EPJ Web of Conferences **153**, 06006 (2017) DOI: 10.1051/epjconf/201715306006
ICRS-13 & RPSD-2016

Adaptive Multilevel Splitting for Monte Carlo particle transport

Henri Louvin[1,⋆], Eric Dumonteil[2], Tony Lelièvre[3], Mathias Rousset[3], and Cheikh M. Diop[1]

[1] *CEA Saclay, DEN/DM2S/SERMA/LTSD, F-91191 Gif-sur-Yvette*
[2] *IRSN, PSN-EXP/SNC, F-92262 Fontenay-aux-Roses*
[3] *Université Paris-Est, CERMICS (ENPC), INRIA, F-77455 Marne-la-Vallée*

Abstract. In the Monte Carlo simulation of particle transport, and especially for shielding applications, variance reduction techniques are widely used to help simulate realisations of rare events and reduce the relative errors on the estimated scores for a given computation time. Adaptive Multilevel Splitting is one of these variance reduction techniques that has recently appeared in the literature. In the present paper, we propose an alternative version of the AMS algortihm, adapted for the first time to the field of particle tranport. Within this context, it can be used to build an unbiased estimator of any quantity associated with particle tracks, such as flux, reaction rates or even non-Boltzmann tallies. Furthermore, the efficiency of the AMS algorithm is shown not to be very sensitive to variations of its input parameters, which makes it capable of significant variance reduction without requiring extended user effort.

1 Introduction

The challenge in using Monte Carlo particle transport simulations for shielding applications is to minimize the computation time required to attain a reasonable variance on the quantity of interest, called *score*.

The basic approach of variance reduction techniques is to modify the simulation behaviour so as to increase rare events occurrence while keeping an unbiased estimator of the score.

In this view, *multilevel splitting techniques* were introduced to the field of particle transport by Kahn and Harris [1]. The principle of these techniques is to increase the number of simulated particles when approaching areas of interest of the geometry. Practically, the simulated space is divided into regions of importance, and the particles that pass from a less important to a more important region are duplicated. Each of the duplicated particles is given half the weight of the original to ensure that the simulation remains unbiased. Thus, more computation time is spent to simulate interesting particles rather than new particles from the source.

The downside of these techniques is that they require a fair knowledge of the system in order to accurately define the importance regions. More recently, a new method called Adaptive Multilevel Splitting (or AMS) has been proposed by Cérou and Guyader [2], and studied in a more general setting by Bréhier et al. [3]. This method also aims to duplicate the interesting particles of the simulation, but uses the knowledge accumulated during the simulation itself to determin on the fly which particle should be duplicated and at which point. One of the most interesting feature of AMS is that it requires little knowledge of the system to yield interesting results. The efficiency of the AMS in Monte Carlo simulations and its properties makes it attractive for computational physics problems that require precise rare event simulation. To this end, AMS was successfully extended to molecular dynamics simulations by Aristoff et al. for the resampling of reactive paths [4].

In this paper, we aim to apply the AMS algorithm to Monte Carlo particle transport, and demonstrate its efficiency for rare event simulations. In Section 2, we will describe a mathematical version of the AMS algorithm specifically designed to fit the requirements of particle transport. We introduce in Section 3.1 the context of the study, which is neutral particle transport with the Monte Carlo method. The core of this work is presented in Section 4, in which we introduce for the first time a pratical implementation of AMS within a Monte Carlo particle transport simulation. This version of the AMS algorithm was implemented in the development version of the Monte Carlo particle transport code TRIPOLI-4®. In the last Section of this paper, we present some of the results obtained using AMS within TRIPOLI-4®. These examples illustrate the validity of the algorithm as described in Section 4, as well as the efficiency of AMS as a variance reduction technique.

2 Mathematical setting

We present in this section a specific mathematical setting of the AMS algorithm, which will be afterwards fairly easy to adapt to the context of particle transport. This version

⋆e-mail: henri.louvin@cea.fr

of the AMS is a variant of the algorithm which perfectly fits the theoretical frame from [3], so that the estimator of the rare event occurence probability introduced in the following is unbiased.

2.1 Objective and setup

Let $X = (X_t)_{t\in\mathbb{N}}$ be a discrete-time Markov chain with values in $\mathbb{R}^d$, $d \in \mathbb{N}^*$. We define $\mathcal{P}$ as the path space, containing all possible realisations of the Markov chain X. Given T a subset of $\mathbb{R}^d$, we define the entrance time of the Markov chain X in T:

$$\tau_T = \inf\{t \in [0;\tau_f] : X_t \in T\},$$

where τ_f is the final stopping time of the Markov chain X, which we suppose finite.

