Adaptive Multilevel Splitting: Historical Perspective and Recent Results

Frédéric Cérou, Arnaud Guyader, Mathias Rousset

Adaptive Multilevel Splitting: Historical Perspective and Recent Results

Frédéric Cérou[1]
INRIA–Rennes & Univ Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France
frederic.cerou@inria.fr

Arnaud Guyader
LPSM, Sorbonne Université, 75005 Paris, France, & CERMICS, École des Ponts ParisTech, 77455 Marne la Vallée, France
arnaud.guyader@upmc.fr

Mathias Rousset
INRIA–Rennes & Univ Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France
mathias.rousset@inria.fr

Abstract

About ten years ago, the Adaptive Multilevel Splitting algorithm (AMS) was proposed to analyse rare events in a dynamical setting. This review paper first presents a short historical perpective of the importance splitting approach to simulate and estimate rare events, with a detailed description of several variants. We then give an account of recent theoretical results on these algorithms, including a central limit theorem for Adaptive Multilevel Splitting. Considering the asymptotic variance in the latter, the choice of the importance function, called the reaction coordinate in molecular dynamics, is also discussed. Finally, we briefly mention some worthwhile applications of AMS in various domains.

Index Terms — Sequential Monte-Carlo, Interacting particle systems, Rare events

2010 Mathematics Subject Classification: 82C22, 82C80, 65C05, 60J25, 60K35, 60K37

[1]Corresponding author.

1 Introduction

This paper proposes a review on the Adaptive Multilevel Splitting (AMS) algorithm to simulate rare events associated with a stochastic dynamical system. First, let us explain what we call a rare event: it is an event with non-zero, but very small probability. We assume that the probability is so small that typically we do not have any realization of the event of interest within a reasonable simulation time through a naive Monte-Carlo approach. To give an idea, this probability will typically be smaller than 10^{-10}, so that the number of simulations needed to observe only a handful of realizations is not tractable.

We can give two generic examples where it is required to precisely estimate small probabilities. First, when the rare event is some kind of catastrophe, and it is needed to know exactly how small it is, e.g. for air traffic management, or insurance. Second is when we need to estimate the mean time of return into some set for a stochastic dynamical system. This can be a transition that is not rare at the macroscopic scale, but if the dynamical system can only be simulated with a very small timestep, then that event becomes rare in the simulation timescale. We will have these two situations in mind throughout the present paper.

In the case of such a rare event, it is easy to see mathematically why a naive Monte-Carlo approach, also called crude Monte-Carlo or direct numerical simulation, is not suited. Assume we want to estimate a probability $p = \mathbb{P}(X \in R)$, with $p > 0$ but very low, for some random variable or process X, and a measurable set R. The naive approach is to draw an N i.i.d. sample $(X_1, \dots, X_N)$, with the same distribution as X, and compute the empirical probability

$$\hat{p} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_R(X_i).$$

Since the variables $\mathbf{1}_R(X_i)$ are i.i.d. Bernoulli with probability of success p, the variance of $\hat{p}$ is simply $\mathbb{V}(\hat{p}) = p(1-p)/N$. If we consider the normalized variance $\mathbb{V}(\hat{p}/p) = (1-p)/(Np) \approx 1/(Np)$, we see that the estimator is getting quickly worse when p goes to 0. To keep it within reasonable bounds, we would need to take N of order $1/p$, which is intractable for a very low probability p.

Therefore, to construct a good estimate, one has to use some variance reduction technique. For this problem, there are broadly two families of solutions: one is Importance Sampling, and the other one is Importance Splitting. The

latter is the main subject of this paper, but let us write a few words on the former.

The idea of Importance Sampling is to sample from an auxiliary distribution in order to make the rare event less rare. Let us denote by $(Y_1, \ldots, Y_N)$ the new i.i.d. sample. One usual requirement is that the distribution of X is absolutely continuous w.r.t. the law of Y, and we denote the corresponding Radon-Nikodym derivative by $\frac{dP_X}{dP_Y}$. We then estimate the probability p by

$$\tilde{p} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_R(Y_i) \frac{dP_X}{dP_Y}(Y_i).$$

This estimator is clearly unbiased. Nonetheless, it is not always obvious to choose a good importance distribution to sample from, that is one which will give an estimator with a small normalized variance. In fact, if the sampling distribution is badly chosen, things can be very bad, as one is not even guaranteed to have a finite variance (see for example [26]).

Importance Sampling is a very common approach to reduce variance, and there is a huge amount of literature on the subject, that we will not discuss here. Let us just mention the monograph [9] in the context of rare events, and [36] for illustrations in molecular dynamics.

