Criticality calculations on pebble-bed HTR-PROTEUS configuration as a validation for the pseudo-scattering tracking method implemented in the MORET 5 Monte Carlo code

Benoit Forestier\textsuperscript{a,}\textsuperscript{*}, Joachim Miss\textsuperscript{a}, Franck Bernard\textsuperscript{a}, Aurélien Dorval\textsuperscript{a}, Olivier Jacquet\textsuperscript{b}, Bernard Verboomen\textsuperscript{c}

\textsuperscript{a} Institut de Radioprotection et Sureté Nucléaire, Fontenay aux Roses, France
\textsuperscript{b} Independent consultant, France
\textsuperscript{c} Belgian Nuclear Research Center (SCK-CEN), Belgium

Abstract

The MORET code is a three dimensional Monte Carlo criticality code. It is designed to calculate the effective multiplication factor (\(k_{\text{eff}}\)) of any geometrical configuration as well as the reaction rates in the various volumes and the neutron leakage out of the system. A recent development for the MORET code consists of the implementation of an alternate neutron tracking method, known as the pseudo-scattering tracking method. This method has been successfully implemented in the MORET code and its performances have been tested by means of an extensive parametric study on very simple geometrical configurations.

In this context, the goal of the present work is to validate the pseudo-scattering method against realistic configurations. In this perspective, pebble-bed cores are particularly well-adapted cases to model, as they exhibit large amount of volumes stochastically arranged on two different levels (the pebbles in the core and the TRISO particles inside each pebble).

This paper will introduce the techniques and methods used to model pebble-bed cores in a realistic way. The results of the criticality calculations, as well as the pseudo-scattering tracking method performance in terms of computation time, will also be presented.

1. Introduction

The MORET code [Miss et al, 2007] is a three dimensional criticality Monte Carlo code developed by IRSN, the French institute for radiological protection and nuclear safety. It is designed to calculate the effective multiplication factor (\(k_{\text{eff}}\)) of any 3D geometrical configuration as well as the reaction rates in the different volumes and the neutron leakage out of the system. As part of the French criticality package CRISTAL [Gomit et al, 2003], the multigroup version of MORET is, in conjunction with the APOLLO2 assembly level deterministic code, the most commonly used Monte Carlo code for French criticality calculations.

A recent development for the MORET code consists of the implementation of a new simulation option designed to perform neutron tracking using an alternate method known as the pseudo-scattering tracking method, or the Woodcock tracking method.
[Woodcock et al., 1965; Forestier et al., 2007]. The main advantage of this method is to accelerate the simulation (in particular for geometrical systems with lots of volumes) without introducing any bias in the simulation. As a counterpart, this tracking algorithm shows bad performances when large differences are observed on the cross-sections in the system (of a few orders of magnitude). This option has been successfully implemented in the multigroup version of the MORET 5 code, so that the user is now able to specify which part of the geometry is to be treated using the Woodcock tracking method. The MORET 5 code has thus the particularity to allow the use of different tracking algorithm for different parts of the geometry.

Extensive performance studies were led for very simple geometrical configurations, in order to survey the efficiency of the pseudo-scattering method, in terms of computation time, against various parameters such as the number and type of volumes or their chemical content. The results of these studies show a reduction of the computation time up to a factor 5, even for configurations with only a few hundred volumes.

In this context, the aim of the present work is double:

- firstly, to validate the use of the pseudo-scattering tracking method implemented in the MORET 5 code for realistic configurations; in this perspective, pebble-bed reactors (PBR) are well-adapted and challenging cases to study, with a huge amount of volumes irregularly arranged on two levels of heterogeneity;
- secondly, to perform calculations on pebble-bed core configurations, involving new methods to treat stochastic fuel pebble distribution in pebble-bed reactors.

To fulfil the modelling task, we have developed some dedicated tools, also introduced in this paper. These will be the basis for the development of a full toolkit (available for users as a pre-treatment module).

2. The pseudo-scattering method

The Woodcock tracking method, also known as the pseudo-scattering or fictitious-scattering tracking method, is an alternate to the “classical” way to perform particle transport in Monte Carlo codes. It has been developed during the sixties, for the GEM Monte Carlo code [Woodcock et al., 1965], in order to speed up the calculations for geometrical configuration with a large amount of volumes.