Given an observable $\phi_T : \mathcal{P} \to \mathbb{R}$ such as $\phi_T = 0$ on the event $\tau_f < \tau_T$, we would like to estimate the average $\langle\phi_T\rangle$ of ϕ_T. Let us suppose now that the probability for (X_t) to enter T before τ_f is very small, i.e. $\mathbb{P}(\tau_T < \tau_f) \ll 1$. In such a case of rare event simulation, the AMS algorithm proposes to reduce the variance on the estimated average by increasing the number of Markov chains reaching the subset T. AMS is an iterative algorithm that consists of several steps. The first step is a basic Monte Carlo simulation of multiple replicas of the Markov chain X with a common initial state. Each of the following steps consists in the resampling of some Markov chains amongst the replicas, with an initial condition getting closer to T at each iteration of the algorithm.

2.2 AMS algorithm

2.2.1 Importance function

We define a so-called *importance function*, mapping $\mathbb{R}^d$ to $\mathbb{R}$:

$$\xi : \mathbb{R}^d \to \mathbb{R},$$

which is used to quantify the proximity of a point in $\mathbb{R}^d$ to the subset of interest T. The only requirement imposed on ξ is that there exists a constant $Z_T \in \mathbb{R}$ such that

$$\begin{aligned} \xi(x) &\geq Z_T \ \text{ if } \ x \in T \\ \xi(x) &< Z_T \ \text{ if } \ x \notin T. \end{aligned} \tag{1}$$

We further define the *importance* of a Markov chain $X = (X_t)_{t\in\mathbb{N}}$ as the supremum of ξ along the chain:

$$\Xi(X) = \sup_{t\in\mathbb{N}} \xi(X_t). \tag{2}$$

The ξ function is probably the most important ingredient of the AMS algorithm, since it is used to quantify the proximity of a path to the subset T. It is therefore important to choose a good function ξ with regards to the rare event we are trying to simulate. Even if the AMS algorithm is proven to yield an unbiased result *regardless of* ξ, the choice of an optimized importance function is expected to improve the variance reduction efficiency. We will discuss further the issue of the quality of the function ξ in Sect. 5.

2.2.2 Initialization

The AMS algorithm consists in an interacting system of weighted replicas. Given $n > 0$, we simulate n i.i.d. replicas of the Markov chain X, denoted by $X_0^j = (X_{0,t}^j)_{t\in\mathbb{N}}$, $j \in \{1,\dots,n\}$. For all j, the initial state $X_{0,0}^j$ is a point x_0 located outside of T. We define Z_0 as

$$Z_0 = \xi(x_0).$$

Let us denote by $q \in \mathbb{N}$ the current iteration number, and by k the number of replicas resampled at each iteration.Within an iteration of the AMS algorithm, every replica has *the same weight*. The global weight at iteration q will be denoted W_q, with W_0 set to 1.

For the sake of simplicity, we present in the following the case where distinct paths cannot have the same importance. The general case is discussed in Section 2.3. Now we can start iterating on $q \geq 0$.

2.2.3 Iterations

(i) For each $j \in \{1,\dots,n\}$, denote by $\tau_{q,f}^j$ the stopping time of the Markov chain X_q^j, and by S_q^j its importance (See (2):

$$S_q^j = \Xi(X_q^j). \tag{3}$$

(ii) Sort the sample $(S_q^1,\dots,S_q^n)$ in increasing order:

$$S_q^{(1)} < \dots < S_q^{(k)} < \dots < S_q^{(n)}.$$

(iii) Denote by Z_{q+1} the k-th order statistics $S_q^{(k)}$:

$$Z_{q+1} := S_q^{(k)}.$$

(iv) If $Z_{q+1} \geq Z_T$, stop iterating and go to the final step (See Sect. 2.2.5)

(v) For all $j \in \{1,\dots,k\}$, resample replica $X_q^{(j)}$ according to the resampling kernel described in Sect. 2.2.4, and denote it by X_{q+1}^j

(vi) For all $j \in \{k+1\dots,n\}$, define $X_{q+1}^j = X_q^{(j)}$

(vii) Update the global weight:

$$W_{q+1} = \frac{n-k}{n} W_q$$

(viii) Increment q and go to step (i)

2.2.4 Resampling process

At each iteration of the AMS algorithm, k replicas are resampled. These Markov chains undergo the following resampling procedure:

Let us denote the current iteration number by q and the associated resampling level by Z_{q+1}.

For each of the k replicas $X^{(j)}, j \in \{1,\dots,k\}$ to be resampled, one of the remaining replica $X^{(i)}, i > k$ is *randomly* selected for duplication. We know for sure that the

Markov chain $(X_t^{(i)})_{t\in[0;\tau_f^{(i)}]}$ contains at least a state whose importance is greater than Z_{q+1} (otherwise it would have been resampled). We can therefore define

$$\tau_q^{(i)} = \inf\left\{t \in [0;\tau_{q,f}^{(i)}] : \xi(X_{q,t}^{(i)}) > Z_{q+1}\right\}$$

as the first time at which the replica $X^{(i)}$ has an importance greater than Z_{q+1}.

