M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

Three-Dimensional Neutron Streaming Calculations Using Adaptive Multilevel Splitting

Henri Louvin,[a] Eric Dumonteil,[b] Tony Lelièvre[c]

[a]*CEA Saclay, DEN/DM2S/SERMA/LTSD, F-91191 Gif-sur-Yvette* [b]*IRSN, PSN-EXP/SNC, F-92262 Fontenay-aux-Roses*
[c]*Université Paris-Est, CERMICS (ENPC), INRIA, F-77455 Marne-la-Vallée*
henri.louvin@cea.fr

Abstract - *TRIPOLI-4® is the first Monte Carlo particle transport code to include the variance reduction algorithm known as Adaptive Multilevel Splitting (AMS), which aims to help simulate rare events with Monte Carlo. This work presents a description of the AMS algorithm adapted to the field Monte Carlo particle transport, some insight on its implementation within TRIPOLI-4®, and the results obtained on a three-dimensional streaming problem.*

I. INTRODUCTION

Adaptive Multilevel Splitting (AMS) is a variance reduction algorithm introduced to the field of applied mathematics in 2007 by Cérou and Guyader [1]. Originally designed to help estimate rare events occurrence probabilities in Monte Carlo simulations of continuous Markov chains, it has been theoretically extended to the simulation of discrete Markov chains by Bréhier et al. [2].

Based on those theoretical results, the AMS algorithm has been adapted to the specificities of Monte Carlo particle transport [3], and implemented in the CEA transport code TRIPOLI-4®. In this context, it is used to increase the number of realisations of rare events, thus reducing the variance on estimated quantities related to these events.

The algorithm mechanism is described in Section II. We present in Section III. some details concerning the AMS implementation in TRIPOLI-4®, and its interactions with the module generating importance maps. The central part of this work is the study of a three-dimensional streaming problem, consisting in the transport of neutrons in a seven-legged air labyrinth in concrete. The results obtained both with and without AMS use are presented and discussed in Section IV.

II. THE AMS ALGORITHM

The AMS algorithm is designed to help simulate rare events in Monte Carlo simulations. In the context of shielding simulations, it is used to help the simulated particles to reach an area of interest of the geometry, in order to reduce the variance on a score (or tally) estimated in that area.

In TRIPOLI-4® , the simulated particles are divided into groups, called *batches*, which are simulated one after the other. The AMS algorithm is an iterative method that is applied at the end of each batch. Each AMS iteration consists in two steps: First, the simulated particles of the batch are sorted with regard to their importance (See Section 1.). Then, the lowest-rated particles are resampled by splitting the other tracks as described in Section 2. The number of particle that are resampled at each iteration of the algorithm is defined before the first iteration and remains constant throughout the simulation. It has already been proved to have a limited impact on the variance reduction efficiency, and is usually set to 1% of the simulated particles [3].

1. The Sorting Step

A. Importance Functions

In order to use the AMS algorithm, one has to be able to determine which regions of the geometry are of interest to the simulation. Therefore, the geometry has to be associated to an importance function, which maps any point of the phase space to an importance value, related to the probability for a particle located at a given point P to contribute to the final score. We denote it by

$$I(P) = I(\mathbf{X}, \mathbf{\Omega}, E), \tag{1}$$

where $\mathbf{X}$, $\mathbf{\Omega}$ and E are the position, direction and energy of the point P, respectively.

It has to be noted that within AMS, this function is only used to rank the particles *with respect to one another*, so that the value of the importance at a given point does not need to have a signification on its own. This property is one of the strengths of the AMS algorithm, as it permits the use of trivial importance functions for any problem, such as the invert of the distance to the area of interest.

It is believed (though not theoretically proved), that the *adjoint score* of the problem is the more efficient importance function for AMS. Indeed, the adjoint score at a given point of the phase space is the average score generated in the area of interest by a particle emitted from this point, thus giving the most precise estimation of the interest of this point to the score.

However, the determination of the adjoint flux is a problem as complex as the original one. Furthermore, if one has access to the adjoint score, the solution of the direct problem is directly available as the value of the adjoint score at the source point, obviating the need to perform the simulation in the first place.

Consequently, the best candidate as importance function for the AMS algorithm is an approximation of the adjoint score. We present and compare in Section IV. the results obtained with AMS using two importance functions, one purely spatial, and the other automatically generated by TRIPOLI-4®.

B. Particle Tracks Importance

In order to sort the particles, a value of importance has to be attached to each of them.