The idea of Importance Splitting, that is the family AMS belongs to, is to simulate according to the original distribution in a sequential way, to discard the trajectories (or samples) going far away from R, and to split/branch/clone those that get closer. This will be made more precise in the sequel. Ä Before proceeding, let us just mention that, in what follows, we will focus our attention on the so-called "dynamic case". By this, we mean that the rare event of interest writes $p = \mathbb{P}(X_\tau \in R)$, with X a strong Markov process and τ a stopping time (see Section 2.1). The so-called "static case" corresponds to the situation where, typically, X is a random vector in $\mathbb{R}^d$, S is a function from $\mathbb{R}^d$ to $\mathbb{R}$, and one wants to estimate the probability $p = \mathbb{P}(S(X) > q)$ where q is known and such that p is strictly positive but very low. There are of course some strong connections between multilevel splitting methods for the dynamic case and the static case, but the results obtained in both situations are not exactly the same ones. However, even if we have chosen to focus on the dynamic case, we will also mention here and there some results available in the static case.

Finally, note that the key word "multilevel" in Adaptive Multilevel Splitting is completely unrelated to the "Multilevel Monte Carlo" developed by Giles (see, e.g., [24] or [25]) although there are some attempts in the literature to use both ideas in conjunction (see for example [44]).

2 A little history

2.1 The origins

Interestingly enough, one of the first presentations of both Importance Sampling and Importance Splitting is generally thought to be [30]. Their original problem came from particle physics, where the goal is to simulate and study the particle transmission through an obstacle. Of special interest is the probability that the particle goes through the obstacle without being absorbed. If, for instance, a nuclear device is to be considered safe, that probability should be extremely low.

Let us state the problem in a more abstract form. Let X be a time homogenous strong Markov process in $\mathbb{R}^d$ starting at $t = 0$ with a known distribution η_0. Let $\mathcal{F}_t^X = \sigma(X_s,\, 0 \leqslant s \leqslant t)$ denote its natural filtration that we will assume for simplicity to be right-continuous. These assumptions ensure that the hitting time τ of any (measurable) set is a stopping time, and that the law of X conditional on the past of τ is again the law of X with initial condition X_τ.

Note that, in many cases, one may assume that we have preliminarily added the time in the state space (i.e., one coordinate of the state X is the time variable). Then there is no loss of generality in considering time inhomogenous processes and observables that may depend on time.

We suppose that τ is a.s. finite, which includes the case of a deterministic time. The problem is then to construct an estimator of

$$p = \mathbb{P}(X_\tau \in R),$$

and to give a sample of trajectories such that its empirical measure is an approximation of the distribution of $(X_s, 0 \leqslant s \leqslant \tau)$ given that $X_\tau \in R$.

To be more specific, let us give two different instances of the latter. For the original problem in [30], we have a random killing time σ_a (absorption of the particle), and R is the outside of a confinement tank. If we denote σ_R the hitting time of the outside, then $\tau = \sigma_a \wedge \sigma_R$.

Another typical setting arises in molecular dynamics, detailed below in Section 4.4. Here X is a diffusion process, with a drift that derives from a potential, and a constant (full rank) noise intensity. We want to simulate *reactive trajectories*, which are trajectories that reach a region R (typically another well in an energy landscape), before visiting a recurrent set (typically a neighborhood of the bottom of the current well of the potential). If we denote σ_r the latter stopping time, we have in that case $\tau = \sigma_R \wedge \sigma_r$.

The basic idea is to use a real-valued function Φ, also called an importance function or a reaction coordinate in molecular dynamics, to measure, even roughly, how "close" the process may be from the rare event. Ideally, imagine that $\Phi(x) = \Phi^\star(x) = q^\star(x)/p$ where

$$q^\star(x) = \mathbb{P}(X_\tau \in R | X_0 = x)$$

is the so-called committor function, which we will extensively discuss later (to write the latter in all generality, we need to assume that the time t is a component of the state vector x, but this is not necessary for the two examples given just above). Then each time a sample trajectory crosses a surface $\{\Phi^\star(x) = 2^k\}$, for $k \in \{1, \dots, \lfloor \frac{\log p}{\log(1/2)} \rfloor\}$, we *split*, or *clone* that trajectory in 2, that is afterwards, we simulate 2 independent trajectories, issued from the same passage point into the level set $\{\Phi^\star(x) = 2^k\}$. Of course we also divide by 2 the mass of each cloned trajectory. If we do that until step $k = \lfloor \frac{\log p}{\log(1/2)} \rfloor$, the remaining paths have a probability larger than $1/2$ of hitting R. The latter probability can therefore be estimated by crude Monte-Carlo. Moreover, the number N of trajectories is random, but its expectation is constant and equal to N_0, the size of the initial sample.

There is a huge difficulty when using this simple method on a practical example, because the function $\Phi^\star(x) = \mathbb{P}(X_\tau \in R | X_0 = x)/p$ is unknown, and its computation is intractable, otherwise the rare event problem would be solved. In practice a function Φ which is imprecise is used, often resulting from heuristics proposed by practitioners. By "imprecise", we mean that best asymptotic variance for the AMS estimator of p is reached when taking $\Phi = \Phi^\star$, which unfortunately is impossible in most situations of interest. This crucial point will be discussed in Section 3.5.