The resampled Markov chain $(Y_t)_{t\geq 0}$ is defined as a replica of the Markov chain X with initial condition

$$Y_0 = X^{(i)}_{q,\tau_q^{(i)}}.$$

2.2.5 Final step

Once the algorithm stops iterating, we denote by N the number of completed iterations. Given the bounded observable ϕ_T introduced in Sect. 2.1, we define

$$\hat{\phi}_T = \sum_{j=1}^{n} W_N^j \phi_T(X_N^j) \tag{4}$$

as an **unbiased** estimator of the average $\langle\phi_T\rangle$ [3].

2.3 Interpretation of the replicas weights

In this section we provide a practical interpretation of the AMS weights to give an intuition on the estimator (4). The mathematical proofs of the unbiasedness and consistency of the AMS estimator are not presented in this paper. We refer the reader to [2] and [3] for theoretical support.

At each iteration $q \geq 0$, the level Z_{q+1} is chosen in such a way that the probability for a path X_q^j to have an importance greater than Z_{q+1} (i.e. $\mathbb{P}(S^j_q \geq Z_{q+1}|S^j_q \geq Z_q)$) is estimated by

$$\hat{p}_q = 1 - \frac{k}{n}.$$

Keeping that in mind, we can see that the weights of the replicas at iteration $q+1$ are nothing more than an estimate of the probability $\mathbb{P}(S^j_{q+1} \geq Z_{q+1})$.

In other words, the AMS algorithm provides us at each iteration q with a set of paths X_q^j, $j \in \{1,\ldots,n\}$, carrying a global weight, which can be interpreted as an estimate of the probability to have this particular set of paths instead of the paths sampled at iteration 0.

2.4 About the number of resampled replicas

The algorithm presented in Sect. 2.2 is an ideal case. In reality, it may occur that multiple replicas have the same importance. In that case, the number of replicas having an importance less or equal to the k-th lowest importance may very well be greater than k. When such a situation arises, **every** path which importance is less or equal to the level has to be resampled.
This modification has to be taken into account in the replicas weights. If the current iteration is the q-th and the number of particles to be resampled is $K' > k$, then the weight update at step (vii) of the algorithm has to be changed to:

$$W_{q+1} = \frac{n-K'}{n} W_q.$$

In some pathological configurations, all the replicas may happen to have the same importance, which would lead to extinction at the next iteration. In that case, the iterating process is interrupted and the algorithm goes straight to the final step (See Sect. 2.2.5), eventually yielding a null contribution.

3 Monte Carlo neutral particle transport

3.1 Neutron transport theory

The transport theory describes the mathematical framework needed to solve the equations governing the behavior of a gaz of non-interacting particles (e.g. neutrons, photons) in different materials, under different assumptions (spins are not taken into account, relativistic effects are neglected etc.). It has been discussed at length in [6] or [7]. In fissile materials, neutrons (and photons) can induce fissions, which results in the production of more particles. The modelling of such systems is achieved using the so-called critical linear Boltzmann equation. When the materials are not fissile, the standard linear Boltzmann equation can be used to describe the particle transport.

It is assumed that the neutral particles in homogeneous medium are transported along straight lines between collisions points and are randomly reoriented at each interaction (with eventual change of energy) [8]. As the flights lengths are exponentially ditributed, the underlying stochastic process governing the neutral particle transport is called *exponential flights*. We refer the interested reader to [9, 10] for recent developments on this topic.

3.2 Monte Carlo simulation of neutral particle transport

Numerically, transport problems are stochastically solved using Monte Carlo transport codes [11]. Those codes simulate directly the particles trajectories according to laws of probability provided by the so-called nuclear data (cross sections, energetic and angular transfers, secondary particles emission spectrums, etc.), which are available in international libraries. The flights lengths between interactions are randomly sampled, as well as the outgoing direction and energy of the particle after each collision with the medium.

Every Monte Carlo transport code relies on the particle tracking routine, which simulates the random life of the particles by transporting them from one interaction to another, until they are absorbed or leak out of the geometry.

Within a Monte Carlo simulation, the life of each transported particle is build step by step, which means that the state of the particle at a given time is determined exclusively from the knowledge of its state after the previous interaction. This makes the random process of particle transport a discrete-time markovian process. Therefore, the sequence of the phase-space coordinates of the particle (namely position, direction and energy) at the successive interactions is a Markov chain, so that the AMS

algorithm described in Section 2 can be implemented.

4 Neutral particle transport with AMS

The goal of this section is to present a practical implementation of the AMS algorithm in the context of Monte Carlo particle transport.

4.1 Definitions and notations

Before going any further, let us introduce some definitions and notations which will be used throughout the following.

