenting a quick theoretical review of the collision probability method, followed by the geometrical analysis required for the analytic volumes calculation and generation of the tracking lines. Then, the verification process will be presented, followed by the results of the preliminary validation performed on a simple random distribution of spherical fuel elements.

2. Theory

The neutron transport can be solved using a number of methods. Pertinent to this analysis, we use the integral form of the Boltzmann transport equation, which generally requires a spatial discretization analysis known as tracking. Using the formalism presented by Hébert [3], the integral form of the multi-group, steady-state transport equation can be written, assuming isotropic sources and scattering, as:

$$\phi_g(\vec{r}) = \frac{1}{4\pi}\int_{4\pi}\int_0^{\infty} e^{-\tau_g(s)} Q_g(\vec{r} - s\hat{\Omega}) ds\, d^2\Omega \tag{1}$$

where ϕ_g is the neutron flux at position $\vec{r}$ in group g, τ_g is the optical path defined as:

$$\tau_g(s) = \int_0^s \Sigma_g(\vec{r} - s'\hat{\Omega}) ds' \tag{2}$$

with Σ_g the transport corrected total cross-section. The isotropic neutron source in group g, Q_g, can be expressed as a sum of fission-produced and scattered neutrons:

$$Q_g(\vec{r}) = \sum_{h=1}^{G} \Sigma_{s,0,g\leftarrow h}(\vec{r})\phi_h + \frac{1}{k_{eff}} \sum_{j=1}^{J} \chi_{j,g} \sum_{g=1}^{G} \nu\Sigma_{f,j,h}(\vec{r})\phi_h(\vec{r}) \tag{3}$$

Here $\Sigma_{s,0,g\leftarrow h}$ is the transport corrected scattering cross section, $\nu\Sigma_{f,j,h}$ the neutron production from fission by isotope j, and $\chi_{j,g}$ the fission spectrum.

2.1 The collision probability method

The collision probability method relies on a spatial discretization of this equation, requiring the subdivision of the unit cell in N sub-volumes, each with volume V_i, $i = 1 \ldots N$. Each of the sub-regions will be assumed to contain a source $Q_{i,g}$ that is uniform over its volume, as well as a uniform total macroscopic cross-section $\Sigma_{i,g}$. Using the averaged flux in region V_i:

$$\phi_{i,g} = \frac{1}{V_i}\int_{V_i} \phi_g(\vec{r}) d^3r \tag{4}$$

we can reformulate equation (1), after multiplication by $\Sigma_{i,g}$ and integration over V_j, as:

$$\phi_{i,g} = \sum_{j=1}^{J} Q_{j,g}\, p_{ij,g} \tag{5}$$

where, using s as an index representing the distance travelled on a track in direction $\hat{\Omega}$, we have defined the reduced collision probability $p_{ij,g}$, as:

$$p_{ij,g} = \frac{1}{4\pi V_i}\int_{V_j}\int_{4\pi}\int_{-\infty}^{\infty} \Im_{V_i}(s,\hat{\Omega}) e^{-\tau_g(s)} ds d^2\Omega d^3r \tag{6}$$

with $\Im_{V_i}(s,\hat{\Omega})$ a characteristic function worth 1 if point *s* of track trajectory $\hat{\Omega}$ is in volume V_i, and 0 otherwise. This reduced collision probability is defined in such a way that it remains finite as $\Sigma_{j,g}$ tends to zero, and so that $\Sigma_{j,g} p_{ij,g}$ is the probability for a neutron born in region *i*, in energy group *g*, to have its first collision in region *j*.

2.2 Tracking

An angular discretization is then performed, where planes $\Pi_{\hat{\Omega}}$ are defined orthogonal to direction $\hat{\Omega}$. Each plane is characterized by orthogonal lines (*tracks*), each with a starting point $\vec{p}'$ on plane $\Pi_{\hat{\Omega}}$, and propagating in direction $\hat{\Omega}$. This allows us to rewrite the volume integrals as integrals over the whole tracking as:

$$\int_{4\pi}\int_{V_j} F(\vec{r},\hat{\Omega})d^3rd^2\Omega = \int_{4\pi}\int_{\Pi_{\hat{\Omega}}}\int_{-\infty}^{\infty} \Im_{V_j}(s',\hat{\Omega})F(s',\hat{\Omega})ds'd^2p'd^2\Omega \quad (7)$$

