



# Adaptive multilevel splitting for Monte Carlo particle transport

Henri Louvin, Eric Dumonteil, Tony Lelièvre, Mathias Rousset, Cheikh M. Diop

## ► To cite this version:

Henri Louvin, Eric Dumonteil, Tony Lelièvre, Mathias Rousset, Cheikh M. Diop. Adaptive multilevel splitting for Monte Carlo particle transport. EPJ N - Nuclear Sciences & Technologies, 2017, 3, pp.29. 10.1051/epjn/2017022 . cea-01794118

**HAL Id: cea-01794118**

**<https://cea.hal.science/cea-01794118>**

Submitted on 2 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Adaptive multilevel splitting for Monte Carlo particle transport

Henri Louvin<sup>1,\*</sup>, Eric Dumonteil<sup>2</sup>, Tony Lelièvre<sup>3</sup>, Mathias Rousset<sup>3</sup>, and Cheikh M. Diop<sup>1</sup>

<sup>1</sup> CEA Saclay, DEN/DM2S/SERMA/LTSD, 91191 Gif-sur-Yvette, France

<sup>2</sup> IRSN, PSN-EXP/SNC, 92262 Fontenay-aux-Roses, France

<sup>3</sup> Université Paris-Est, CERMICS (ENPC), INRIA, 77455 Marne-la-Vallée, France

Received: 6 January 2017 / Received in final form: 18 May 2017 / Accepted: 30 August 2017

**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 (AMS) 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 algorithm, adapted for the first time to the field of particle transport. 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 like pulse-height tallies and other spectra. 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 delimited by so-called splitting levels, 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 does not use an a priori definition of importance regions. Instead, the splitting levels are determined on the fly, following a selection mechanism based on the classification of the simulated particle histories.

One of the most interesting features of AMS, which will be illustrated in this work through various numerical simulations, is that it yields very robust results even if the importance function only reflects a poor knowledge of the system. 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 the simulation of path-dependent quantities and applied 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 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 practical 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<sup>®</sup>. In

\* e-mail: [henri.louvin@cea.fr](mailto:henri.louvin@cea.fr)

the last section of this paper, we present some of the results obtained using AMS within TRIPOLI-4<sup>®</sup>. 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 of the AMS is a variant of the algorithm that perfectly fits the theoretical frame from [\[3\]](#), so that the estimator of the rare event occurrence 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  $D$  a subset of  $\mathbb{R}^d$ , we define the entrance time of the Markov chain  $X$  in  $D$ :

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

where  $\tau_f$  is an almost surely finite stopping time of the Markov chain  $X$  such that

$$X_{t \geq \tau_f} = X_{\tau_f}.$$

In the context of particle transport, this stopping time is the number of collisions after which a particle is absorbed. An absorbed particle is obviously not transported anymore, which results in its state being the same at any time after absorption.

Given an observable  $\phi_D : \mathcal{P} \rightarrow \mathbb{R}$  such that  $\phi_D(X) = 0$  on the event  $\tau_f < \tau_D$ , we would like to estimate the average  $\mathbb{E}(\phi_D(X))$  of  $\phi_D$ . Let us suppose now that the probability for  $X$  to enter  $D$  before  $\tau_f$  is very small, so that estimating  $\mathbb{E}(\phi_D(X))$  requires to sample the event  $\tau_D < \tau_f$ . In such a case of rare event simulation, the AMS algorithm proposes to reduce the variance on the estimation of  $\mathbb{E}(\phi_D(X))$  by increasing the number of Markov chains reaching the subset  $D$ . 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  $X$ . Each of the following steps consists in the resampling of the less interesting Markov chains amongst the replicas. Each resampled replica is a duplication of a more interesting one up to a certain time, defined using the information on the system gathered during the initial simulation.

### 2.2 AMS algorithm

#### 2.2.1 Importance function

In order to define which replicas are of interest to the simulation, the AMS algorithm has to be associated to a function quantifying the importance of a given Markov

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

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

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

$$\begin{aligned} \xi(x) &\geq I_{\max} & \text{if } x \in D \\ \xi(x) &< I_{\max} & \text{if } x \notin D. \end{aligned} \quad (1)$$

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

$$I(X) = \sup_{t \in [0; \tau_f]} \xi(X_t). \quad (2)$$

The  $\xi$  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  $D$ . It is therefore important to choose a good function  $\xi$  with regard to the rare event we are trying to simulate. Even if the AMS algorithm is proven to yield an unbiased result *regardless of*  $\xi$ , 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  $\xi$  in [Section 5](#).