We define the position of a particle in the 6-dimensional *phase space* $\mathcal{S}$ as the set of coordinates $(\mathbf{X}, \mathbf{\Omega}, E)$, where $\mathbf{X}$ denotes the particle position in the 3-dimensional space, $\mathbf{\Omega}$ its direction, and E its energy.

A particle is considered *alive* while it is transported from an interaction to the next. When an interaction results in the capture of the particle, the transport stops and the particle is referred to as *killed*. We call *track* of a particle the sequence of its interaction points.

We call *geometry* the subspace of the phase space in which the simulation takes place. If a particle is transported to a point of the phase space that is not included in the geometry, this particle is considered leaking out of the geometry and is instantly killed.

The aim of the simulation is to estimate a *score* (typically a flux or a reaction rate) in a particular volume of the geometry we will refer to as *detector* or simply *target*.

4.2 Importance map

As we saw in Sect. 2.2.1, the geometry has to be provided with an importance function, so as to determine the regions of the system that are of interest to the simulation. Technically, we will use a function that associates any point of the phase space to an importance value, which is related to the probability for a particle located at this point to contribute to the final score. We will refer to this function as *importance function* or *importance map*, and denote it by

$$I(\mathbf{X}, \mathbf{\Omega}, E).$$

In order to satisfy the requirement (1), it is sufficient to define an importance I_{target} larger than any other value on the map, and to assign it to every point of the target volume. In TRIPOLI-4®, I_{target} is set to the numerical equivalent of positive infinity regardless of the importance values outside the target volume, so as to avoid mistakes.

4.3 The AMS algorithm

As described previously, the AMS algorithm is an iterative method. Each iteration includes the definition of a single splitting level alongside with the corresponding splitting process.

4.3.1 Initialization (step $q = 0$)

Let n be the initial number of simulated particles, and k the **user-defined** number of particles that are to be duplicated at each iteration of the algorithm. The initial particles are simulated using an analog Monte Carlo simulation (*i.e.* without variance reduction), and the track of each transported particle is kept in memory.

There are actually many ways to store a particle track. For the AMS algorithm to be the most performant, this track should only contain the emission point of the particle and the coordinates of the particle after each interacion with the geometry. The coordinates of a particle at an interaction point but before the interaction takes place should not be stored, as the importance function relate to the probability for a particle simulated from a given point of the phase space to contribute to the score, and that probability usually takes into account the fact that the particle is transported before interacting with the medium.

The global weight at iteration q is denoted by w_q, and w_0 is set to 1. We can now start iterating on $q \geq 0$.

4.3.2 Iterations ($q \geq 0$)

Each track T is given an importance value, defined as the maximum importance of the points of its track:

$$I(T) = \max_{i \in track} I(\mathbf{X}_i, \mathbf{\Omega}_i, E_i). \tag{5}$$

The n tracks are sorted according to their importance, and the k-th smallest track importance defines the splitting level I_{q+1}.

If I_{q+1} is greater than I_{target}, the iterating process stops. Otherwise, all the tracks that have an importance less or equal to the level I_{q+1} are suppressed, and the same number of tracks are randomly selected (with replacement) amongst the remaining tracks. The selected tracks are duplicated at their first collision point of importance greater than I_{q+1}, according to the resampling kernel presented in Sect. 2.2.4. The global weight is then updated:

$$\begin{aligned} w_{q+1} &= \frac{n - K_q}{n} w_q \\ &= \prod_{i=0}^{q} (1 - \frac{K_i}{n}), \end{aligned}$$

where K_q is the number of resampled particle tracks at iteration q (See Sect. 2.3).

4.3.3 Final step and scoring

As soon as a splitting level I_q is greater than I_{target}, the algorithm stops iterating. We define $N = q - 1$. The score $\hat{\phi}_{target}$ in the target is then computed using the standard estimators of Monte Carlo particle transport, as for an analog simulation, but using the last set of tracks.

An unbiased estimator of the flux in the target is built by weighting the estimated flux $\hat{\phi}_{target}$ by the probability of reaching the last splitting level (See Sect. 2.2.5):

$$\hat{\phi} = \hat{\phi}_{target}.w_N.$$

4.4 Strengths of the AMS

4.4.1 Number of contributions to the score

In order to reduce the variance when simulating rare events, it is interesting to have multiple small contributions to the score rather than a few large contributions and many zeroes.

The stopping criterion of the AMS algorithm offers a guarantee that *at least* $n - k + 1$ particles will reach the target and contribute to the score with a small weight w_N, except for pathological cases. However, the cases of extinction are rare and caused by the use of a very unadapted importance map.

4.4.2 Robustness with regard to the importance function

The estimator constructed as described in Sect. 4.3 is **unbiased** regardless of the parameter k [2]. The same holds true for any importance map under the single constraint [3]:

$$I_{target} \in]I_{source}, +\infty[. \tag{6}$$

In our implementation of the AMS algorithm, this condition is *always verified*, as we set the importance of the target to infinity regardless of the importance map (see Sect. 4.2). This allows us to use any function as importance. However, if the map does not properly describe the system, the result, though unbiased, will probably take a really long time to converge.