We can therefore evaluate the volumes and reduced collision probabilities as:

$$V_{j,\hat{\Omega}} = \int_{\Pi_{\hat{\Omega}}}\int_{-\infty}^{\infty} \Im_{V_j}(s,\hat{\Omega})ds'd^2p' \quad (8)$$

$$p_{ij,g} = \frac{1}{4\pi V_i}\int_{4\pi}\int_{-\infty}^{\infty}\left[\int_{\Pi_{\hat{\Omega}}}\int_{-\infty}^{\infty} \Im_{V_j}(s,\hat{\Omega})\Im_{V_i}(s',\hat{\Omega})e^{-\tau_g}ds'd^2p'\right]dsd^2\Omega \quad (9)$$

In practice, since an analytic integration cannot be performed, the tracking of the geometry will be done by supposing a finite number of directions $\hat{\Omega}_m$, each with an associated weight ω_m, and a surface track density on every plane, each track also being weighted with weight T_n, and crossing regions identified by indices *k* or *l*. We can then evaluate numerically equations (8) and (9) as:

$$V_{j,m} \cong \sum_n T_n \sum_k \delta_{k,j} \quad (10)$$

$$p_{ij,g} \cong \frac{1}{\Sigma_i \Sigma_j V_i}\sum_m \omega_m \sum_n T_n \sum_k \delta_{k,i} \sum_l \delta_{h,j} e^{-\tau_{k,l}}\left[1-e^{-\Sigma_i L_k}\right]\times\left[1-e^{\Sigma_j L_h}\right] \quad (11)$$

3. Geometry Analysis

The implementation in DRAGON of the capacity for the analysis of multi-layered spherical particles has two components. First, the tracking itself, which is the main component, although mathematically quite trivial, and second, the analytical volumes calculation, which serves as a tracking verification, by comparing analytical and numerical volumes, computed using equation (8). Both steps are performed by the module NXT: [4].

3.1 Spherical 3-D tracking

The NXT: module generates a number of tracking lines, based on user-specified track density and angular quadrature, each line being characterized by its starting point (x_l, y_l, z_l), and its orientation, or direction cosines (a_x, a_y, a_z). Each sphere is also characterized by the coordinates of its centre (x_s, y_s, z_s), and its radius, R. Hence, we have the following equations, respectively for the line and the sphere:

$$\frac{x - x_l}{a_x} = \frac{y - y_l}{a_y} = \frac{z - z_l}{a_z} = t \tag{12}$$

$$(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2 = R^2 \tag{13}$$

These can easily be solved for the intersections coordinates $(x_1, y_1, z_1) = (t_1 a_x + x_l, t_1 a_y + y_l, t_1 a_z + z_l)$ and $(x_2, y_2, z_2) = (t_2 a_x + x_l, t_2 a_y + y_l, t_2 a_z + z_l)$, where t_1 and t_2 are the roots of the quadratic equation $(a_x t + x_l - x_s)^2 + (a_y t + y_l - y_s)^2 + (a_z t + z_l - z_s)^2 = R^2$

3.2 Volumes

The volumes that need to be computed here are the volumes of spherical shells (which are trivial) and those corresponding to the intersection of a Cartesian and spherical region. The methodology employed for the latter relies on a linear combination of the volumes of the intersection located *below*, to the *left* and *behind* each corner of the parallelepiped formed by the intersection of the six planes delimiting the Cartesian region. Using indices + and - to identify the two planes relative to each axis, Figure 1 presents a 2-D analogy of this method, where we can calculate the area B, as a sum of surfaces $S_{[+,+]}$= A+B+C+D, $S_{[+,-]}$= C+D, $S_{[-,+]}$= A+C and $S_{[-,-]}$= C, so that B = $V_{[+,+]}$ - $V_{[+,-]}$ - $V_{[-,+]}$ + $V_{[-,-]}$

Extending this analogy to the three-dimensional problem, the volume of the total intersection will be given by:

$$V_{Inter} = V_{[+,+,+]} - V_{[-,+,+]} - V_{[+,-,+]} - V_{[+,+,-]} + V_{[+,-,-]} + V_{[-,+,-]} + V_{[-,-,+]} - V_{[-,-,-]} \tag{14}$$

Translating the centre of the sphere to the origin, and applying the same translation to each of the planes, we can see that a number of types of intersections can be found. The most trivial case occurs when no intersection occurs between the three concerned planes and the sphere, where the volume is null except if the intersection of the three planes occurs in the x_+, y_+ and z_+ octant, in which case the desired volume is that of the whole sphere. Otherwise, if an intersection between at least one of the three Cartesian planes and the sphere takes place, one of the following three cases has to be considered:

- No intersection between any of the three planes occurs inside the sphere;
- One, two or three intersections of two of the three planes is found inside the sphere, but the intersection of the three planes is located outside the sphere;
- The intersection of the three planes is found inside the sphere.