The method applied in most Monte Carlo transport calculations is to simulate the neutron histories by tracking each neutron through homogeneous material. The distance to the next collision is randomly sampled according to an exponential law which depends on the total cross-section for the material considered. Since the geometry usually consists of various material regions, the tracking has to be stopped and the exponential law adjusted each time the neutron enters a new material region. In practice, in order to perform these actions, the code needs to know the distance between the neutron and the various region boundaries. This is a rather time consuming procedure, especially when lots of volumes are to be considered, which has to be performed at each collision site and material boundary crossing.

The idea behind the Woodcock method [ANSWER, 2003; Woodcock et al., 1965] is to virtually homogenize the whole system by giving it a unique total cross-section ($\Sigma^M$), that is equal to the maximum value of the total cross-sections for the various material present in the system ($\Sigma^i$, $\Sigma^i_T$, $\Sigma^i_\delta$, $\Sigma^n_\delta$). This is performed by introducing the concepts of virtual collisions that do absolutely nothing) and virtual cross-sections. The virtual cross-sections ($\Sigma^i_\delta$, $\Sigma^i_\delta$, $\Sigma^n_\delta$) are evaluated, for each material, so that:

$$\Sigma^M = \Sigma^i + \Sigma^i_\delta = \Sigma^i_T + \Sigma^i_\delta = \ldots = \Sigma^n + \Sigma^n_\delta$$

(1)

with

$$\Sigma^M = \text{Max}(\Sigma^i_T, \Sigma^i_\delta, \ldots, \Sigma^n_\delta)$$

(2)

Once this homogenisation was done, the neutron free path to the next collision can be sampled using a single value of cross-section ($\Sigma^M$). There is no need to stop the tracking at each material boundary, or even calculate the distances to the boundaries surfaces. Instead, it is sufficient to determine the material lying at the neutron position – which is quicker than calculating the distances to boundaries – in order to decide whether a real or virtual collision is taking place (the real collision probability is given by $P = \Sigma^R / \Sigma^M$). For this reason, the main advantage of the method is to speed up calculations for systems with a large number of
boundaries whereas its main drawbacks are its inability to treat efficiently systems exhibiting large variations on the cross-sections (lots of useless virtual collisions occurring), and the unavailability of the track-length estimator. Additionally, it is worth to mention that this tracking method introduce no bias at all. Everything is as if a virtual material (that does not affect the neutrons) would have been introduced in each region, with a density freely adjusted in order to fulfil the homogenisation of the whole geometry.

3. Methods used to model the double heterogeneity

The interest of the pseudo-scattering tracking method arises when the system exhibits a large amount of volumes. Moreover, these volumes should not be regularly arranged otherwise it would be more efficient to use the array capabilities of the MORET code instead of pseudo-scattering. For this reasons, the pebble-bed core configurations are well-adapted cases to study for our validation purposes.

3.1. In bed with pebbles

Analysis of pebble-bed reactors are usually performed using triangular pitch meshes. Nevertheless, the above considerations led us to consider a more realistic approach, where the pebbles are arranged in the core as if they were dropped by a pneumatic fuelling machine. Obviously, it is inconceivable to create this kind of input files for the MORET code “by hand”. As a result, we needed to develop an automated method to process the pebbles’ positions in the core. Various methods, consisting of stochastically distributing the pebbles, were investigated, but remained unsatisfying.

The biggest hitch for this task comes from the difficulty to stack the pebbles as if they were packed by gravity, with each pebble in contact with the ones just above and below. The chosen solution has been to use the physics engine ODE (Open Dynamics Engine) [Smith, 2006] which is a free C/C++ library, developed by R. Smith, for simulating articulated rigid bodies dynamics. It is able to solve the Newtonian equation for motion, can handle rigid bodies attached by mean of joints (hinge, slider...) and has built-in collision detection functionalities. It is also flexible, easy to use and can be widely parameterised. Thus, by modelling the pebble-bed reactor core in ODE, it is possible to simulate the falling and heaping of pebbles in a core configuration and then to get back the pebbles position in order to create the corresponding MORET 5 input file.