4.4.3 Usability

In addition to the usual parameters of any Monte Carlo particle transport simulation, AMS introduces only two free parameters: the number of duplicated particles k, and the importance map $I(\mathbf{X}, \mathbf{\Omega}, E)$.

As we already pointed out, any choice of k and I yields an unbiased result. The parametrization of the AMS algorithm is therefore quite simple. It is sufficient to define a purely geometric importance function, such as the inverse to the distance to the target, and to set any value for k less than n to reduce the variance. More efficiency can be achieved through the use of a precise importance map, taking into account the direction and energy of the particles.

4.4.4 Parallelization

A Monte Carlo simulation performed with AMS yields a single value as estimation of the quantity of interest. Therefore, multiple independant simulations are required to obtain a precise result. The arithmetic mean of the estimations over all simulations is computed, as it is the case in regular Monte Carlo. As the simulations remain totally independant until their output is gathered to compute the average result, a straightforward way of parallelizing AMS Monte Carlo is to run independant simulations on several processors simultaneously.

5 Validation and results

The AMS algorithm was implemented in the development version of the Monte Carlo particle transport code TRIPOLI-4® [5]. This 3D continuous-energy Monte Carlo code is developed by the Service d'Etudes des Réacteurs et de Mathématiques Appliquées (SERMA) at CEA Saclay.

In the following, we present some preliminary results obtained with TRIPOLI-4® using AMS and compare them to simulations with no variance reduction. The Monte Carlo simulations that do not use any variance reduction techniques are usually called *analog*, because they are directly analogous to the natural transport of the particles.

We aim to validate the AMS algorithm implementation and to estimate its efficiency as a variance reduction technique by comparing its results and performances to analog TRIPOLI-4® simulations.

5.1 Figure of merit

In a Monte Carlo simulation, the flux is estimated for several statistical independent groups of particles (or *batches*). The average flux and its associated variance are estimated from the series of batches. The purpose of variance reduction is to reduce the variance of the result for a given computation time. The efficiency of a variance reduction technique lies upon two factors. On one hand, the relative error σ of the average flux, and on the other hand the simulation time T. If we denote by N_B the total number of simulated batches, we have

$$\sigma \sim \frac{1}{\sqrt{N_B}} \quad \text{and} \quad T \sim N_B. \tag{7}$$

We use the figure of merit (FOM) to evaluate the efficiency of a Monte Carlo simulation. The FOM is defined as

$$FOM = \frac{1}{\sigma^2 T}, \tag{8}$$

and should remain relatively constant during the simulation according to (7). For different simulations of the same system, the simulation producing the largest FOM is either proportionally faster or more precise than the others. Therefore, the FOM can be used as a performance measurement for variance reduction.

5.2 Importance map calculation

The AMS algorithm implemented in TRIPOLI-4® can use several functions as importance maps.
The user can define a *purely geometrical* importance, such as the distance between a point and the particle source or the invert of the distance between a point and the AMS target.

Most of the time though, a more precise importance function is required to take into account the direction and energy of the simulated particles. To that end, the AMS has been linked to the so-called `INIPOND` module of

TRIPOLI-4®, which is used for other variance reduction techniques in this code and to automatically pre-computes an importance map (see [5]).

5.3 Bypass simulation

The first system consists of an extruded box filled with Helium. The dimensions of the box are $10cm \times 10cm \times +\infty$. A neutron flux is produced from an 15-MeV isotropic neutron source at one corner of the box and estimated inside an infinite cylinder of 1cm in diameter placed at the opposite corner. Inbetween is a massive infinite cylinder composed of highly concentrated Bore. The mean free path of neutrons in the Bore slab is less than 0.5 cm. The geometry for this problem is shown in Fig. 1.

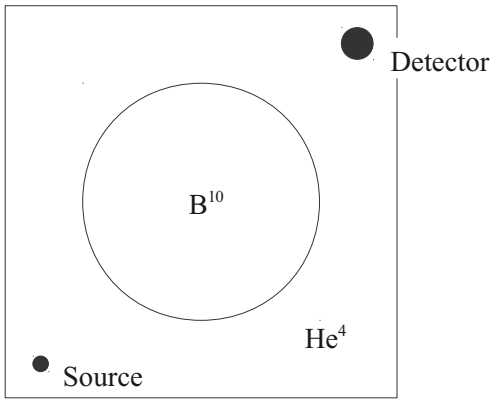

Figure 1. Bypass simulation configuration

The results obtained using TRIPOLI-4® without variance reduction and TRIPOLI-4® using the AMS algorithm are presented in Table 1. Several AMS simulations with 10^4 initial particles were performed, with different values for the parameter k, from 1 duplicated particle (0.01%) per iteration up to 5×10^3 (50%). In all simulations, the `INIPOND` importance map was used.