#### 2.2.2 Initialization

The AMS algorithm consists of 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 each  $j$ , the initial state  $X_{0,0}^j$  is a point located outside of  $D$ . We then define the initial importance level  $Z_0$  as:

$$Z_0 = \inf_{j \in [1; n]} (\xi(X_{0,0}^j)),$$

so that every replica has an importance greater or equal to  $Z_0$ .

Let us now 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 common 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

- 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 Eq. (2)):

$$S_q^j = I(X_q^j). \quad (3)$$

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

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

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

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

- If  $Z_{q+1} \geq I_{\max}$ , stop iterating and go to the final step (see Sect. 2.2.6).
- For each  $j \in \{1, \dots, n\}$ , denote by  $X_q^{(j)}$  the Markov chain for which importance is  $S_q^{(j)}$ .
- For all  $j \in \{1, \dots, k\}$ , resample the replica  $X_q^{(j)}$  according to the resampling kernel described in Section 2.2.4, and denote it by  $X_{q+1}^j$ .
- For all  $j \in \{k+1, \dots, n\}$ , define  $X_{q+1}^j = X_q^{(j)}$ .
- Update the common weight:

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

- Increment  $q$  and go to step (i).

### 2.2.4 Resampling process

At each iteration of the AMS algorithm,  $k$  replicas are resampled. Let us denote the current iteration number by  $q$  and the associated resampling level by  $Z_{q+1}$ .

For each of the  $k$  replicas  $X_q^{(j)}, j \in \{1, \dots, k\}$  to be resampled, one of the remaining replica  $X_q^{(i)}, i > k$  is *randomly* selected for duplication. We know for sure that the Markov chain  $(X_{q,t}^{(i)})_{t \in [0; \tau_{q,t}^{(i)}]}$  contains at least a state whose importance is greater than  $Z_{q+1}$  (otherwise it would have been resampled). We can therefore define

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

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

The resampled Markov chain  $(Y_t)_{t \geq 0}$  is defined as a copy of  $X_{q,t}^{(i)}$  for all  $t \in [0, \tau_q^{(i)}]$ , and is then completed independently using the original Markov kernel, up to the stopping time  $\tau_f$ . This resampled Markov chain replaces the original one for the next iteration of AMS.

### 2.2.5 Stopping criterion

The iterating process terminates at iteration  $q$  either if  $Z_{q+1}$  is greater than  $I_{\max}$  or if all paths have the same importance.

Given that the probability of reaching the target is non-zero, there is always a possibility for the simulated Markov chains to reach the subset  $D$  (even directly from their source point). Since any replica that reaches  $D$  has an importance greater than any other replica in the simulation, they are never resampled and may be used as base for the resampling of all the others.

Consequently, the values of the resampling levels and splitting levels will keep rising until the stopping criterion is met. It maybe however possible that no replica ever reaches  $D$  during the simulation. Since the simulated space is finite, it is also impossible to have resampled replicas whose importance values keep growing at each iteration. In pathological situations, the resampling points will ultimately be the highest rated in the duplicated Markov chain, and the algorithm is doomed to stop when all replicas have the same importance. In that case, the algorithm is simply interrupted and goes straight to the final step, eventually yielding a null contribution.

### 2.2.6 Final step

Once the algorithm stops iterating, we denote by  $Q$  the number of completed iterations. Given the bounded observable  $\phi_D$  introduced in Section 2.1, we define

$$\hat{\phi}_D = \sum_{j=1}^n W_Q^j \phi_D(X_Q^j), \quad (4)$$

as an **unbiased** estimator of  $\mathbb{E}(\phi_D(X))$  [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.  $\mathbb{P}(S_q^j > Z_{q+1} | S_q^j > Z_q)$ , since the importance of every paths at this iteration is greater than  $Z_q$ ), 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  $\mathbb{P}(I(X) > Z_{q+1})$ .

In other words, the AMS algorithm provides us at each iteration  $q$  with a set of paths  $X_q^j, j \in \{1, \dots, n\}$ , carrying a common 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 Section 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 whose 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 replicas to be resampled is  $K_q > k$ , then the

weight update at step (viii) of the algorithm has to be changed to:

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

### 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 gas of non-interacting particles (e.g. neutrons and 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 an homogeneous medium are transported along straight lines between collisions points and are randomly reoriented at each interaction (possibly with change of energy) [8]. As the flight lengths are exponentially distributed, 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 may be 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 flight 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 trajectories 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 trajectory 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 that 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  $(\mathbf{X}, \boldsymbol{\Omega}, E)$ , where  $\mathbf{X}$  denotes the particle position in the 3-dimensional space,  $\boldsymbol{\Omega}$  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 Section 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 maps 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 the *importance function* or the *importance map*, and denote it by

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

In order to satisfy the requirement (1), it is sufficient to define an importance  $I_{max}$  larger than any other value of the map, and to assign this value to any point within the target volume. As the importance value is only used to sort tracks with respect to one another, it is strictly equivalent to choose for  $I_{max}$  the maximum value of the importance function or any value greater than that. This is why TRIPOLI-4<sup>®</sup> sets  $I_{max}$  to numerical infinity (i.e. the maximum representable floating-point number in the code) regardless of the importance map.