Note also that the number of clones 2 was suggested by the authors in [30], but of course any other value could be chosen: if it is not an integer, we can randomise the number of clones in order to keep the mean value constant. The Multilevel Splitting (MS) algorithm with an integer number of clones (to keep it simple) is detailed in Algorithm 1, and illustrated in Figure 1. We just assume that we have a continuous function Φ and a real number $L_{\max}$ such that, defining the stopping time

$$S_{L_{\max}} = \inf \{ s \geqslant 0,\, \Phi(X_s) > L_{\max} \},$$

we have the condition

$$X_\tau \in R \Rightarrow \tau \geqslant S_{L_{\max}},$$

which precisely ensures that the trajectories contributing to the target event $\{X_\tau \in R\}$ have all reached the open set $\{\Phi(x) > L_{\max}\}$. The choice of an

open set $\{\Phi(x) > L_{\max}\}$ is merely conventional, and the case $\{\Phi(x) \geqslant L_{\max}\}$ could be treated similarly. In the same way, the choice $\{\Phi(X_s^i) > L_j\}$ instead of, say $\{\Phi(X_s^i) \geqslant L_j\}$, in the definition of the thresholds is conventional. The only important condition is that the successive thresholds constitute a decreasing sequence of sets for inclusion.

Algorithm 1 Multilevel Splitting (MS)

Require: Initial distribution η_0, Importance Function Φ and levels $-\infty = L_0 < L_1 < \cdots < L_J = L_{\max}$ for a given number of levels J, cloning rates $r_1, \ldots, r_J$, initial sample size N_0

Initialization: $X_0^1, \ldots, X_0^{N_0}$ i.i.d. from η_0

for $j = 1$ to J **do**

  for $i = 1$ to N_{j-1} **do**

    Run trajectory i until next level $\{\Phi(X_s^i) > L_j\}$ or final time τ_i (the first reached)

  end for

  Discard trajectories that did not reach L_j

  Clone r_j times those which did

  Denote N_j the total number of resulting trajectories

  Reorder the trajectories with number from 1 to N_j

end for

for $i = 1$ to N_J **do**

  Run trajectory i until final time τ_i

end for

Estimate the probability of the rare event by

$$\hat{p}_{\text{ms}} = \frac{1}{N_0} \frac{1}{\prod_{j=1}^{J} r_j} \sum_{i=1}^{N_J} \mathbf{1}_R(X_{\tau_i}^i)$$

A crucial choice in Algorithm 1 is the branching rates r_j. Let us denote by S_{L_j} the hitting time of the threshold $\{\Phi > L_j\}$:

$$S_{L_j} = \inf\{s,\ \Phi(X_s) > L_j\},$$

and by

$$p_{L_j} = \mathbb{P}(S_{L_j} \leqslant \tau)$$

the associated probability. We will also denote

$$\theta_j = p_{L_j}/p_{L_{j-1}} = \mathbb{P}(S_{L_j} \leqslant \tau | S_{L_{j-1}} \leqslant \tau).$$

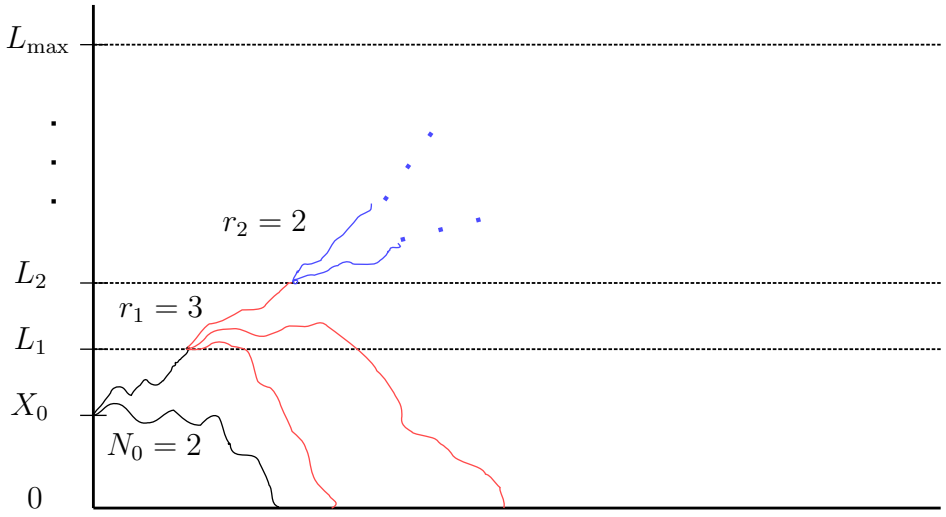

Figure 1: Algorithm 1 (MS), where X is to reach $L_{\max}$ before 0.