We observe that the AMS significantly improves the FOM regardless of the parameter k. The number of iterations required for the AMS algorithm to complete increases with k up to $\frac{k}{n} = 0.5\%$. Above this value, the computation time per simulation remains stable and the FOM gain relatively constant.

The smallest relative error for a given number of independent simulations is obtained with $k = 1$ (0.01%). However, this configuration requires a very large number of iterations for the AMS to complete, which has a strong impact on the computation time. It is therefore more interesting (regarding the FOM gain) to chose a greater k in order to speed up the simulation.

The study of the Bypass problem puts into light the apparent robustness of the AMS efficiency with regard to the parameter k. As long as it is not too small, its impact on the algorithm efficiency in reducing the variance remains stable. However, it is to be noted that the geometry of the Bypass configuration is fairly simple. We may find cases in which the number of splitting per iteration have a greater impact on the algorithm efficiency.

5.4 Neutron flux attenuation in water

The source of the second problem is a directional neutron point source, emitting neutrons according to a Watt spectrum and placed at the entry of a straight box filled with water. The detector is placed at 3 m from the source, and the neutron flux is estimated in 10-cm-thick slices between the source and the detector.

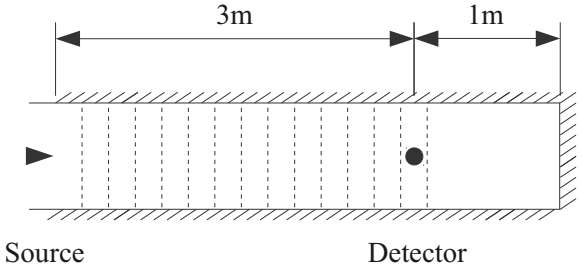

Figure 2. Water box simulation system

The neutron flux attenuation was estimated using both analog and AMS TRIPOLI-4® simulations. In a first realization, the importance function used for the AMS was the distance from the source point. The AMS was performed using 10^3 initial particles per batch and $k = 10^2$ splitting per iteration. Data were collected for both simulations during 65h. Fig. 3 presents the estimated flux with respect to the distance in the box, and the associated 99.7% confidence intervals. The FOM for these simulations are shown in Fig. 4.

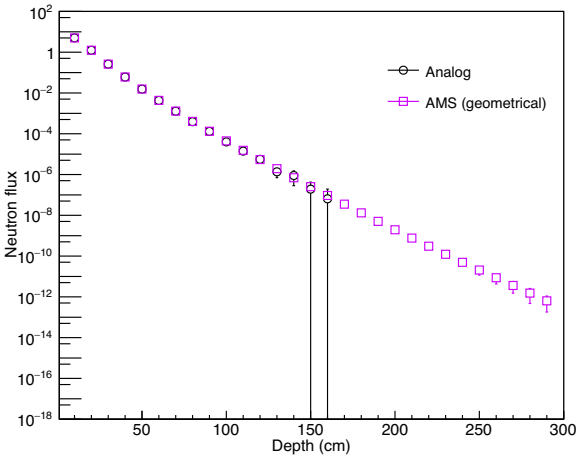

Figure 3. Flux attenuation in the Water box

Table 1. Comparison of TRIPOLI-4® with and without AMS for flux estimation in the Bypass geometry (48h calculations)

Simulation	Number of independent simulations	Mean	Relative error	FOM	Gain
Analog	14300	4.3042×10^{-10}	8.6838×10^{-2}	0.0008	**1**
AMS (0.01%)	210	4.0541×10^{-10}	1.5377×10^{-2}	0.0246	**31**
AMS (0.05%)	1705	4.2244×10^{-10}	7.1384×10^{-3}	0.1135	**147**
AMS (0.1%)	4038	4.1462×10^{-10}	3.8072×10^{-3}	0.3992	**517**
AMS (0.5%)	10170	4.1518×10^{-10}	2.3973×10^{-3}	1.0075	**1307**
AMS (1%)	10900	4.1547×10^{-10}	2.3948×10^{-3}	1.0123	**1313**
AMS (5%)	10884	4.1494×10^{-10}	2.4280×10^{-3}	0.9816	**1273**
AMS (10%)	11472	4.1451×10^{-10}	2.3604×10^{-3}	1.0387	**1347**
AMS (50%)	10829	4.1591×10^{-10}	2.3750×10^{-3}	1.0259	**1331**