When a new particle is simulated, a particle *track* is created. We denote by $P_0 = (\mathbf{X}_0, \mathbf{\Omega}_0, E_0)$ the emission point of a particle and the first point of its track. This particle travels along straight lines between collisions with the medium, each collision resulting either in the absorption of the particle or in a random change of its direction and energy. If the particle undergoes N collisions before being absorbed or leaking out of the geometry, we define its *track* T as follows:

$$T = (P_0, \ldots, P_N), \tag{2}$$

where $P_i = (\mathbf{X}_i, \mathbf{\Omega}_i, E_i)$ represents the properties of the particle outgoing its i-th collision with the medium. Using these notations, we introduce the importance of a particle track as:

$$I(T) = \max_{i \in [0,N]} I(P_i). \tag{3}$$

C. Definition Of The Splitting Level

Let n be the number of simulated particles, k the number of tracks to be resampled per iteration and q the number of the current iteration.

After the n particles have been absorbed or leaked out of the geometry, the AMS algorithm computes the importance $I(T_j),\ j \in [1, n]$ of each particle track. The splitting level is then defined as the k-th smallest value in the sample $(I(T_1), \ldots, I(t_n))$. Let us denote it by Z_q.

Each particle track having an importance less or equal to Z_q is deleted, and the number K_q of suppressed particles at this iteration is kept in memory. It has to be noted that K_q may very well be greater than k if multiple tracks have the same importance. However, K_q can never be strictly less than k.

2. The Splitting Step

A. Resampling Process

Once the sorting step of iteration q is over, the AMS proceeds to resample the K_q particle tracks that have been deleted from the simulation, so that n distinct tracks are available for the next iteration of the algorithm.

For each of the particles to be resampled, one of the $n - K_q$ remaining tracks is randomly selected. Let us denote it by $T_j = (P_i),\ i \in [0, N_j]$. The emission point P_{split} of the new particle is then defined as:

$$P_{split} = \inf \left\{ i \in [0, N_j] : I(P_i) > Z_q \right\}, \tag{4}$$

which is the first point of the track T_j having an importance greater than Z_q.

B. AMS Global Weight

The resampling process splits K_q particle tracks amongst a set of $n - K_q$ tracks, therefore all particles weights have to be weighted at the end of iteration q by a factor

$$\begin{aligned} W_q &= \frac{n - K_q}{n} \\ &= 1 - \frac{K_q}{n}, \end{aligned}$$

to ensure unbiasedness [1].

In practice, the cumulated weight factor due to the AMS process from iterations 1 to q is the same for every particle, and is stored in a global weight

$$\begin{aligned} w_q &= \prod_{i=1}^{q} W_i \\ &= \prod_{i=1}^{q} (1 - \frac{K_i}{n}). \end{aligned}$$

3. Scoring Step

The AMS algorithm stops at the end of iteration q if $n - K_q + 1$ particles have reach the area of interest. An estimator can be constructed for any score ϕ in the area of interest. If we denote by $\hat{\phi}_{MC}$ the value estimated using a standard Monte Carlo estimator, then

$$\hat{\phi}_{AMS} = \hat{\phi}_{MC} \times w_q \tag{5}$$

is an unbiased estimator of the quantity ϕ [2].

III. IMPLEMENTATION IN TRIPOLI

This section presents some details of the AMS implementation within TRIPOLI-4®.

1. Importance Functions

A. Spatial Importance Functions

The AMS implementation in TRIPOLI-4® allows the use of purely spatial importance functions. Given $\mathcal{D}$ a point, a line or a simple 3D-surface (plane, cylinder or sphere) within the geometry, two separate importance functions are available in the code: Either the importance increases with the distance to $\mathcal{D}$, or it decreases. Formally, the spatial importance of a point $(\mathbf{X}, \mathbf{\Omega}, E)$ is given by either one of the two functions I_{from} and $I_{towards}$:

$$I_{from}(\mathbf{X}, \mathbf{\Omega}, E) = dist(\mathbf{X}, \mathcal{D}), \tag{6}$$

$$I_{towards}(\mathbf{X}, \mathbf{\Omega}, E) = \frac{1}{dist(\mathbf{X}, \mathcal{D})}. \tag{7}$$

In order to take into account eventual preferential pathways, the AMS can be provided with an ordered sequence of spatial points, which are used by the code to create a so-called path. In that case, the importance increases along the path, and decreases with the distance to the path. Those importance functions must be used with caution, as they do not take into account the direction nor the energy of the particles. They also overlook most of the geometry details. However, these spatial importances are easily set up and do not require extended knowledge of the system, allowing for an easy parametrization of the AMS method.