![Fig. 1, arrangement of 9,800 pebbles in the fully modelled HTR-PROTEUS cavity used for random core configurations and pebble cross-section as they will be modelled in the MORET 5 code.](image)
An ODE-based program (see Forestier, (2008) for description and performance review), designed to automatically generate clean MORET 5 input for pebble-bed configurations, is now operational. It is planned to be delivered to the user in a pre-treatment module, along with other scripts that give access to higher level functionalities (cf. 3.3).

Note also that, for now, the pebble-bed generator is only fitted to small scale systems.

3.2. Thousands of TRISO embedded in each pebble

In a similar point of view, we also want to be able to model each pebble as a set of really randomly distributed TRISO particles within a carbon matrix. This is a rather simple task and we are already able to automatically create that kind of realistic MORET modelling for pebbles. A pre-treatment script has been developed to give access to this functionality by the mean of a single MORET keyword and will be part of the MORET pre-treatment toolkit.

As an example, figure 1 shows a pebble-bed configuration as it can now be modelled in the MORET 5 code. In the lower part, we can see a three dimensional view of a pebble-bed reactor core filled, using ODE, with 9800 pebbles, dropped randomly from the top of the core. The upper part of the picture represents a cross-section of a 3-cm-radius pebble, showing the stochastic distribution of 9394 TRISO particles within a 2.35-cm-radius carbon matrix that is surrounded by a 0.65-cm-thick graphite layer.

3.3. The pre-treatment module

The so-called pre-treatment module for the MORET code is a set of scripts/tools that are meant to be run automatically, before any calculation, without any intervention of the user. Each script gives access to a new keyword (a specific syntax has been built up) and offers a higher level functionality such as a complex shape or an arrangement of bodies that would involve a tedious work to create « by hand » the proper input file for the MORET 5 code. For now, a few functionalities have been developed, including the creation of a pebble filled with TRISO particles, the creation of a pebble bed, the creation of a hexagonal close packed lattice or the creation of n-sided polygonal cavity like the PROTEUS core used for deterministic pebble arrangement.

These scripts/programs are designed to be run before the MORET code. They take MORET 5 input files including some pre-treatment keywords, edit them in the way specified by the given parameters and output the modified MORET 5 input file. In the user’s point of view, this module brings some simple, compact and totally transparent way to write complex input files or to follow some modelling recommendations.

In the developers’ point of view, this approach is also beneficial because these functionalities are independent from the MORET code itself. In this perspective, each new functionality can be developed using any programming language or technology and the module can be promptly updated without needing any re-validation process of the MORET code. This allows a high reactivity to the users’ needs.

4. Modelling the pebble bed HTR-PROTEUS

4.1. General considerations

The PROTEUS apparatus [Mathews and Williams, 1996; Williams, 1996] (see Fig. 2) can be described as a cylinder of graphite, 3.30 m in height and 3.26 cm in diameter. A central cavity (the core), having a 22-sided irregular polygonal horizontal cross-section, with a flat-to-flat separation of 1.25 m, is situated 78 cm above the bottom of the lower graphite reflector. An upper axial graphite reflector is inserted as a stopper in the upper part of the cavity and is maintained by an aluminium structure. The core is partially filled with a mixture of pure graphite pebbles and fuel pebbles, loaded either in deterministic or in random arrangement. The number of pebbles loaded is of the magnitude of a few thousand depending on the core configuration.

Depending on the core configuration some modifications are brought to the cavity by mean of graphite filler pieces:

- for deterministic arrangement of pebbles, 12 vertical filler pieces are added in the cavity, so that the pebbles can be manually meshed within a ~1.9 m high and 1.20 m wide cylinder with a 12-sided polygonal cross-section;
for random arrangement of pebbles, the 22-sided cavity floor is carpeted with 21 filler pieces, giving it the shape of a 10-degrees-slope funnel.