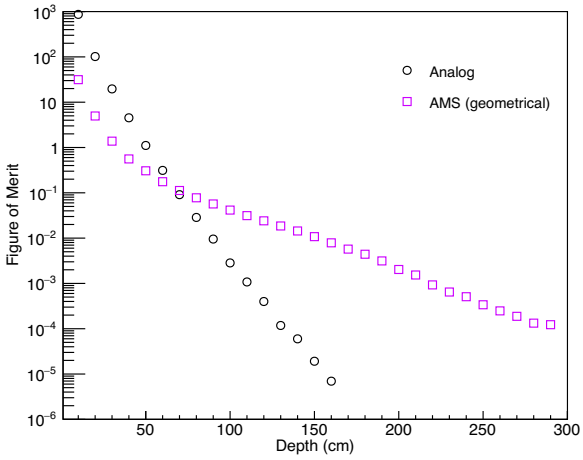

Figure 4. FOM comparison in the Water box

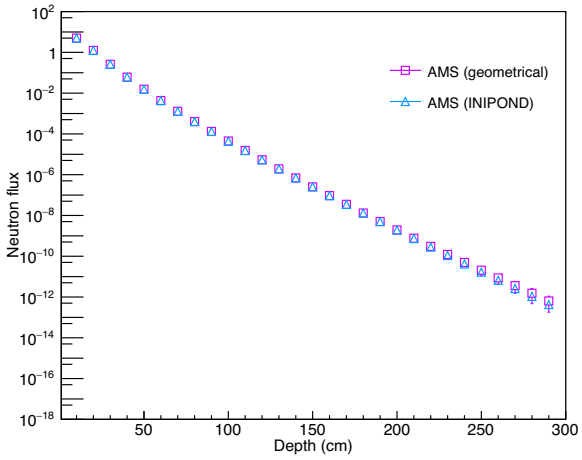

Figure 5. Flux attenuation in the Water box

It can be seen in Fig. 3 that the analog simulation in unable to estimate a flux farther than 160cm from the source, because of the strong attenuation of neutrons in water. However, the AMS simulation is able to accurately determine the neutron flux all the way from the source to the detector. Near the source point, the analog simulation is more efficient than the AMS, as shown in Fig. 4. The neutron flux in this part of the box is high enough for TRIPOLI-4® not to need variance reduction, which places the AMS at a disadvantage due to its impact on runtime.

The AMS mechanism starts to be interesting at a depth of 70cm. The gain in FOM increases up to 10^3 in favor of AMS at 160cm, from which point the analog simulation fails to estimate the flux.

In order to illustrate the role of the importance map within the AMS algorithm, the deep-penetration simulation was performed a second time, using the same parameters for AMS except for the geometrical importance function, which was replaced by the importance estimated using the `INIPOND` module. The fluxes obtained with both importance maps are shown in Fig. 5, and the associated FOM in Fig. 6.

We observe that the use of the `INIPOND` map over the geometrical importance significantly improves the AMS efficiency up to a factor 4 at the target. We can notice that the FOM obtained with both maps are equivalent for depths below 160cm. However, we have to keep in mind that the parametrization of the `INIPOND` module required to get an accurate map can be far more complex than the input of a simple geometrical function.

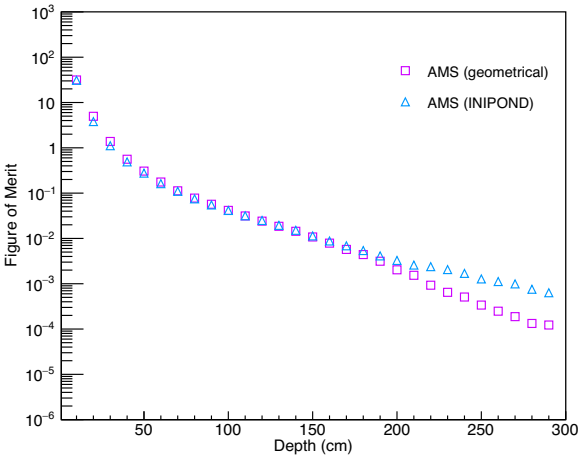

Figure 6. FOM comparison in the Water box

5.5 Gamma spectroscopy

The AMS algorithm has the specificity to provide a group of particle tracks associated to a *global weight* (see Sect. 4). This feature let us hope to use AMS to reduce the variance on deposited scores, such as deposited spectrum (pulse height tally), energy or charge, for which all particles contributing to the score **must** have the same weight.

The use of other variance reduction techniques for quantities that depend on collective effect of a group of particles is a complex problem [12]. Currently, TRIPOLI-4® does not even support variance reduction with deposited scores. However, AMS offers a possibility to have an efficient variance reduction for deposited scores using the same algorithm than for other types of scores.

We present in Fig. 7 a deposited spectrum obtained by simulating the response of a high-purity Germanium (HPGe) detector placed at 50cm of a Sodium source. The gain in efficiency with AMS is shown in Fig. 8.