To have a useful algorithm, we need that the products $r_j\theta_j$ are all close to 1. If they are much smaller, there is a high probability that no trajectory hits R (like in naive Monte-Carlo); if they are much larger, the algorithm simulates a very large number of correlated trajectories. An analysis of the cost/variance trade-off of this approach can be found in [32, 31] for discrete dynamical models. In a nutshell, to get the maximum efficiency, the probabilities from one level to the next should all be equal, and the $r_j\theta_j$ should be equal to 1.

Also note that a version of the algorithm suitable for computing $\mu_\infty(R)$, where μ_∞ is the invariant probability measure of the process X, is proposed in [46, 45] and subsequent papers by the authors of [46]. Added to the splitting mechanism, it also provides a pruning rule for the trajectories that go downwards in terms of the importance function Φ, keeping only one trajectory all along the time axis. The use of pruning mechanisms to speed up the algorithm even when not considering the invariant probability measure is discussed in [34], which also provides a discussion on the early variants of importance splitting.

As we will see below, we can modify the algorithm so that we do not need to choose the branching rates a priori, and get a tractable algorithm without any real drawback.

2.2 Further developments

The fact that the number of active trajectories in Algorithm 1 is random makes it difficult to use. Moreover, if the branching rates are not well chosen,

it becomes also inefficient. This is why it was proposed in [23, 18, 11] a variant that keeps the number of active trajectories (or replicas) constant, which is detailed here as Algorithm 2 and Figure 2. In this algorithm, one may have at step j the equality $N_j = N$, in which case no splitting occurs (see also the discussion in Section 3.4). Moreover, in the latter again, the choice $\{\Phi(X_s^i) > L_j\}$ instead of, say $\{\Phi(X_s^i) \geqslant L_j\}$, in the definition of the thresholds is just conventional.

Algorithm 2 Sequential Monte-Carlo (SMC)

Require: Initial distribution η_0, Importance Function Φ and levels $-\infty = L_0 < L_1 < \cdots < L_J = L_{\max}$ for a given number of levels J, sample size N

Initialization: $X_0^1, \ldots, X_0^N$ i.i.d. from η_0

for $j = 1$ to J **do**

  for $i = 1$ to N **do**

    Run trajectory i until next level $\{\Phi(X_s^i) > L_j\}$ or final time τ_i (the first reached)

  end for

  Discard trajectories that did not reach L_j

  Set N_j the number of remaining trajectories, and I_j the set of their indices

  for $i \in \{1, \ldots, N\} \setminus I_j$ **do**

    Choose uniformly at random an index in I_j, clone it, and replace former trajectory i by it

  end for

end for

Estimate the probability of the rare event by

$$\hat{p}_{\text{smc}} = \left\{ \prod_{j=1}^{J} \frac{N_j}{N} \right\} \times \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_R(X_{\tau_i}^i)$$

This version is sometimes referred to as Fixed Effort (in opposition to Fixed Splitting, see [23]) but we propose to call it the Sequential Monte-Carlo (SMC) variant of multilevel splitting. Indeed, it is in fact a special case of a very general particle method usually named Sequential Monte-Carlo (see for example [22]). It was originally designed for non-linear filtering problems arising in Signal Processing, but has been reshaped as a very powerful abstract framework related to discrete time Feynman-Kac formulae in the monographs [18, 19]. We will see in Section 3 how these general results can be used to deduce mathematical properties of Algorithm 2.

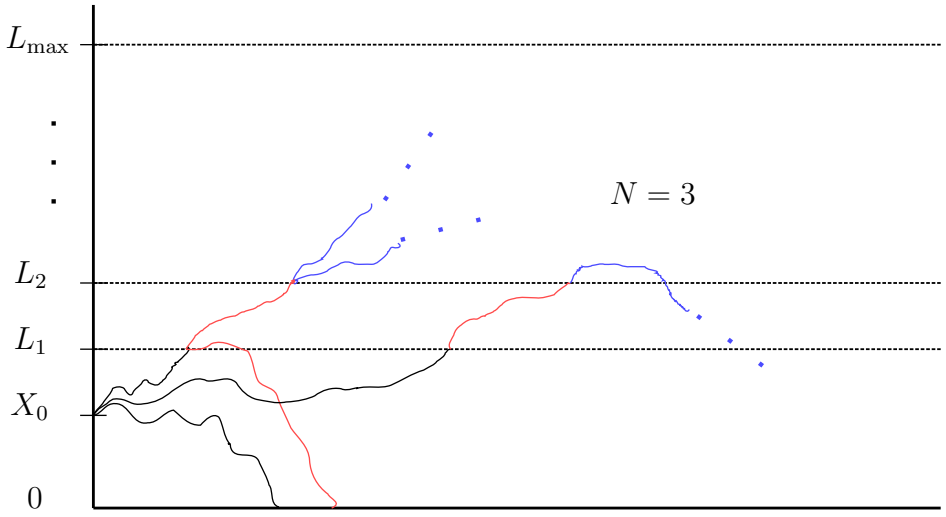

Figure 2: Algorithm 2 (SMC), where X is to reach $L_{\max}$ before 0.