Calculations will be performed for core configurations involving a random distribution of pebbles (core 4). In this context, the pebble-bed will be modelled in a stochastic way, as if the pebbles had been dropped from the top of the cavity, with the aid of a pneumatic fuelling machine. Note that because of the filler pieces modifying the cavity floor, we will not be able to model properly the loaded core with meshed pebbles. Besides, we do not have any information concerning the shape of the pebble-bed top surface, which could form a bump, due to the loading of the core that has been performed by dropping pebbles from one single point. The shape of the modelled bump is highly correlated to the pebble-bed generator settings (friction coefficient, jitter magnitude on pebble launching...), so that, without information about this bump or the fuelling machine, it is difficult to properly parameterise the pebble-bed generator. Furthermore, fully realistic modelling of pebble-bed reactor (with 9400 TRISO particles in each fuel pebbles) implies the use of a huge total number of volumes (~3.10^8 in our case). Performing such calculations is thus very memory demanding and will not be possible with our computation means.

As the Woodcock tracking method is currently only implemented in the multigroup route of the MORET code, we need to compute, with an external assembly code, equivalent cross-sections for each of the media considered in the system (a medium can be a pebble, an assembly, a mix of various chemical media...). These cross-sections are obtained by mean of the APOLLO2 assembly level deterministic code, which is also part of the French criticality package CRISTAL [Gomit, 2003].

4.2. About the efficiency of the Woodcock method, a priori balance of relative interest for each tracking option

We have learned from the extensive survey of the Woodcock method performances that it is really difficult to guess, whether or not its use will be beneficial, for a given configuration; and that it is even more difficult to guess which part of the system should be treated using this pseudo-scattering tracking method, in order to optimise the calculation time. Indeed, there is a small number of parameters that affect the performances of the Woodcock tracking method: the number of bodies (the more bodies we have, the more CPU-time will be gain), the shapes of the bodies (the Woodcock method is more efficient with boxes than with cylinders or spheres), the size of each fissile volume (the Woodcock tracking will have a slower convergence with small fissile volumes), the media considered and the relative volumes they occupy (a critical loss of efficiency can be observed when there is a large discrepancy between the cross-sections of the various media). Nevertheless, the relative importance of each parameter is deeply impacted by the time spent by each neutron in the various volumes and is therefore difficult to evaluate a priori.

Concerning the modelling of Pebble-bed HTR-PROTEUS, we can easily see that the huge number of bodies will be very advantageous to the Woodcock tracking, whereas the spherical shapes of the volumes will tend to limit this advantage. We can also note that, for realistic modelling of TRISO particles, the smallness of the UO2 core will probably tend to lengthen the convergence. Actually, the big uncertainty concerning the benefits of the Woodcock method for such configurations comes from the huge differences in the cross-sections of the media in presence. As pebble-bed reactors are gas-cooled, the slowing down of the
Woodcock tracking tends to become dramatic, especially when the core is not entirely filled and thus exhibits a large void in the geometry. To limit these effects, the proportion of air in the « Woodcock part of the geometry » has been limited by dividing the core into two parts: the active part, where the pebbles lie and that is likely to undergo the Woodcock treatment; and the passive part, consisting of a single cavity filled with air and that will be treated with the «conventional» tracking method. Even when splitting the core, this aspect of the modelling can highly handicap the Woodcock tracking method. In particular, it can become crucial for configurations where the upper part of the pebble-bed forms a bump, resulting in a larger volume of air in the active part of the core.

In conclusion, we cannot say, without running the code, whether the number of volumes will be sufficient to balance the loss of efficiency brought by the presence of air or not. Neither can we say in what proportion the efficiency will be affected by the presence of a bump on the top of the pebble-bed or which part of the geometry should be treated using the Woodcock tracking (the whole core or only the pebbles).

5. Results

Various calculations have been performed with the APOLLO2 - MORET 5 codes. The results for these calculations will be given, compared and commented in terms of computation time and effective multiplication factor (k<sub>eff</sub>). In order to do CPU-time comparisons, the calculations have been performed twice for a single configuration: once with the «conventional» neutron tracking algorithm and a second time using the Woodcock tracking method.