3.2.1 No intersection between the planes inside the sphere

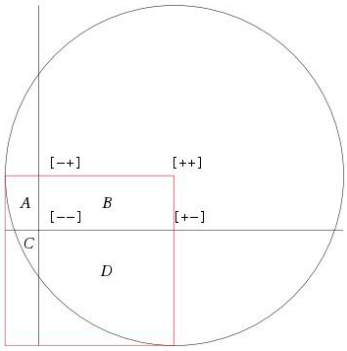

Figure 1: 2-D analogy of the formula for calculating volumes

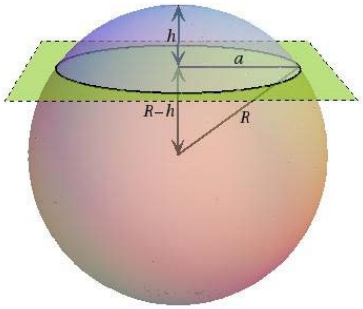

Figure 2: Intersection of a single plane with a sphere. Here, $a^2=R^2-z^2$.

When no intersection of the planes takes place inside the sphere, the only volumes that need to be considered are spherical caps (see figure 1). These volumes can be calculated using a simple integral over the axis orthogonal to the plane:

$$V_{Cap} = \int_{R-h}^{R} \pi(R^2 - z^2)dz = \frac{h^2\pi(3R-h)}{3} \tag{15}$$

Here, it is important to consider where the intersection between the three planes occur, the desired volume being null, or given by a combination of V_{Cap} calculated according to each axis.

3.2.2 Intersection of two planes inside the sphere

The intersection of two planes inside the sphere creates the region illustrated in figure 2. The volume represented by V_{Int} can be calculated again by an integral:

$$\begin{aligned} V_{Int_{(x,y)}} &= \int_0^{y_p}\int_0^{x_p} \sqrt{R^2 - x^2 - y^2}\,dxdy \\ &= \frac{1}{6}\left[\begin{array}{l} 2x_p y_p \sqrt{R^2 - x_p^2 - y_p^2} - y_p(y_p^2 - 3R^2)\arctan\left(\dfrac{x_p}{\sqrt{R^2 - x_p^2 - y_p^2}}\right) \\ - x_p(x_p^2 - 3R^2)\arctan\left(\dfrac{y_p}{\sqrt{R^2 - x_p^2 - y_p^2}}\right) - 2R^3\arctan\left(\dfrac{x_p y_p}{R\sqrt{R^2 - x_p^2 - y_p^2}}\right) \end{array}\right] \end{aligned} \tag{16}$$

The required volume will, once again, be strongly dependant on the quadrant in which the intersection of the three planes occurs, and the number of intersections occurring inside the sphere.

3.2.3 Intersection of the three planes inside the sphere

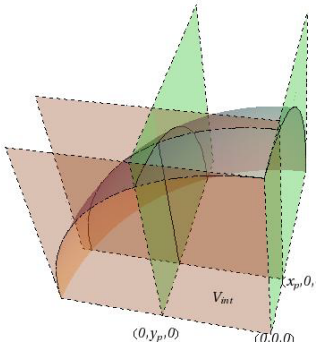

Figure 3: Intersection of two planes within a sphere

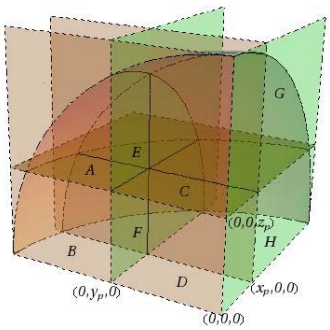

Figure 4: Intersection of three planes within a sphere

The intersection of the three planes inside the sphere creates the region illustrated in figure 3. By simple observation, we can find the desired volume using an algebraic sum of already known volumes:

$$V_{(x,y,z)} = \frac{V_{Cap_x} + V_{Cap_y} + V_{Cap_z} - V_S}{4} + V_{Int_{(x,y)}} + V_{Int_{(x,y)}} + V_{Int_{(x,y)}} - x_p y_p z_p \quad (17)$$

FORTRAN routines have been written to calculate the volumes, verified with the equivalent in Matlab [5], and then incorporated in the NXT: set of routines of Dragon.