#### 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 minimal number of particles that are resampled at each iteration of the algorithm. Notice that  $n$ ,  $k$  and the importance function  $\xi$  are numerical parameters that can be chosen by the user, and that the estimator of  $E(\phi(X))$  is unbiased whatever these choices.

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.

In practice, each point of the saved track is a set of coordinates  $(\mathbf{X}, \mathbf{\Omega}, E)$  representing the particle at an interaction point, where  $\mathbf{X}$  is the position of the interaction, and  $(\mathbf{\Omega}, E)$  the direction and energy of the particle after the collision.

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:

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

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

If  $Z_{q+1}$  is greater than  $I_{max}$ , the iterating process stops. Otherwise, all the tracks that have an importance less or equal to the level  $Z_{q+1}$  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  $Z_{q+1}$ , according to the resampling kernel presented in [Section 2.2.4](#). The global weight is then updated:

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

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  $Z_q$  is greater or equal to  $I_{max}$ , or when all tracks importances are equal, the algorithm stops iterating. We then define  $Q$  as the total number of completed iterations. 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. 2.2.6](#)):

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

### 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  $w_Q$ , except for pathological cases. However, the cases of extinction are rare and caused by the use of a very inadequate importance function, such as an importance function favouring particles that are obviously going the wrong way, or an overly coarse discretized map yielding to several particles having the same importance.

#### 4.4.2 Robustness with regard to the importance function

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

$$I_{max} > I_{source}, \quad (6)$$

where  $I_{source}$  is the minimal possible importance value for a particle at its source point. The purpose of this condition is to ensure that the importance of particle track entering the target volume is greater than any particle coming from the source point and not reaching the target. That way, the tracks having an importance  $I_{source}$  can be resampled while the tracks that have reached the target remain in the simulation, yielding the same results as an analog simulation.

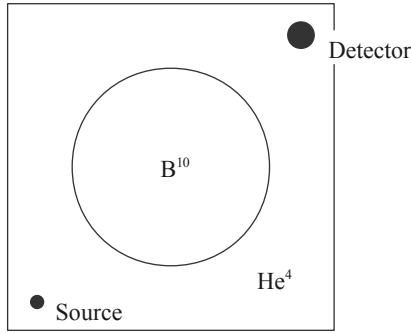
In our implementation of the AMS algorithm, this condition is *always verified*, as we set the importance of the target to numerical infinity regardless of the importance map (see [Sect. 4.2](#)). This allows us to use any function as importance. However, if the map does not properly describe the system, the result, though unbiased, may be plagued by a large variance.

Although the AMS algorithm is adaptive, the obtained estimator of the quantity of interest is always unbiased. Therefore, one can ensure the quality of the result by running multiple AMS simulations with different importance functions until the confidence intervals overlap.

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

As we already pointed out, any choice of  $k$  and  $\xi$  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 for deep penetration problems, or a path from the source to the detector for streaming configuration, and to set any value for  $k$  less than  $n$  to



**Fig. 1.** Bypass simulation configuration.

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 as well as the subtleties of the geometry.

#### 4.4.4 Parallelization

A Monte Carlo simulation performed with AMS yields a single value as estimation of the quantity of interest. In order to obtain a precise result, multiple independent simulations are required. The arithmetic mean of the estimations over all simulations is computed together with the associated variance, in the same way as in usual Monte Carlo, which requires multiple independent simulations.

As the simulations remain totally independent until their output is gathered to compute the average results, a straightforward way of parallelizing AMS Monte Carlo is to run independent simulations (initialization, iterations and scoring) simultaneously, each on a distinct processor. That way the algorithm does not need any modification to achieve parallelism.

## 5 Validation and results

The AMS algorithm was implemented in the development version of the Monte Carlo particle transport code TRIPOLI-4<sup>®</sup> [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<sup>®</sup> 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<sup>®</sup> 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 a 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 on two factors. On one hand, the relative error  $\sigma$  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

$$\sigma \frac{\bar{I}}{\sqrt{N_B}} \text{ and } T \tilde{N}_B. \quad (7)$$

We use the Figure Of Merit (or FOM) to evaluate the efficiency of a Monte Carlo simulation. The FOM is defined as

$$\text{FOM} = \frac{1}{\sigma^2 T}, \quad (8)$$

and should remain relatively constant as  $N_B$  increases 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<sup>®</sup> 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<sup>®</sup>, which is used for other variance reduction techniques in this code and to automatically pre-compute an importance map (see [5]).