B. The INIPOND Module

TRIPOLI-4® has a module that pre-computes importance maps, which is extensively discussed in [4-6]. Designed for the Exponential Transform variance reduction method, the computed importance map can now be used by the AMS algorithm.

The importance map is computed on a spatial and energetic mesh. Given a spatial detector $\mathcal{D}$, the underlying importance function is assumed to be factorized in spatial, angular and energetic parts:

$$I(\mathbf{X}, \mathbf{\Omega}, E) = I_s(\mathbf{X}, g) \times I_a(\mathbf{X}, \mathbf{\Omega}, g) \times I_e(g), \tag{8}$$

where g denotes the energy group containing E. The three parts of the importance are as follows:

$$I_s(\mathbf{X}, g) = \exp\left(- \int_0^{dist(\mathbf{X},\mathcal{D})} K(\mathbf{X} + r.\mathbf{\Omega_0}, g) dr \right), \tag{9}$$

$$I_a(\mathbf{X}, \mathbf{\Omega}, g) = \frac{\Sigma_t(\mathbf{X}, g)}{\Sigma_t(\mathbf{X}, g) - K(\mathbf{X}, g)\mathbf{\Omega}.\mathbf{\Omega_0}}, \tag{10}$$

$$I_e(g) = \frac{1}{\beta + 1} \frac{(E^g_{sup})^{\beta+1} - (E^g_{inf})^{\beta+1}}{E^g_{sup} - E^g_{inf}}, \tag{11}$$

where Σ_t is the total macroscopic cross section and $\mathbf{\Omega_0}$ the direction of interest (related to the slope of the importance map). The values of K are assumed constant for each material and each energy group, and are either derived from a Placzek-like equation as described in [6], or set by the user. E^g_{inf} and E^g_{sup} denote the bounding values for energy group g, and the β parameter is set by the user in order to adjust the global strength of the biasing and define the energetic profile of the importance map at the detector.

C. Volume And Energy Weighting

Regardless of the importance function, each region of the geometry is attributed an weighting factor. Any point within the region will see its importance weighted by this factor. It has a default value of 1 for every volume, but the user can choose to modify this factor for any of the regions involved in the simulation.

2. AMS Optimization

A. Crossing Points

Most of the time, the geometries considered are composed of many regions, each of these regions having specific properties that impact the particle transport (travel length, collision probabilities,...). When a particle passes from one region to another during a flight, it can be stopped as it crosses the interface between the regions, and a new flight length is resampled from this crossing point, taking into account the properties of the new region.

In that case, the characteristics of the particle at the crossing point depend only of the state of the particle coming out of the last collision and the physical properties of the first region. Similarly, the next collision point can be determined based solely on the characteristics of the particle at the crossing point and the physical properties of the second region.

Consequently, if the crossing points are added to the particle track just as real collision points, the enriched track remains a Markov chain. The AMS can then be used with the enriched tracks without modifications [2]. This allows for a more precise estimation of tracks importances, adding an importance estimation between some collision points without requiring additional computing.

B. Track Storage

One of the downsides of the AMS algorithm is that it requires to keep the particle tracks accessible in memory at all times. However, it is not mandatory for the algorithm to store every point of the tracks.

The points composing the tracks are used in two ways during the AMS iterations:

- To compute the importance of the track they are a part of
- To define the splitting points during the resampling process

In practice, a value of importance is assigned to each of the tracks and updated at each collision point, thus obviating the need to search for the maximum of importance amongst the track points.

According to (4), the splitting point is defined as the first point of the track which importance is greater than the current AMS splitting level. This means that the only points that may be chosen for resampling are those having an importance greater than the importance of every previous points of their track. The importance has to be estimated at every collision point, but a point is stored in the track only if its importance is greater than the importance of the track. In that case, the importance of the track itself is updated.

IV. 3D NEUTRON STREAMING PROBLEM

1. Problem Description

The analysis we propose here is the study of neutrons streaming through a three-dimensional labyrinth filled with air, and located in a concrete cube of side 15 m. The concrete is an ilmenite-limonite concrete of density $2.9 g/cm^3$ and the air has a density of $0.001293 g/cm^3$. The labyrinth is shown in Fig. 1, and the elemental composition of both air and concrete are detailed in Table I.