4. Verification

4.1 Tracking

Tracking verification was performed for a unit cell containing a multi-layered sphere. Selecting a few lines (5 or 6) out of the whole set, both track lengths and intersection points with the elements of the geometry, for various positions and radii of the spherical elements, were analysed, looking for errors greater than the numerical precision.

A secondary visual verification was also performed, using module TLM:, that can be used to generate a Matlab file that can illustrate either the whole set of tracking lines (figure 4), or the intersections of the tracking lines with specified planes (figure 5), the former option producing figures that become rapidly overcrowded with an increasing number of lines being represented, but being however very useful for detecting segments out of their respective regions, or regions not crossed by any lines.

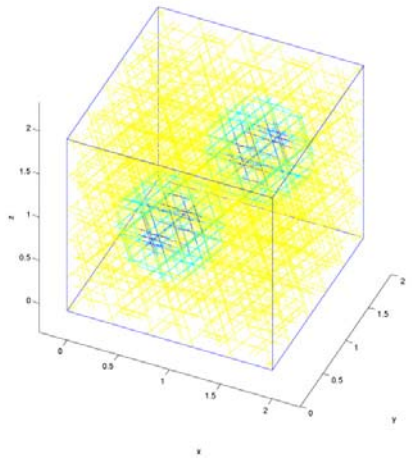

Figure 5: Tracking visualization using the TLM: module.

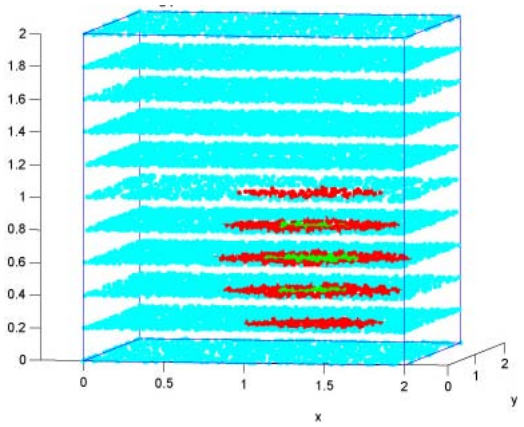

Figure 6: Intersection of tracking lines with regularly spaced planes for one region.

4.2 Volumes

The first verification was performed by observing that error induced in analytically calculated volumes by NXT: were all within the numerical precision of the machine used. Another step regarding verification has been performed, using the numerical volumes calculated with the weighted segments presented in section 2.2, to ensure both internal code consistency, and that the refinement of tracking parameters (track density and angular quadrature) leads to a better numerical volume evaluation. An example can be seen in table 1, for a simple 3 x 3 x 3 Cartesian geometry containing a three-layered sphere, and using an equal weight angular quadrature [6].

Table 1: Comparison between analytical and numerical volumes

Track density (cm^{-2})	Quadrature order	RMS error (%)	Maximum error (%)	Average error (%)
10	2	18.24550	90.06638	1.48001
	8	16.64095	90.06638	-2.55817
	16	16.85245	90.06638	-1.58190
20	2	2.48498	10.06485	0.47423
	8	3.53168	10.06484	1.35797
	16	3.72187	10.06482	1.65092
50	2	1.92751	7.78483	-0.04890
	8	1.79627	7.78483	0.52046
	16	1.78751	7.78484	0.64180
100	2	0.75645	2.73456	0.29326
	8	0.74120	2.86485	-0.00515
	16	0.53025	2.73458	0.01692

It should be observed that the track density is the main factor in determining the precision of the numerical volumes calculation, which can easily be explained by the very weak angular dependence of equation (8) for spherical geometries, which have no privileged spatial direction. However, when

CPs need to be calculated, a large enough number of angular directions is needed to represent neutrons directions.