Recall that, in the static case, X is a random vector in $\mathbb{R}^d$, S is a function from $\mathbb{R}^d$ to $\mathbb{R}$, and one wants to estimate the probability $p = \mathbb{P}(S(X) > q)$ where q is known and such that p is strictly positive but very low. The corresponding algorithm in this context was proposed and studied by several others, see for example [20] and [5]. More recently, the authors of [27] propose a framework that encompasses both the static and the dynamic case.

2.3 Adaptive Multilevel Splitting

To make the algorithm even more easy to use, the next idea is to set the levels adaptively. Instead of choosing the levels to be reached, one chooses the number K of trajectories to discard out of N, which is equivalent to choosing the proportion $(N - K)/N$ to be kept. The level used at each iteration is then an empirical quantile of the maximal levels reached by the set of trajectories. The resulting algorithm, known as Adaptive Multilevel Splitting, is given as Algorithm 3 and illustrated in Figure 3. It was first mentioned in [23], and then formalized and studied in dimension 1 in [13].

An argument to go from Algorithms 1 and 2 to Algorithm 3 is to consider the variance of a standard splitting approach, and to remark that the optimal way to choose the intermediate levels is to fix them such that the successive conditional probabilities are constant. This point is detailed in Section 3.5.

Note that depending on the choice of K, which can depend on N, we can have two interesting regimes. First in the case where $(N-K)/N = \theta \in (0, 1)$ (see [13]), and another regime when $K = 1$, meaning that $(N-K)/N = 1-1/N$

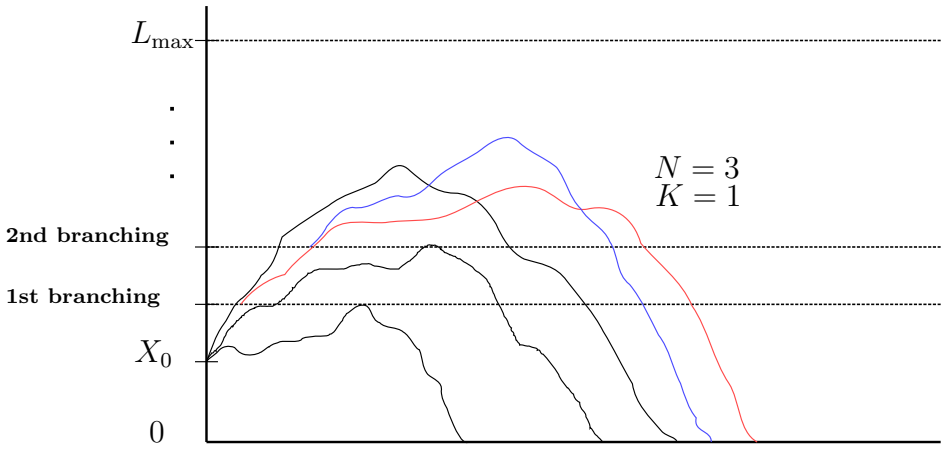

Figure 3: Algorithm 3 (AMS), where X is to reach $L_{\max}$ before 0.

(see [15]). The case $K = N$ is not interesting because it is equivalent to a naive Monte-Carlo method. Also notice that if, at some stage m, one has $I_m = \{1, \ldots, N\}$, meaning that $M_{(K)} = \ldots = M_{(N)}$, then the algorithm is stuck and there is a failure. However, there are some ways to avoid this pathological situation (see Remark 2.1 below).

It turns out that one can adapt easily the AMS method to the static case. The case $(N - K)/N = \theta \in (0, 1)$ is then often referred to as Subset Simulation (see for example [2, 3] and [10, 14]). The case $(N - K)/N = 1 - 1/N$ was introduced and studied in [28] and shares some connections with Nested Sampling (see [41, 42] and [17]).

Coming back to Algorithm 3, a first remark is that the number of iterations is random, but that does not mean the computing time is out of control. Indeed, if $(N-K)/N \to \theta \in (0, 1)$, then it is easily checked formally that the number of iterations converges in probability when $N \to \infty$ to $\lfloor \frac{\log p_{L_{\max}}}{\log \theta} \rfloor$. This was proved rigorously in dimension 1 in [13], and in the static case in [14], Theorem 3.1, whose proof can easily be adapted to the present framework.

When $(N-K)/N \to 1$, some work has been done when $K = 1$, the so-called "last particle" case, which corresponds to the maximum number of iterations. Then it is shown in [12], for X a uniformly elliptic diffusion, that the number of iterations is of order $O_{\mathbb{P}}(-N \log p)$. Hence it grows linearly with N (as usual) but only logarithmically with p, which is a nice property when p is very low.