The neutron source is an 2-MeV isotropic point source placed at the center of a cubic room of side 4 m, and is represented by a blue point on Fig. 1. The entrance of the labyrinth is located at a corner of this room. The labyrinth itself is a 42-meters-long tunnel composed of seven straight sections and six 90° bends, having a rectangle cross section of dimensions 3 m × 50 cm. Two types of tallies are scored in that geometry. The first tally consists in estimating the flux on a mesh covering the entire geometry, using a track length estimator. The second one is an average surface flux tally, estimated on surfaces placed at various distances along the tunnel and at the very end of the labyrinth. The chosen surfaces are the interfaces between the square sections composing the labyrinth,

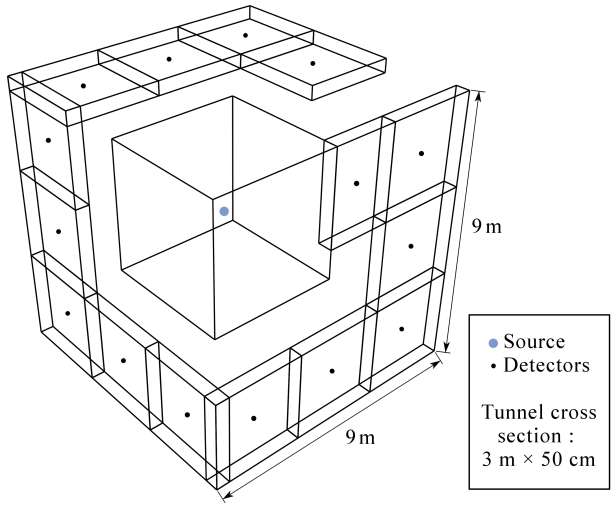

Fig. 1. The labyrinth geometry.

Element	Air	Concrete
Hydrogen	-	00.66
Oxygen	21.00	36.45
Sodium	79.00	-
Magnesium	-	00.15
Aluminium	-	00.80
Silicon	-	03.06
Sulfur	-	00.08
Calcium	-	05.83
Titanium	-	16.03
Iron	-	36.93

TABLE I. Elemental composition of the simulation materials as a percentage by weight.

which centers are marked by black points in Fig. 1. In order for the AMS to simulate particles contributing to all of these tallies, we chose as detector the surface located at the end of the tunnel, and will refer to it as such for the remaining of this work.

2. Spatial Importance

A. Construction Of A Spatial Importance

As stated in Section III., TRIPOLI-4® can be provided with a spatial function to serve as importance for the AMS algorithm. Given the problem at stakes, it seems highly probable that the particles contributing to the flux at the detector are the one that travel along the tunnel. Therefore, we chose in first approximation an importance function with the following characteristics:

- In the source room, the importance decreases with the distance to the labyrinth entrance.
- In the tunnel, the importance increases between the entrance of the labyrinth and the end of the tunnel.
- In the concrete, the importance is zero.

In practice, a path is given to TRIPOLI-4® as the sequence of source and black points displayed of Figure 1 (See Section III. for details), and the weighting factor of the volume corresponding to the concrete is set to 0. The resulting importance function is shown in Figure 2

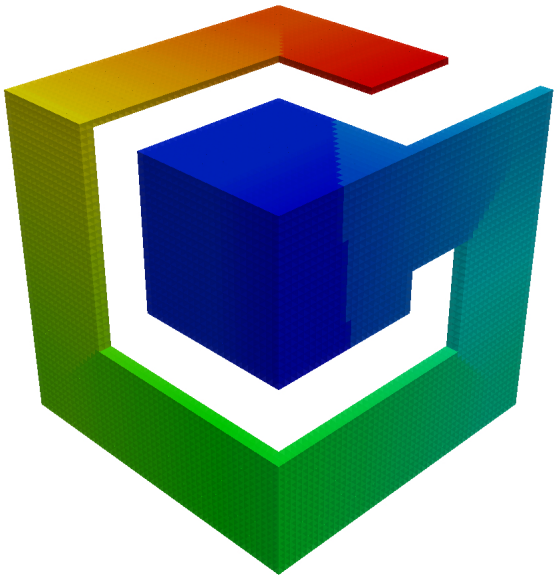

Fig. 2. Spatial importance for the labyrinth geometry.

B. Mesh Tally

Figure 3 shows the flux obtained on the mesh tally covering the labyrinth with the AMS simulation, using the spatial importance as importance function and running the algorithm for 90 minutes on a single core. We can see on Fig. 3 that the AMS algorithm allows TRIPOLI-4® to estimate a flux in every cell of the mesh from the source point to the end of the labyrinth, despite the 20 orders of magnitude attenuation. Furthermore, we notice that the estimation of the flux outside the tunnel is not impaired by the zero-valued importance of the concrete. This is not surprising, as the AMS algorithm does not modify the transport of the particles in the geometry, but merely resample particles selected by the importance map.