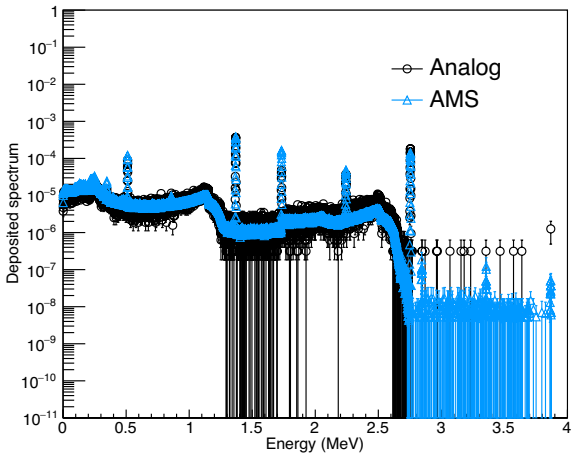

Figure 7. Deposited spectrum in the HPGe detector

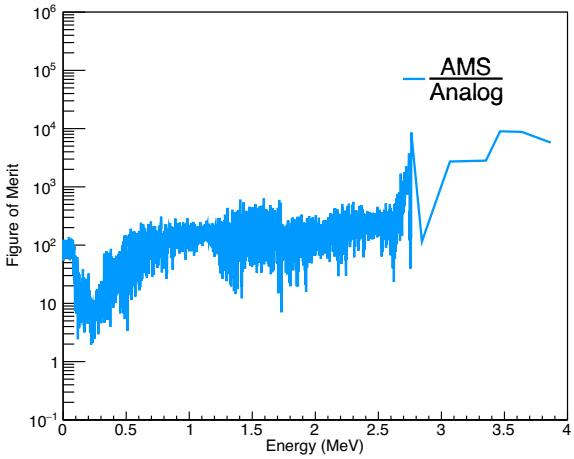

Figure 8. AMS gain in FOM over the deposited spectrum

These results are from photon-only simulations, as the coupled photons/electrons/positrons AMS is not yet fully operational.

6 Conclusion

These preliminary results show that the use of the Adaptive Multilevel Splitting algorithm in TRIPOLI-4® reduces significatively the variance. Its connection to the TRIPOLI-4® importance map calculation module enables us to use advanced parametrizations for the importance. The guarantee to have an **unbiased result** regardless of the importance map or the number of particles resampled at each iteration makes the AMS algorithm a viable alternative to existing variance reduction techniques in the most severe configurations. Furthermore, it can also be used for variance reduction on deposited scores without need to alterate the algorithm, as seen in Sect 5.5.

In the future, we plan to implement AMS variance reduction on deposited scores using coupled photons/electrons/positrons simulations as well as coupled neutrons/gamma problems.

Another concern is the comparison of the AMS efficiency to existing variance reduction techniques. A paper by the same authors is in preparation regarding this study [13].

Aknowledgement

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

The work of T. Lelièvre and M. Rousset is supported by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement number 614492.

References

[1] H. Kahn, T.E. Harris, Nat. Bur. Stand. Appl. Math. Series, *Estimation of particle transmission by random sampling*, **12**:27-30 (1951)

[2] F. Cérou, A. Guyader, Stoch. Anal. Appl., *Adaptive multilevel splitting for rare event analysis*, **25(2)**:417-443 (2007)

[3] C.-E. Bréhier, M. Gazeau, L. Goudenège, T. Lelièvre, M. Rousset, Ann. App. Prob., *Unbiasedness of some generalized Adaptive Multilevel Splitting algorithms* (to be published)

[4] F. Cérou, A. Guyader, T. Lelièvre, D. Pommier, J. Chem. Phys., *A multiple replica approach to simulate reactive trajectories*, **134**:054108 (2011)

[5] 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, Ann. Nucl. Energ., *TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code*, **82**:151-160 (2015)

[6] J. J. Duderstadt, W. R. Martin, *Transport theory* (1979)

[7] G. I. Bell, S. Glasstone, *Nuclear reactor theory* (1970)

[8] K. Pearson, Nature, *The problem of the random walk*, **72**:294 (1905)

[9] A. Zoia, E. Dumonteil, A. Mazzolo, S. Mohamed, J. Phys. A: Math. Theor., *Branching exponential flights: travelled lengths and collision statistics*, **45(42)**:425002 (2012)

[10] A. Zoia, E. Dumonteil, A. Mazzolo, S. Mohamed, Phys. Rev. E, *Collision densities and mean residence times for d-dimensional exponential flights*, **83(4)**:041137 (2011)

[11] N. Metropolis, S. Ulam, J. Am. Stat. Assoc., *The Monte Carlo method*, **44(247)**:335-341 (1949)

[12] T. E. Booth, Nucl. Sci. Eng., *A Monte Carlo variance reduction approach for non-Boltzmann tallies*, **116**(2):113-124 (1994)

[13] E. Dumonteil, T. Lelièvre, H. Louvin, M. Rousset, *Adaptive Multilevel Splitting and Importance Sampling for Monte Carlo particle transport* (in preparation)