A second remark is that, in Algorithm 3, the number K_m of discarded trajectories becomes random when more than one trajectory have the same current

Algorithm 3 Adaptive Multilevel Splitting (AMS)

Require: Initial distribution η_0, Importance Function Φ, sample size N, minimal number K of trajectories to discard at each step, final level $L_{\max}$

Initialization: $X_0^1, \ldots, X_0^N$ i.i.d. from η_0

Set $m \leftarrow 0$ (iteration index)

for $i = 1$ to N **do**

Run each trajectory to its end τ_i

Set $M_i \leftarrow \max_{0 \leqslant s \leqslant \tau_i} \Phi(X_s^i)$

end for

Sort the M_i's from low to high, so that $M_{(1)} \leqslant \ldots \leqslant M_{(N)}$

Set current level $L \leftarrow M_{(K)}$

while $L < L_{\max}$ **do**

$m \leftarrow m + 1$

Discard all the trajectories for which $M_i \leqslant L$

Let K_m be the number of such trajectories (hence $K_m \geqslant K$)

Define I_m as the set of indices of the remaining trajectories

for $i \in \{1, \ldots, N\} \setminus I_m$ **do**

Choose uniformly at random an index in I_m

Clone the corresponding trajectory until the first time it enters $\{\Phi > L\}$

From that time, simulate the cloned trajectory up to its end time τ_i

Replace the trajectory with index i by that new trajectory

Set $M_i \leftarrow \max_{0 \leqslant s \leqslant \tau_i} \Phi(X_s^i)$

end for

Sort the M_i's from low to high, so that $M_{(1)} \leqslant \ldots \leqslant M_{(N)}$

Set current level $L \leftarrow M_{(K)}$

end while

Set $M = m$ the total number of iterations

Estimate the probability of the rare event by

$$\hat{p}_{\text{ams}} = \left\{ \prod_{m=1}^{M} \frac{N - K_m}{N} \right\} \times \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_R(X_{\tau_i}^i)$$

score L. If we simplify and discard exactly K at each iteration, the algorithm is still usable, but unsuited for parallelization due to the presence of a (small) bias. This will be discussed in more details in Section 3.2.

Remark 2.1. *[A resampling variant] There is a trick which enables to treat differently the equality case in the order statistics $M_{(1)} \leqslant \ldots \leqslant M_{(N)}$ in Algorithm 3 which may result in $K_m > K$. The variant is as follows:*

- *First impose arbitrarily a total order among the particles' scores $M_{(1)} \leqslant \ldots \leqslant M_{(N)}$ by choosing it uniformly at random among the compatible ones.*
- *Second, discard the K first trajectories with respect to that order, instead of the $K_m \geqslant K$ trajectories with a lower or equal score.*
- *Third and last, the duplicated remaining particles trajectories are cloned until either the first time they enter $\{\Phi \geqslant L\}$, or the first time they enter $\{\Phi > L\}$ (if the latter exists). Note that a decision rule has to be chosen here.*

The probability of the rare event is now estimated by

$$\left\{ \prod_{m=1}^{M} \frac{N-K}{N} \right\} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_R(X_{\tau_i}^i).$$

This variant, with an appropriate decision rule in the third step, is expected to obey similar theoretical results (unbiasedness, consistency, CLT), although no rigorous proof is currently available. One advantage of this variant is that it usually prevents extinction, the latter being replaced by more trials attempting to strictly increase the current level.

One should also remark that this variant has been numerically tested in Section 5.1 of [8] (the so-called Version 2), in the case where the remaining particles trajectories are cloned until the first time they enter $\{\Phi \geqslant L\}$. The authors observe that the associated estimator of the rare event probability is then biased. For other, unbiased variants which prevent extinctions, we refer the interested reader to Section 3.5.1 *in [8].*

3 Mathematical results

3.1 Some notation

We will denote S_ℓ the hitting time of the open threshold ℓ, that is

$$S_\ell = \inf\{s,\ \Phi(X_s) > \ell\}.$$

For simplicity in the presentation, from now on we will assume that the rare event is exactly

$$X_\tau \in R \iff S_{L_{\max}} \leqslant \tau, \tag{3.1}$$

so that the trajectories contributing to the rare event are exactly those who reach the threshold $\{\Phi > L_{\max}\}$. In particular, this implies that the rare event probability becomes

$$p = p_{L_{\max}} = \mathbb{P}(S_{L_{\max}} \leqslant \tau).$$

We will also need the following "committor-like" function $q(\varphi)$, defined for any test function φ by

$$q(\varphi)(x) := \mathbb{E}[\varphi(X_\tau)\mathbf{1}_R(X_\tau)|X_0 = x]. \tag{3.2}$$

Note that, by construction, the committor function mentioned earlier is just

$$q^\star = q(\mathbf{1}).$$

Let us then define the probability measure η as the distribution of X_τ, given that $X_\tau \in R$. That is, for every test function φ,

$$\eta(\varphi) = \frac{1}{p}\mathbb{E}[\varphi(X_\tau)\mathbf{1}_R(X_\tau)] = \frac{1}{p}\mathbb{E}[\varphi(X_\tau)\mathbf{1}_{S_{L_{\max}}\leqslant\tau}] = \frac{1}{p}\eta_0(q(\varphi)).$$