C. Surface Flux Tally

In order to validate the results obtained with AMS on the surface flux tally, an analog TRIPOLI-4® simulation was also performed. Both simulations ran on the same computer for 67 hours, each on a single core.
The neutron flux was estimated at the surfaces defined in Sec-

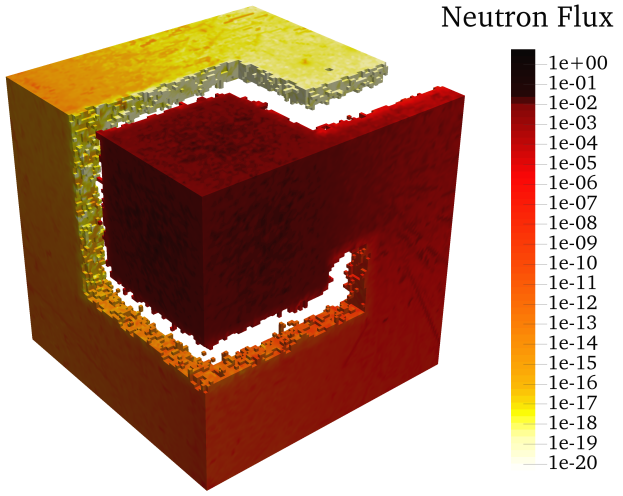

Fig. 3. Neutron flux obtained with AMS on a mesh tally using the spatial importance function.

tion 1., for both analog and AMS TRIPOLI-4® simulations. The results obtained are shown in Figure 4 with respect to the distance between the surfaces and the labyrinth entrance. The dotted vertical lines indicate the locations of the 90° bends in the tunnel. The relative error on the estimation is reported in Figure 5 for both simulations.

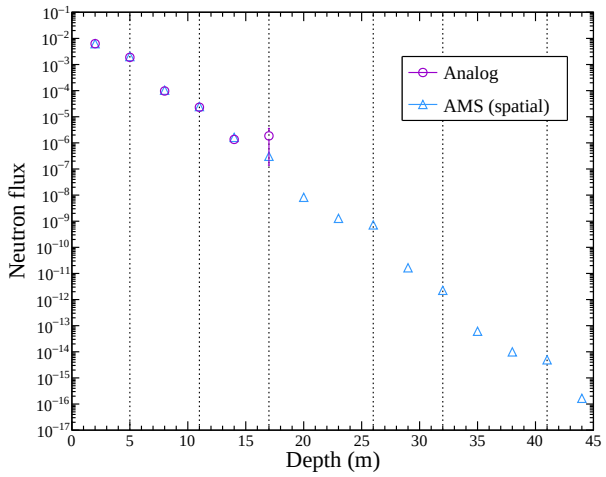

Fig. 4. Neutron surface flux obtained with AMS and analog TRIPOLI-4® along the labyrinth.

The comparison of the results obtained with the surface flux tallies puts into light the efficiency of the AMS algorithm. The analog simulation fails to accurately estimate a flux deeper than 16 meters into the tunnel. At a lower depth, we can see that the results of the AMS simulation are in very good agreement with the values obtained from the analog simulation.

The relative standard deviations for each of the surface flux estimation are reported in Figure 5. It is worth noting that the errors on the first two surfaces are smaller for the analog simulation than with AMS. This is a consequence of the extra computational time required for the AMS simulation to classify and resample particles, which is penalizing if the score is too easy to estimate.

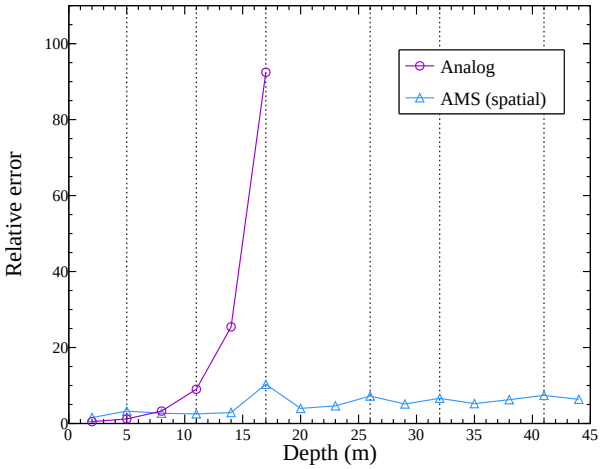

Fig. 5. Percent relative error on the neutron surface flux for AMS and analog TRIPOLI-4® along the labyrinth.