5. Validation against a Monte-Carlo calculation

We finally present in this section a preliminary validation run that has been performed on a simple geometry consisting of either of eight spherical pins disposed so as to form a regular arrangement of pins, symmetrical in both translation and reflection (see figure 6), or randomly positioned fuel elements in the same global cell (figure 7). In both arrangements, the cell was a graphite-filled cube of 2 cm side length, while the fuel elements were 8.2% enriched, uniform uranium spheres, of radium 0.1 cm.

All calculations were performed using a JEF2.2 input-based library, in ACE format for SERPENT, and Draglib format [7], using a SHEM 361 group structure [8] for DRAGON calculations. All self-shielding calculations were performed using the subgroup projection method, described in ref. [9].

5.1 Explicit positions of the fuel elements

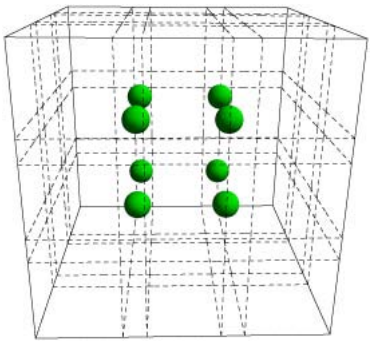

Figure 7: Test cell containing 8 regularly positioned spherical pins.

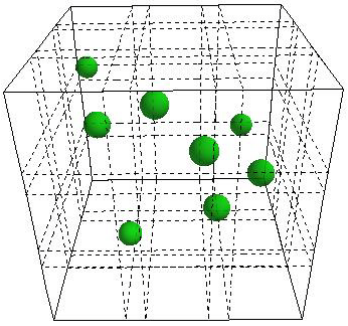

Figure 8: Test cell containing 8 randomly positioned spherical pins.

This regular arrangement has been validated against the same geometry analysed using the Monte-Carlo code SERPENT. The DRAGON calculations were performed using two resonance self-shielding methods:

- The first (C1) and more straightforward option consists in tracking the whole geometry and using this information for the resonance self-shielding procedure;
- The second (C2) faster process, where the self-shielding is performed over a single pin.

This has been done so as to determine the extra computational burden of performing the self-shielding on the whole geometry and to determine its effect on the precision. The SERPENT and DRAGON results for k_{eff}, the integrated flux and the cell averaged fission and absorption cross sections are presented in table 2.

Table 2: k_{eff} and reaction constants comparison for regularly positioned pins

Parameter	Serpent	Δ Serpent (%)	Dragon (C1)	diff. (C1) v. Serpent	Dragon (C2)	diff. (C2) v. Serpent
k_{eff}	1.72700	0.00024	1.72589	-1.1 mk	1.72465	-2.3 mk
Integrated Flux	4.77599E+02	0.00052	4.75599E+02	-0.4 %	4.75365E+02	-0.5 %
$\nu\times\Sigma_f$	3.61633E-03	0.00056	3.62886E-03	0.3 %	3.62806E-03	0.3 %
Σ_a	2.09409E-03	0.00052	2.10261E-03	0.4 %	2.10365E-03	0.5 %

Self-shielding the resonant cross sections over a single pin seems to reduce the precision of our k_{eff} evaluation as compared with the C1 option (same tracking options: 20 tracks per cm, with 44 tracking directions for both geometries). However, the computing time is significantly reduced with the alternative self-shielding treatment (C2) (from 134 to 64 minutes on our system). Increasing ten-fold the tracking densities selected for the C2 evaluation (which is nearly equivalent to the number of effective tracks in the pins for the C1 model), leads to a value of k_{eff} =1.72657 in 197 minutes, for a difference with SERPENT of -0.4 mk. Therefore, both the C1 and C2 models lead to equivalent results provided that the tracking densities are equivalent. Accordingly, a good compromise between the tracking parameters and self-shielding strategies could be reached so that precision is maximized and computational time minimized.

5.2 Random positioning of the fuel elements

A new pseudo-random positions generator using sequential addition [10] was also implemented in DRAGON. Using this technique with an imposed density of 0.4%, we were able to randomly generate eight pins positions, obtaining the distribution shown in figure 7 (the maximum density we can obtain with this generator is of about 21%). This distribution has been fed to both SERPENT and the two self-shielding methods described in section 5.1. The results are given in table 3.