We also introduce the unnormalized measure

$$\gamma(\varphi) = p\,\eta(\phi) = \mathbb{E}[\varphi(X_\tau)\mathbf{1}_R(X_\tau)] = \mathbb{E}[\varphi(X_\tau)\mathbf{1}_{S_{L_{\max}}\leqslant\tau}] = \eta_0(q(\varphi)).$$

Let us denote

$$\hat{p}_{\mathrm{ms}},\ \hat{p}_{\mathrm{smc}} \ \text{ and } \ \hat{p}_{\mathrm{ams}}$$

the estimates of $p = \mathbb{P}(X_\tau \in R)$ for the previous algorithms, meaning respectively Multilevel Splitting (Algorithm 1), Sequential Monte-Carlo (Algorithm 2) and Adaptive Multilevel Splitting (Algorithm 3). Note that condition (3.1) simplifies formulas since $\mathbf{1}_R(X^i_{\tau_i}) = 1$ almost surely in each estimator.

Let us also denote

$$(X^1_{\mathrm{ms}}, \dots, X^{N_J}_{\mathrm{ms}}) \qquad \text{and} \qquad \hat{\eta}_{\mathrm{ms}} = \frac{1}{N_J}\sum_{i=1}^{N_J}\delta_{X^i_{\mathrm{ms}}}$$

the set of particles at the end of Algorithm 1 and the corresponding empirical measure. Accordingly, $\hat{\gamma}_{\mathrm{ms}} = \hat{p}_{\mathrm{ms}}\,\hat{\eta}_{\mathrm{ms}}$ stands for the unnormalized empirical measure. Note that $\gamma(\mathbf{1}) = p$ and $\hat{\gamma}_{\mathrm{ms}}(\mathbf{1}) = \hat{p}_{\mathrm{ms}}$. We define $\hat{\eta}_{\mathrm{smc}}$, $\hat{\gamma}_{\mathrm{smc}}$, $\hat{\eta}_{\mathrm{ams}}$, $\hat{\gamma}_{\mathrm{ams}}$ in the same manner.

3.2 A few words about the bias

It turns out that the estimates $\hat{p}_k$ of the probability p for $k \in \{\mathrm{ms}, \mathrm{smc}, \mathrm{ams}\}$, and more generally the unnormalized measures $\hat{\gamma}_k(\varphi)$ for any test function φ, are all unbiased. For Algorithm 1 it is quite straightforward, see [1].

For Algorithm 2, it is a general property of Sequential Monte-Carlo methods, as expected from a Monte-Carlo method used to compute an expectation, here in the form of an unnormalized Feynman-Kac measure (see for example [18] page 112). The key point is to remark that $\hat{\gamma}_{\mathrm{smc}}(\varphi)$ can be expressed as the terminal value of a martingale with initial value given by the empirical average of $q(\varphi)$ obtained after the initialization, that is $\frac{1}{N}\sum_{i=1}^{N} q(\varphi)(X_0^i)$ which is obviously unbiased.

For Algorithm 3, the property is a consequence of a very general result given in [6]. The discussion in Section 3.4 of the present paper explains (without rigorous proof) that Algorithm 3 can be interpreted as a limit of Algorithm 2 when the number J of levels goes to infinity. This suggests that the unbiasedness property is inherited by Algorithm 3. Alternatively, a specific martingale analysis could be performed as done in a diffusive case in [12]. In Algorithm 3, it is important to take care of the multiple values for the maximum of Φ along the set of trajectories. If no care is taken, then the resulting algorithm might be biased. Note that in [13] no such result was obtained precisely because no care was taken of these multiple values.

The variant in Remark 2.1 is also expected to yield unbiased estimators of unnormalized quantities, although up to now there exists no rigorous proof for this variant. The equality case requires to handle in some way a martingale indexed by two indices, endowed with lexicographic order.

It is known that the normalized estimates $\hat{\eta}_k(\varphi)$ are all biased with a bias of order $1/N$ for $k \in \{\mathrm{smc}, \mathrm{ams}\}$. The estimate $\hat{\eta}_{\mathrm{ms}}(\varphi)$ is expected to be biased as well but has still not been carefully studied.

If we want to parallelize the computation, and we are only interested in the estimate of the probability p, the unbiasedness property allows us to run several independent versions of the algorithm in parallel, and then take the empirical mean of all the resulting estimates. If we want a conditional mean of an observable, then we have to parallelize within each algorithm. In Algorithm 3, we can use for example K as the number of trajectories we can resimulate in parallel. Note that they will not have the same length, so they all need to wait for the longest one to finish.