3. Automatically Generated Importance

A. INIPOND Parametrization

In order to have a more refined description of the problem in the importance map, we wish to take the energy and direction into account. To that end, we used the INIPOND module of TRIPOLI-4® to build a discretized importance map.

The spatial mesh is composed of $50 \times 50 \times 50$ cells. The dimensions of the cells are approximately $40cmx10cmx10cm$ in the tunnel, and their size increases progressively with the distance to the labyrinth. The energetic domain of the simulation (20 to $1.E^{-11}$MeV) is divided in 6 energy groups, detailed in Table II. The Placzek coefficients are left to be computed by INIPOND, and the β parameter is set to 1 (we refer the reader to Section III. for details).

The importance map obtained for energy group 2 is shown in Figure 6. We can see that the INIPOND module successfully evaluated the importance map for the entire geometry, also computing an importance for each cell within the concrete. The course and variations of the importance function is the same inside each of the energy groups. The only notable difference between the groups is the span value of the importance. We show in Figure 7 the importance of various cells along the tunnel for each of the energy groups, evaluated in the direction of the tunnel.

B. Comparison On The Surface Flux Tally

The simulation leading to the results presented in this sections has been performed at the same time as the AMS with

Group	Maximal Energy	Minimal Energy
1	20 MeV	1 MeV
2	1 MeV	100 keV
3	100 keV	5 keV
4	5 keV	0.625 eV
5	0.625 eV	1.E-3 eV
6	1.E-3 eV	1.E-3 eV

TABLE II. Energetic discretization of the importance map.

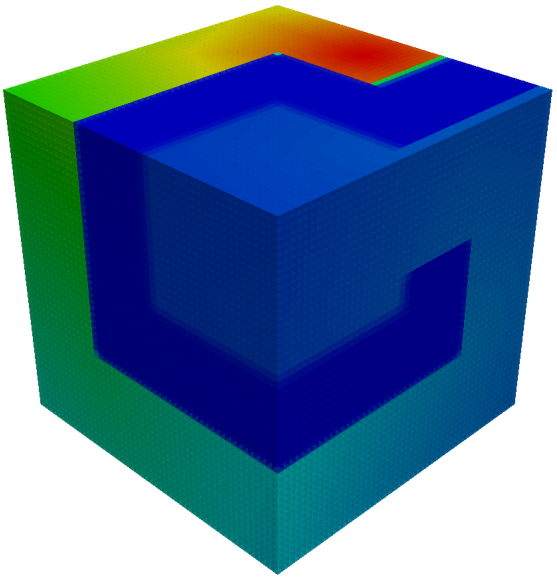

Fig. 6. Automatically generated importance map for energies between 2 MeV and 0.1 MeV.

spatial importance and the analog simulation discussed in the previous section. Every simulation ran on the same computer for 67 hours, each on a single core. Figure 8 shows the comparison between the results obtained with both importance functions.

The surface flux estimated by the two AMS simulations are in very good agreement for each of the considered flux. However, the variance on the estimated values for the AMS using INIPOND's importance is overall greater than when the purely spatial importance function is used, as shown in Figure 9. The most impacting parameter varying between the two importance functions is the energy. Therefore, it seems that the energetic component of the INIPOND importance map is not correctly taken into account.

A further analysis of the energetic profile of the map, as displayed in Figure 7, shows that high-energy groups have an over-estimated importance. Indeed, the importance of a neutron located near the entrance of the tunnel in an energy group seems to be greater than the importance of any point within a lower energy group (at least in the center of the tunnel and in the right direction). This means that high-energy neutrons may be favored over neutrons of lower energy, even if

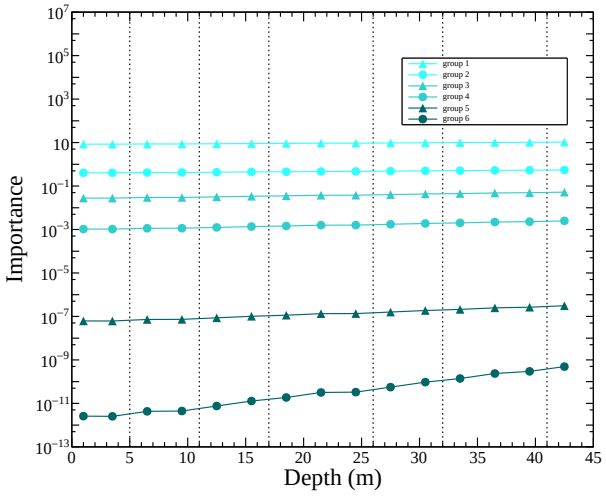

Fig. 7. Energetic profile of the importance map along the tunnel.