### 5.3 Bypass simulation

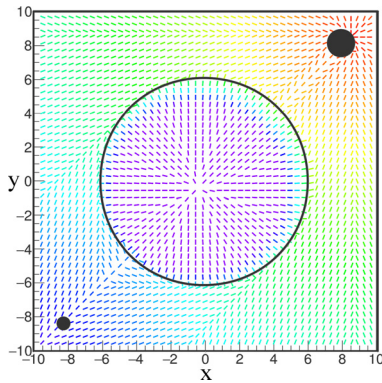
The first system consists of an extruded box filled with helium. The dimensions of the box are 10 cm × 10 cm × +∞. A neutron flux is produced from a 2 MeV isotropic neutron source at one corner of the box and detected inside an infinite cylinder of 1 cm in diameter placed at the opposite corner. In between 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 Figure 1.

The results obtained using TRIPOLI-4<sup>®</sup> without variance reduction and TRIPOLI-4<sup>®</sup> using the AMS algorithm are presented in Table 1. Several AMS simulations with 10<sup>4</sup> initial particles were performed, with different values for the parameter  $k$ , from 1 duplicated particle (0.01%) per iteration up to 5 × 10<sup>3</sup> (50%). Two distinct importance maps were used in this study:

- the “Spatial” importance map, defined using the reciprocal of the distance to the target;
- the importance map computed by INIPOND.

**Table 1.** Comparison of TRIPOLI-4<sup>®</sup> with and without AMS for flux estimation in the bypass geometry (48h calculations).

| Simulation       | $k$ (%) | Number of batches | Mean                     | Relative error | FOM                      | Gain        |
|------------------|---------|-------------------|--------------------------|----------------|--------------------------|-------------|
| Analog           | N/A     | 58 634            | $3.8554 \times 10^{-10}$ | 13.50          | $6.9954 \times 10^{-08}$ | <b>01</b>   |
| AMS<br>(Spatial) | 0.01    | 68                | $4.3179 \times 10^{-10}$ | 4.45           | $6.3347 \times 10^{-07}$ | <b>09</b>   |
|                  | 0.05    | 793               | $4.1640 \times 10^{-10}$ | 2.37           | $2.2742 \times 10^{-06}$ | <b>32</b>   |
|                  | 0.1     | 1999              | $4.2156 \times 10^{-10}$ | 1.74           | $4.1893 \times 10^{-06}$ | <b>59</b>   |
|                  | 0.5     | 5378              | $4.1874 \times 10^{-10}$ | 1.28           | $7.7867 \times 10^{-06}$ | <b>111</b>  |
|                  | 1       | 6123              | $4.2000 \times 10^{-10}$ | 1.25           | $8.1607 \times 10^{-06}$ | <b>116</b>  |
|                  | 5       | 7910              | $4.1821 \times 10^{-10}$ | 0.99           | $1.3139 \times 10^{-05}$ | <b>187</b>  |
|                  | 10      | 7682              | $4.1321 \times 10^{-10}$ | 1.04           | $1.1701 \times 10^{-05}$ | <b>167</b>  |
|                  | 50      | 8661              | $4.1620 \times 10^{-10}$ | 1.09           | $1.0657 \times 10^{-05}$ | <b>152</b>  |
| AMS<br>(INIPOND) | 0.01    | 105               | $4.2411 \times 10^{-10}$ | 1.80           | $3.9036 \times 10^{-06}$ | <b>55</b>   |
|                  | 0.05    | 818               | $4.1417 \times 10^{-10}$ | 0.83           | $1.8636 \times 10^{-05}$ | <b>266</b>  |
|                  | 0.1     | 2007              | $4.1624 \times 10^{-10}$ | 0.62           | $3.3366 \times 10^{-05}$ | <b>476</b>  |
|                  | 0.5     | 4859              | $4.1509 \times 10^{-10}$ | 0.38           | $8.8830 \times 10^{-05}$ | <b>1269</b> |
|                  | 1       | 4572              | $4.1685 \times 10^{-10}$ | 0.36           | $9.6031 \times 10^{-05}$ | <b>1372</b> |
|                  | 5       | 6029              | $4.1727 \times 10^{-10}$ | 0.37           | $9.4906 \times 10^{-05}$ | <b>1356</b> |
|                  | 10      | 5970              | $4.1671 \times 10^{-10}$ | 0.36           | $9.6938 \times 10^{-05}$ | <b>1385</b> |
|                  | 50      | 7690              | $4.1747 \times 10^{-10}$ | 0.37           | $9.5076 \times 10^{-05}$ | <b>1359</b> |

**Fig. 2.** INIPOND importance map for the bypass problem.

The values and gradient of the INIPOND map are represented in Figure 2, on which we can see that the automated module was able to determine the preferential pathways around the Bore cylinder. Every simulation was performed at the same time on the same machine, each of them on a single processor.