Table 3: k_{eff} and reaction constants comparison for randomly distributed pins

Parameter	Serpent	Δ Serpent (%)	Dragon (C1)	diff. C1 v. Serpent	Dragon (C2)	diff. C2 v. Serpent
k_{eff}	1.72719	0.00023	1.727823	0.63 mk	1.726323	0.87 mk
Integrated Flux	4.77141E+02	0.00050	4.77157E+02	0.003 %	4.76870E+02	-0.06 %
$\nu\times\Sigma_f$	3.62017E-03	0.00054	3.62108E-03	0.03 %	3.62011E-03	-0.002 %
Σ_a	2.09608E-03	0.00050	2.09575E-03	-0.01 %	2.09701E-03	0.04 %

The results seen here follow the same trend as those obtained with the uniform positioning of the pins, being however even more accurate. The main reason for this increased accuracy can be explained by observing that the error in the numerical volumes calculated based on the tracking lines is reduced when randomly distributed pins are considered. As a result, the very small pin volumes are globally tracked better and the precision of the CP calculated is improved. This

computing anomaly could however be removed using finer tracking parameters, as discussed in the previous section.

It can finally be noted that the random positioning of the particles increased the k_{eff}, (by about 2 mk for DRAGON calculations) which is to be expected due to the greater chance of seeing some of the fuel elements packed closer together in some portion of the cell, effectively increasing the local packing density.

6. Conclusions

The goal of this project, which was to apply full 3D collision probabilities to spherical geometries, has been successfully achieved. A thorough verification has been made, and preliminary validation has been performed, leading to results consistent with those obtained through Monte-Carlo simulations, both for regularly positioned fuel elements and for randomly generated positions of these fuel elements.

The 3D CP method with explicit spherical representation does not eliminate the need for innovative double-heterogeneity formalism, such as the method proposed by Hébert [11], and can possibly be viewed as an additional validation tool for developers. Further validation still needs to be performed, mainly focusing on results obtained with random generation of spherical particles in a uniform matrix using a specified density, that has already been implemented, or with burnup calculations. Validation against double-heterogeneity treatment has already been proven conclusive, with further validation against TRISO in VHTR fuel to be expected. The next development objectives should eventually focus on implementation of capacities for the treatment of full 3D hexagonal cells, and integration of combined spherical-cylindrical geometries, permitting full treatment of prismatic VHTR geometries.

7. Acknowledgements

The authors would like to thank the National Science and Engineering Research Council of Canada for providing the financial support for this research. The authors would also like to especially thank Nicolas Martin of École Polytechnique de Montréal for many helpful discussions.

8. References

[1] G. Marleau, A. Hébert and R. Roy, "New computational methods used in the lattice code Dragon", Proc. Intl. Top. Mtg. on Advances in Reactor Physics, Charleston, USA, March 8-11, 1992.

[2] J. Leppänen, *PSG2 / SERPENT: A continuous-energy Monte-Carlo reactor physics burnup calculation code*, Methodology, User's Manual, Validation report, 2009.

[3] A. Hébert, *Applied reactor physics*, Presses Internationales Polytechnique, Montréal, 2009.

[4] A. Zkiek and G. Marleau, "Verification of DRAGON : The NXT: tracking module", 28th annual conference of the Canadian nuclear society, St. John, N-B, Canada, 2007.

[5] Matlab, Version 2007a, The MathWorks, Natick, Massachussetts.

[6] B. G. Carlson, *Tables of equal weight quadrature EQ_n over the unit sphere*, Technical Report LA-4734, Los Alamos Scientific Laboratory, 1971.

[7] A. Hébert and Saygin, H., "Development of DRAGR for the Formatting of DRAGON Cross-Section Libraries," Seminar on NJOY-91 and THEMIS for the Processing of Evaluated Nuclear Data Files, Saclay, France, April 7–8, 1992.

[8] A. Hébert and Santamarina, A., "Refinement of the Santamarina-Hfaiedh energy mesh between 22.5 eV and 11.4 KeV," Proc. Int. Conf. on the physics of reactors. Nuclear Power: A Sustainable Resource, PHYSOR 2008, Interlaken, Switzerland, 2008.

[9] A. Hébert, *Development of the subgroup projection method for resonance self-shielding calculations*, Nuclear Science and Engineering, 162, 56-75, 2009

[10] Widom, B., "Random Sequential Addition of Hard Spheres to a Volume,"The Journal of Chemical Physics, 44, 3888–3894, 1965.

[11] A. Hébert, *Scattering Reduction of the Double-Heterogeneity Treatment in Dragon*, Nuclear Science and Engineering, 160, 1-6, 2008.