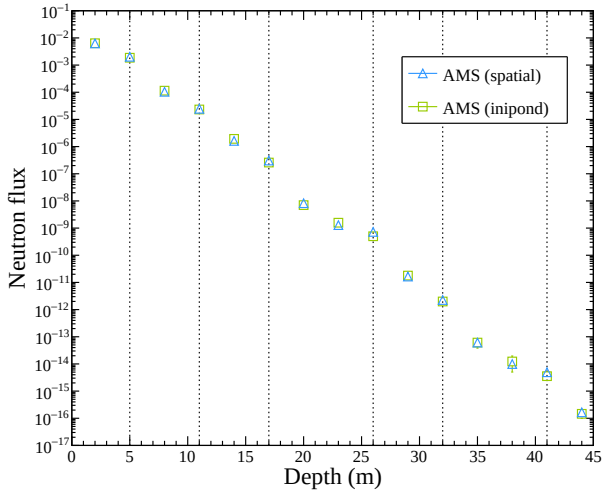

Fig. 8. Comparison of AMS results for the spatial and INIPOND importance functions.

these low-energy neutrons are much closer to the detector area. As a consequence, the AMS algorithm using this importance map will sometimes do unnecessary resampling of interesting neutrons.

On the other hand, the purely spatial importance lets the neutrons energy decrease "naturally" while going further into the tunnel from one iteration to the next. The INIPOND module was designed to provide an importance map adapted to the use of the Exponential Biasing variance reduction method. This method does not compare particles belonging to different energy groups, as does the AMS algorithm. Therefore, the difference of importance between two groups has a specific meaning for each of the methods.

It must be borne in mind that, though obviously not optimal, the INIPOND map does enable the AMS algorithm to estimate the correct average flux on all considered tallies.

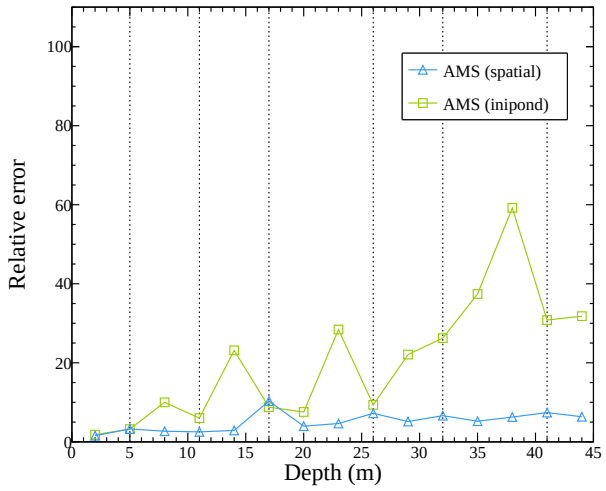

Fig. 9. Comparison of the percent relative errors on the AMS results for the spatial and INIPOND importance functions.

Therefore, the use of the INIPOND map for AMS remains a viable option, and may prove itself very efficient in configurations where mere trivial importance functions are not enough to describe the problem.

It is also worth noting that the use of the same INIPOND map for Exponential Transform did not yield any results. We tried several parametrizations of the module, but were enable to estimate anything more than the results obtained with an analog simulation.

4. Construction Of Reference Values

Due to the complexity of the problem considered, we were unable to get a reference value for the flux using the Exponential Transform variance reduction method. As discussed above, an analog calculation, even run on a multiple processors for several weeks, would not provide a result all the way to the end of the tunnel.

A deterministic approach was considered, but the size of the geometry and the low concentration of the air inside the tunnel are most likely to lead to difficulties calculating the flux, or strong ray effects on the result.

A. Surface Particle Restart

The adopted strategy uses a two-step feature of TRIPOLI-4®'s. In a first simulation, the characteristics of particles crossing a list of boundary is stored and dumped into a file. Then, TRIPOLI-4® uses this file to initialize the source particles in other simulations.

In our case, the geometry was divided in eight parts, the first one containing the source room, and each of the next parts encompassing consecutive bits of the tunnel. After restricting the geometry to its first part, 10^6 particles were simulated from the source point (without variance reduction). The particles states were stored at the interface between the first and the second part of the geometry. Then, seven successive simulations were performed, each of them restricted to a single part of the geometry. Each time, 10^6 particles were simulated using the particle states stored during the previous simulation as source and the particles going out of the restricted geometry were stored to be used in the next simulation.