We observe that the AMS yields an unbiased result for any combination of  $k$  and importance function. What is more, the algorithm always improves the FOM. Even though the AMS using the spatial importance as classifying function is interesting in terms of FOM, compared to the analog result, the use of the INIPOND map taking into account the energy and direction of the neutrons allows for a great improvement in the algorithm efficiency.

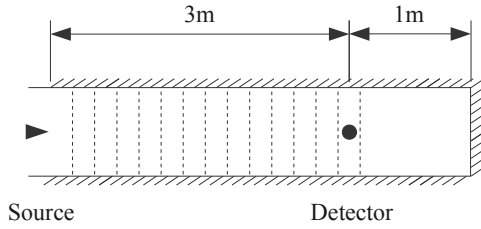
Concerning the impact of the  $k$  parameter, both importance functions show the same comportment. First of all, the number of independent simulations completed increases with  $k$ . This is understandable, as the number of iterations required to reach the detector decreases when  $k$  increases, so that the computation time per simulation is smaller.

The smallest relative error for a given number of independent simulations is obtained with  $k = 1$  (0.01%) for both importance maps. 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 balance between estimation precision and computational speed seems to be reached for a value of  $k$  between 1% and 50% of the initial particles. In this range, the variations in FOM gain are stochastic. Below this range, the overall efficiency of the algorithm drops quite abruptly.

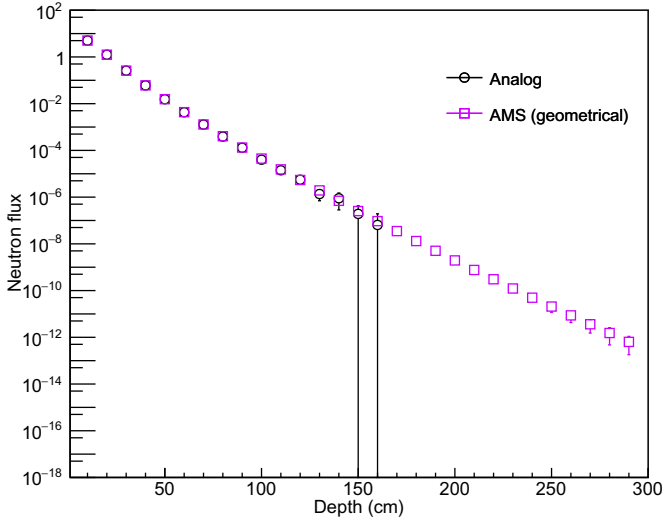
The study of the bypass problem puts into light the robustness of the AMS efficiency with regard to the parameter  $k$  and the importance, as the result is always unbiased. However, it is to be noted that the geometry of the bypass configuration is quite simple.

We may find cases in which the number of splittings per iteration has a greater impact on the algorithm efficiency, such as streaming configurations containing preferential pathways with very small entrances, or any other problem where spatial details have a strong impact. If the number of splittings per iteration is too high, such details may be overlooked by the AMS algorithm, because the distance between the splitting levels would be more important.





**Fig. 3.** Water box simulation system.



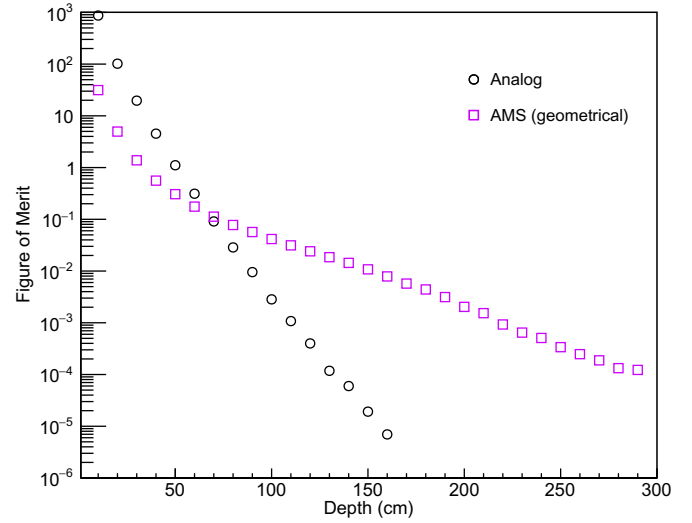
**Fig. 4.** Flux attenuation in the Water box.