5.1. Description of the treated configurations

Only one full scale model for the HTR-PROTEUS can be used either with or without the Woodcock method: a set of homogeneous pebbles arranged in the core as if it had been filled with the aid of the fuelling machine. Every configuration involving a realistic pebble modelling cannot be used with our present computation means; and, because our Woodcock tracking method implementation is not compatible with the MORET lattice capabilities, it is not possible to use a meshed-type model for pebbles. The chosen configuration is core 4.3 [Williams, 1996]. It will be referred as PROTEUS4 in the following. For this single full scale configuration, the 22-sided cylindrical cavity has been modelled in detail along with the 21 filler pieces. A 9,800 pebble-bed (4,900 fuel pebbles and 4,900 moderator pebbles), with a flat top surface, has been created using the pebble-bed generator discussed in section 3 (cf. fig. 1 for a 3D view of the created pebble-bed). The packing factor obtained is real close to the experimental one: 9,800 pebbles stacked with the pebble-bed generator in the fully modelled PROTEUS cavity result in a 149.9 cm high bed whereas the experimental value obtained at PROTEUS, for the equivalent core configuration, is 150 cm [Williams, 1996]. The aluminium structures holding the upper axial reflector have been roughly modelled (with all aluminium present below the bottom surface of the upper axial reflector gathered in a single cylinder). Eventually, as the reactivity correction for the critical loading is given with much detail in the pebble-bed HTR-PROTEUS technical documentation [Williams, 1996], we chose to have a simple model for the carbon reflectors, for which no control/shutdown/safety… rod is modelled. Fig. 3 shows a vertical cross-section of HTR-PROTEUS as it has been modelled in the MORET 5 code. The fuel pebbles are represented in red with a grey outer shell, the moderator pebbles in black and the aluminium structures in grey-green. Note also the 10-degree-slope funnel at the bottom of the core.

As it will be difficult to draw clear conclusions for simulations involving random arrangement of pebbles, an additional calculation point (referred in the following as PROTEUS9) will be given, for a deterministic core loading (core 9.2), in order to allow a straightforward validation of the model.

Additionally, the results for a simple configuration consisting of a single pebble in a 8*8*8-cm<sup>3</sup>-box carbon reflector will be given for different type of pebble models:

- homogeneous (referred as 1-H),
- realistic with homogeneous TRISO particles (referred as 1-TH),
- realistic with true TRISO particle (referred as 1-TT).

Eventually, the same calculations will also be performed for one pebble on an infinite lattice (referred as M-H, M-TH and M-TT respectively). These last ones are particularly fitted for APOLLO2 modelling comparisons.
5.2. Results

Results for the various configurations, in terms of computation time and k\textsubscript{eff}, can be consulted in tables 1 and 2.

First of all, we would like to notice the nice agreement between the k\textsubscript{eff} evaluated using the Woodcock and the “conventional” tracking methods. In addition, the k\textsubscript{eff} results for the three meshed configurations (M-TT, M-TH and M-H) show that it is possible to perform homogenisation over a whole pebble without deeply impacting the final k\textsubscript{eff}.

The k\textsubscript{eff} obtained with the PROTEUS configurations have to be compared with the experimental results given by Williams (1996) (1.01320 for PROTEUS4 and 1.01140 for PROTEUS9). The PROTEUS9 configuration show a nice agreement with the experimental value within a few hundreds pcm. For comparison, the TRIPOLI4 calculation for the same core loading gives a k\textsubscript{eff} of 1.0089(5) [Köberl and Seiler (2004)]. The PROTEUS4 configuration exhibits only 1500 pcm difference with the measured k\textsubscript{eff}. This can be considered satisfactory, taking into account the random arrangement of pebbles in such configuration and the shape of the pebble-bed top-surface. Note also that the discrepancy between these results and the previous ones [Forestier, 2008], is due to the taking into account of \( S(\alpha,\beta) \) and better anisotropy model for carbon.

### TABLE 1
Comparison of the Woodcock method (w) with the “conventional” (c) tracking algorithm: computation time (in s) and figure of merit (\( \varepsilon = 1/\sigma_T \)) for the various configurations.

<table>
<thead>
<tr>
<th>Config.</th>
<th>( T_c ) (s)</th>
<th>( T_w ) (s)</th>
<th>( T_c / T_w )</th>
<th>( \varepsilon_w / \varepsilon_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-TT</td>
<td>8467</td>
<td>2473</td>
<td>3.42</td>
<td></td>
</tr>
<tr>
<td>1-TH</td>
<td>3837</td>
<td>652</td>
<td>5.88</td>
<td></td>
</tr>
<tr>
<td>M-TT</td>
<td>195066</td>
<td>280033</td>
<td>0.697</td>
<td>0.27</td>
</tr>
<tr>
<td>M-TH</td>
<td>167960</td>
<td>48526</td>
<td>3.46</td>
<td>0.75</td>
</tr>
<tr>
<td>PROTEUS4</td>
<td>72568</td>
<td>85118</td>
<td>0.853</td>
<td>0.85</td>
</tr>
</tbody>
</table>

### TABLE 2
Comparison of the Woodcock method (w) with the “conventional” (c) tracking algorithm: k\textsubscript{eff} and mean deviation for the various configurations.