Obviously, less than 10^6 particles were stored at the end of each partial simulation. Consequently, the particles has been massively duplicated, and the flux estimation may suffer from correlations. Furthermore, the distribution of sources at the interface between geometry parts is stochastic, and they may not accurately describe the real distribution of particles crossing those surfaces. The estimation of the variance on the estimated scores in this configuration is a complicated problem, which has not been tackled in this paper. However, this strategy enabled us to obtain some reference values for the surface flux along the tunnel.

B. Validation Of AMS Results

Figure 10 shows the surface flux estimated with both AMS simulations (spatial in blue and INIPOND in green), as well as the reference values estimated using the surface particle restart strategy presented above, represented as a dashed gray line. The three estimations of each surface flux are in perfect agreement.

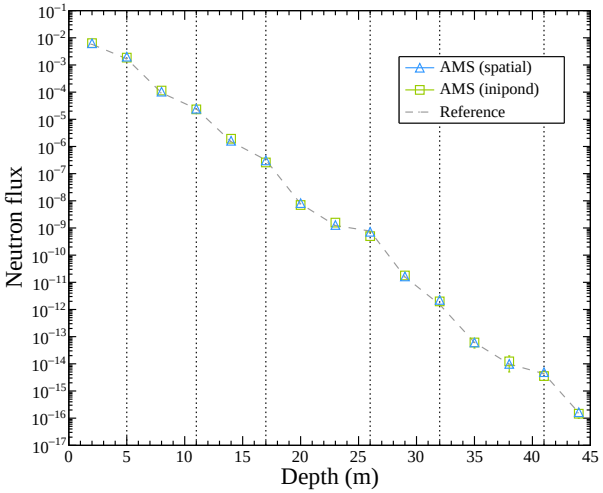

Fig. 10. Comparison of AMS and reference results along the labyrinth.

V. CONCLUSION

The work presented in this paper revolves around a very complex 3D-Streaming problem. The geometry presents very sharp bends in multiple directions. The geometrical properties of the problem makes the use of analog Monte Carlo unreasonable, and will force most variance reduction schemes to use very fine tuning in order to get a proper result. Using Adaptive Multilevel Splitting, TRIPOLI-4® was able to estimate a neutron flux in the this geometry using two simple parametrizations.

This work did also put into light the possibility of improvement for the automated importance map generator of

TRIPOLI-4®Ṫhe current state of the INIPOND module could be made more suitable for AMS use, either by changing the construction of the energetic component, or by adjusting the importance map during the simulation. In any case, the current implementation of AMS in TRIPOLI-4® is already a very efficient tool, both in terms of variance reduction and ease of use.

VI. ACKNOWLEDGMENTS

TRIPOLI-4® is a registered trademark of CEA. We gratefully acknowledge EDF for their long term partnership and their support.

REFERENCES

1. CÉROU, F. and GUYADER, A., "Adaptive Multilevel Splitting for Rare Event Analysis", *Stochastic Analysis and Applications*, **25**, *2*, 417 (2007)
2. C.-E. Bréhier, M. Gazeau, L. Goudenège, T. Lelièvre, M. Rousset, "Unbiasedness of some generalized Adaptive Multilevel Splitting algorithms", *Ann. App. Prob.*, **26**, **6**, 3559 (2016)
3. LOUVIN, H., DUMONTEIL, E., LELIÈVRE, T., ROUSSET, M. and DIOP, CHEIKH M., "Adaptive Multilevel Splitting for Monte Carlo Particle Transport", *Proc. ICRS 13-RPSD 2016*, Paris, October 3-6 (2016)
4. E. Brun, F. Damian, C.M. Diop, E. Dumonteil, F.X. Hugot, C. Jouanne, Y.K. Lee, F. Malvagi, A. Mazzolo, O. Petit, J.C. Trama, T. Visionneau, A. Zoia, "TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code", *Ann. Nucl. Energ.* **82**, 151 (2015)
5. O. Petit, F.X. Hugot, Y.K. Lee and C. Jouanne, "TRIPOLI-4 Version 4 User Guide", *CEA-R-6169, Commissariat à l'Energie Atomique (CEA)* (2008)
6. J.P. Both, J.C. Nimal and T. Vergnault, "Automated importance generation and biasing techniques for Monte Carlo shielding techniques by the TRIPOLI-3 code", *Progress in Nuclear Energy*, **24**, **1-3**, 273 (1990)