#### 5.4 Neutron flux attenuation in water

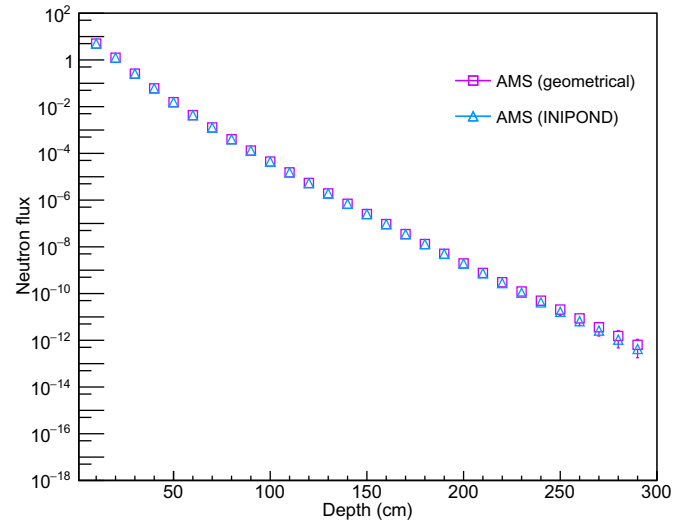
The source of the second problem is a directional neutron point source, emitting neutrons according to a Watt spectrum and placed at the entry of a straight box filled with water. The detector is placed at 3 m from the source, and the neutron flux is estimated in 10-cm-thick slices between the source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

The neutron flux attenuation was estimated using both analog and AMS TRIPOLI-4<sup>®</sup> 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  $k = 10^2$  splittings per iteration. Data were collected for both simulations during 65 h. Both simulation were performed at the same time, on the same machine, each one on a single processor. Figure 4 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 Figure 5.

It can be seen in Figure 4 that the analog simulation is unable to estimate a flux farther than 160 cm 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 Figure 5. The neutron flux in this part of the box is high enough for TRIPOLI-4<sup>®</sup> not to need variance reduction, which places the AMS at a disadvantage due to its impact on runtime.



**Fig. 5.** FOM comparison in the Water box.



**Fig. 6.** Flux attenuation in the Water box.

The AMS mechanism starts to be interesting at a depth of 70 cm. The gain in FOM increases up to  $10^3$  in favor of AMS at 160 cm, 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 Figure 6, and the associated FOM in Figure 7.

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 160 cm. 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.

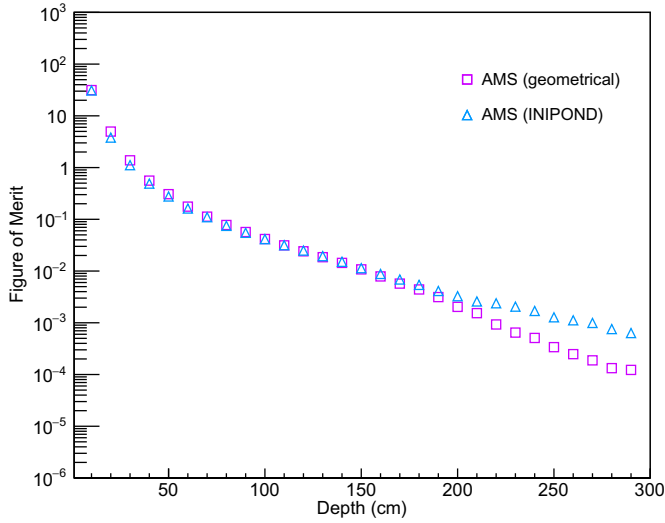


Fig. 7. FOM comparison in the Water box.