<table>
<thead>
<tr>
<th>Config.</th>
<th>k\textsubscript{eff}\textsubscript{c} ± ( \sigma ) (pcm)</th>
<th>k\textsubscript{eff}\textsubscript{w} ± ( \sigma ) (pcm)</th>
<th>Δk\textsubscript{eff} (pcm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-TT</td>
<td>1.76864 ± 62</td>
<td>1.76870 ± 99</td>
<td>6</td>
</tr>
<tr>
<td>M-TH</td>
<td>1.77490 ± 21</td>
<td>1.77416 ± 45</td>
<td>-74</td>
</tr>
<tr>
<td>M-H</td>
<td>1.76821 ± 54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROTEUS4</td>
<td>1.02814 ± 98</td>
<td>1.03051 ± 98</td>
<td>237</td>
</tr>
<tr>
<td>PROTEUS9</td>
<td>1.01473 ± 99</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Concerning the CPU-time spent for both type of tracking methods, we first notice that, as expected, the simulation is greatly speeded up when using the Woodcock method on a single truly modelled pebble. More surprising are the results for the meshed configurations. A closer look at the output for these calculations shows that, for the M-TT configuration, the MORET 5 code had a slower convergence when using the Woodcock tracking method, resulting in 50% more cycles. Besides, for computation time reasons, only 500 neutrons were used for these simulations. One explanation could thus be that, due to the absence of track-length estimator in the Woodcock method, to the small size of the fissile volumes and to the small number of neutrons involved, the advantage brought by the
high number of volumes vanishes because of this statistics handicap. This explanation tends to be strengthened by the better performances obtained when the TRISO are homogenised instead of being truly modelled (see configurations M-TH and 1-TH as compared to configurations M-TT and 1-TT).

Eventually, concerning the PROTEUS4 configuration, the results in terms of CPU-time seem to demonstrate that treating the whole PROTEUS pebble-bed using the Woodcock method isn’t worth, due to the presence of air in the core (note that this conclusion may become even more obvious when considering a pebble-bed with a bump on the top surface). Besides, PROTEUS is a really small size system and the Woodcock tracking method may become advantageous for larger scale reactors.

6. Conclusions and perspectives

The techniques used for realistic modelling of pebble-bed reactors have been introduced. In particular, a new pre-treatment tool designed to generate pebble-bed configuration for the MORET code has been described.

Furthermore, various calculations have been performed with the two neutrons tracking methods implemented in the MORET 5 code. One of these configurations was the full-scale HTR-PROTEUS with random pebble arrangement (core 4.3). Results, in terms of computation time and effective multiplication factor, have then been presented and conclusions have been drawn:

- a great agreement has been observed between the “Woodcock” and “conventional” calculations;
- the Woodcock tracking seems not to be well adapted for full pebble-bed calculations.

At this point, we have learned few things about the Woodcock tracking method performances. Nevertheless, there are still lots of questioning and number of configurations/models to investigate as, for example, using the Woodcock treatment inside each fuel pebble instead of using it for the whole pebble-bed. Note also that calculations on realistic configurations encountered in criticality safety studies, for which the Woodcock tracking treatment could be beneficial, are also planned to be performed.

References


E.R. Woodcock, T. Murphy, P.J. Hemming, T.C. Longworth., 1965, “Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry”, ANL7050.

ANSWER software service, 2003, “Hole Geometry”, ANSWERS/MONK/REPORT/003


B. Forestier et al, 2008, “Criticality calculations on realistic modelling of pebble-bed HTR-PROTEUS as a validation for the Woodcock tracking method implemented in the MORET 5 Monte Carlo code”, to be published in proc for the 2008 International Congress on Advances in Nuclear Power Plants, Anaheim, USA