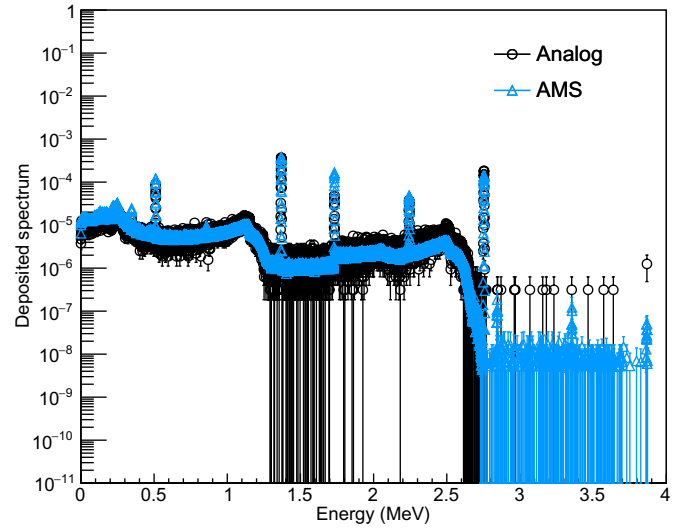


Fig. 9. Deposited spectrum in the HPGe detector.

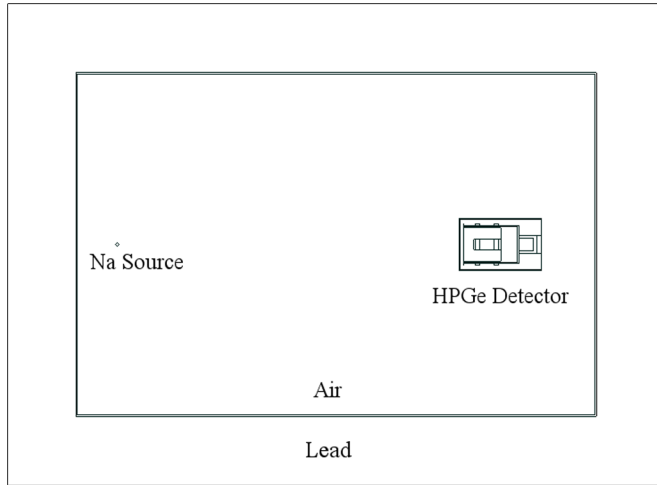


Fig. 8. Gamma spectrometry setting.

## 5.5 Gamma spectrometry

During an AMS iteration, the particle transport is analog. This feature lets us hope to use AMS to reduce the variance on non-Boltzmann scores, i.e. scores that cannot be described by the Boltzmann transport equation. For example, the scores for which the whole particle history has to be recorded are non-Boltzmann. The most common of those scores is the so-called *pulse-height tally*, which is designated as “deposited spectrum” in TRIPOLI-4<sup>®</sup>.

The deposited spectrum score records the total energy deposited in the detector by a source particle from its emission at the source to the termination of all progeny. This implies that the entire history of the primary particle has to be memorized, and that all secondary tracks are correlated. If a photon passes through the detector while depositing 1 MeV of its energy in it, and later on re-enters the detector region and deposits once again 1 MeV before being killed, the whole process registers as a single event of 2 MeV.

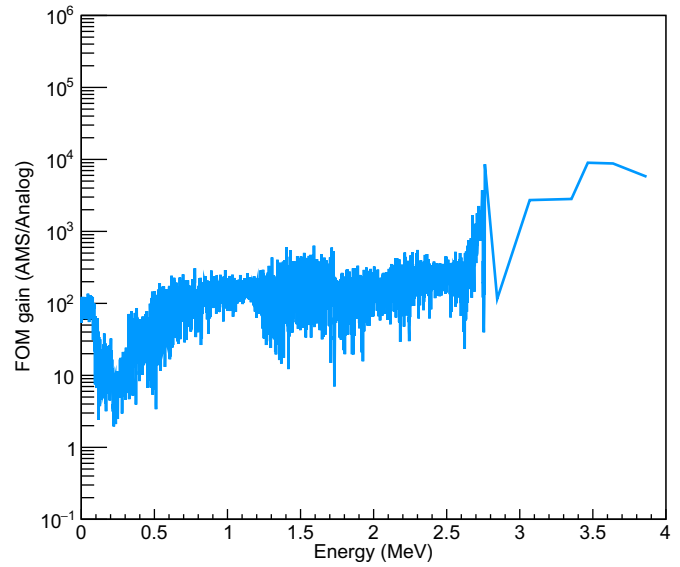


Fig. 10. Ratio of the AMS FOM over the analog FOM as a function of energy for the deposited spectrum.

Consequently, the use of weight-based variance reduction schemes such as MCNP’s Weight Windows or Exponential Biasing in TRIPOLI-4<sup>®</sup> is troublesome, as it requires to construct particle history trees retaining all events between the emission of the particle and the termination of all progeny with the associated weight modifications, and then to adequately take into account the various weight to compute the score [12].

The AMS algorithm offers the possibility of reducing the variance on deposited energy spectra without altering the algorithm. This is made possible by two properties of AMS. First of all, the particle transport between splitting events is analog. Therefore, the computation of a particle’s contribution, if restricted to a single iteration, is performed the same way as for analog Monte Carlo.

The second property is that a particle that has reached the detector is never resampled. Therefore, its contribution remains unchanged in all the iterations following the scoring. At the end of the simulation, each contribution is weighted by the AMS global weight, which represents the occurrence probability of the final set of tracks.

The problem considered to test AMS efficiency for deposited energy spectrum is a gamma spectrometry problem. The setting is shown in Figure 8. A sodium source is placed in an air environment encased in lead. At 50 cm of the source, a realistic high-purity germanium (HPGe) detector is simulated. We aim to estimate the energy spectrum deposited in the detector by the photons coming from the sodium source.

These results are from photon-only simulations, as the coupled photons/electrons/positrons AMS is not yet fully operational. We present in Figure 9 the pulse-height tally obtained by simulating the response of the HPGe detector for a given computation time. Both simulations ran for 6 h, each on a single processor of the same machine. The importance used for the AMS simulation is the importance map calculated by INIPOND.

We can see that both spectra are in perfect accordance, although the spectrum coming from the AMS simulation seems much more converged. In order to quantify this effect, we computed the FOM gain, defined as the ratio of  $FOM_{AMS}$  over  $FOM_{Analog}$ . It is represented in Figure 10.

On the far right side of the deposited spectrum shown in Figure 9, we observe that the AMS simulation is even able to simulate events leading to high-energy depositions that were simply not visible on the analog spectrum. It is really interesting to see how easy it is to use the AMS algorithm in configurations for which other variance reduction schemes need to be adapted. The parametrization of AMS for photons and for deposited spectra remains unchanged and as simple as for other cases.

## 6 Conclusion

The results presented in this paper prove that the Adaptive Multilevel Splitting algorithm is a viable variance reduction scheme for Monte Carlo particle transport codes. Its implementation in TRIPOLI-4<sup>®</sup> and its connection to the 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 robust variance reduction scheme. It seems to be a good alternative to existing variance reduction techniques in the most severe configurations. Furthermore, it can also be used for deposited scores without need to alter the algorithm, as seen in Section 5.5, and allows TRIPOLI-4<sup>®</sup> to use variance reduction for those scores.

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 point of interest is the simulation of larger-scale problems, such as full-core simulations for ex-core neutron flux estimations. The challenge for AMS at such scales is the same as for other variance reduction: the definition of a proper importance map or function, sufficiently precise to reduce the variance, but at the same time not too detailed to prevent memory and storage issues. However, the robustness of AMS with regard to the importance function lets us hope for efficient variance reduction even with roughly estimated importances.

TRIPOLI-4<sup>®</sup> 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, Estimation of particle transmission by random sampling, *Nat. Bur. Stand. Appl. Math. Ser.* **12**, 27 (1951)
2. F. Cérou, A. Guyader, Adaptive multilevel splitting for rare event analysis, *Stoch. Anal. Appl.* **25**, 417 (2007)
3. C.-E. Bréhier, M. Gazeau, L. Goudenège, T. Lelièvre, M. Rousset, Unbiasedness of some generalized Adaptive Multilevel Splitting algorithms, *Ann. Appl. Prob.* **26**, 3559 (2016)
4. F. Cérou, A. Guyader, T. Lelièvre, D. Pommier, A multiple replica approach to simulate reactive trajectories, *J. Chem. Phys.* **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, TRIPOLI-4<sup>®</sup>, CEA EDF and AREVA reference Monte Carlo code, *Ann. Nucl. Energy* **82**, 151 (2015)
6. J.J. Duderstadt, W.R. Martin, *Transport Theory* (Wiley, New York, 1979)
7. G.I. Bell, S. Glasstone, *Nuclear Reactor Theory* (Van Nostrand Reinhold Company, New York, 1970)
8. K. Pearson, The problem of the random walk, *Nature* **72**, 294 (1905)
9. A. Zoia, E. Dumonteil, A. Mazzolo, S. Mohamed, Branching exponential flights: travelled lengths and collision statistics, *J. Phys. A: Math. Theor.* **45**, 425002 (2012)
10. A. Zoia, E. Dumonteil, A. Mazzolo, S. Mohamed, Collision densities and mean residence times for d-dimensional exponential flights, *Phys. Rev. E* **83**, 041137 (2011)
11. N. Metropolis, S. Ulam, The Monte Carlo method, *J. Am. Stat. Assoc.* **44**, 335 (1949)
12. T.E. Booth, A Monte Carlo variance reduction approach for non-Boltzmann tallies, *Nucl. Sci. Eng.* **116**, 113 (1994)