3.3 Sequential Monte-Carlo (discrete levels)

This section details theoretical results for Algorithm 2. Let us recall that S_{L_j} stands for the hitting time of L_j, i.e. $S_{L_j} = \inf\{s,\ \Phi(X_s) > L_j\}$ where the successive levels $-\infty = L_0 < L_1 < \cdots < L_J = L_{\max}$ are given a priori. Recall that $p_{L_j} = \mathbb{P}\left(S_{L_j} \leqslant \tau\right)$ and

$$\theta_j = p_{L_j}/p_{L_{j-1}} = \mathbb{P}(S_{L_j} \leqslant \tau | S_{L_{j-1}} \leqslant \tau).$$

First we need to mention that there is a small but non zero probability of extinction, that is all the particles fail to reach some level L_j. In that case, we consider that we estimate the probability p as 0, just as we would do in naive Monte-Carlo. Fortunately, the probability of such a failure is soon very small, meaning with a reasonable number N of particles. A simple version of Theorem 7.4.1 in [18] is given in the following proposition. Recall that, in all the paper, we assume that $p > 0$.

Proposition 1. *Define the extinction event:*

$$\mathcal{E} := \{\textit{Algorithm 2 fails with an extinction}\}.$$

There exist two constants (depending on the problem being solved) $A > 0$ and $B > 0$ such that for all $N \geqslant 1$,

$$\mathbb{P}(\mathcal{E}) \leqslant Ae^{-BN}.$$

Note that there are versions of the algorithm without extinction (see [33] for a biased version, [1] for an unbiased version, and [21] for a generalization to non-negative potentials), but then the computation time is not bounded.

We also have a law of large numbers which ensures the convergence of the algorithm, as well as a Central Limit Theorem. The asymptotic variance in the latter is of special interest. Its definition requires to introduce the following conditional distributions:

$$\eta_{L_j}(\varphi) := \mathbb{E}[\varphi(X_{S_{L_j}}) | S_{L_j} \leqslant \tau].$$

Recall that the committor function is defined by $q^\star(x) = \mathbb{P}(X_\tau \in R | X_0 = x)$.

Theorem 3.1. *The estimator $\hat{p}_{\text{smc}}$ satisfies*

$$\mathbf{1}_{\mathcal{E}^c}\sqrt{N}\left(\hat{p}_{\text{smc}} - p\right) \xrightarrow[N\to\infty]{\text{Law}} \mathcal{N}(0, \sigma^2)$$

where

$$\sigma^2 = \sum_{j=1}^{J-1} \theta_j \left(p_{L_{j-1}}^2 - p_{L_j}^2\right) \mathbb{V}_{\eta_{L_j}}(q^\star) + p^2 \sum_{j=1}^{J} \left(\frac{1}{\theta_j} - 1\right). \tag{3.3}$$

A similar CLT can be obtained for the final empirical distribution

$$\hat{\eta}_{\text{smc}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{\tau}^{i}}$$

which estimates the conditional distribution $\eta = \mathcal{L}(X_\tau | X_\tau \in R)$, as obtained at the end of Algorithm 2 (see Section 9.4.2 of [18]):

$$\mathbf{1}_{\mathcal{E}^c}\sqrt{N}\left(\hat{\eta}_{\text{smc}}(\varphi) - \eta(\varphi)\right) \xrightarrow[N\to\infty]{\text{Law}} \mathcal{N}(0, \sigma^2(\varphi - \eta(\varphi)))$$

with an asymptotic variance given for any bounded observable φ by

$$\sigma^2(\varphi) = \mathbb{V}_{\eta}(\varphi) + \sum_{j=1}^{J} \theta_j \frac{p_{L_j}^2 - p_{L_{j-1}}^2}{p^2} \mathbb{V}_{\eta_{L_j}}(q(\varphi - \eta(\varphi))). \tag{3.4}$$

Using the same technical apparatus (see Section 9.4.2 of [18]), the latter CLT can be extended to path observables of the empirical distribution

$$\hat{\eta}_{\text{path}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^i_{0\leqslant s\leqslant \tau^i}} \tag{3.5}$$

which estimates the law η_{path} of the full trajectory conditional on the rare event. The generalized pathwise variance is the same as in (3.4), except that the observable dependent committor function

$$\mathbb{V}_{\eta_{L_j}}(q(\varphi)) = \mathbb{V}\left[\mathbb{E}\left(\varphi(X_\tau)\mathbf{1}_{S_{L_{\max}}\leqslant\tau} | X_{S_{L_j}}\right)\right].$$

is now replaced by the variance of the analogous path functional

$$\mathbb{V}\left[\mathbb{E}\left(\varphi(X_{0\leqslant s\leqslant\tau})\mathbf{1}_{S_{L_{\max}}\leqslant\tau} | X_{0\leqslant s\leqslant S_{L_j}}\right)\right]. \tag{3.6}$$

Estimate of the variance

Recently, [35] proposed and showed the convergence of an estimator of the variance applicable to a range of SMC algorithms, including particle filter. The theory applies to a modification of Algorithm 2, in which the resampling of particle is instead a multinomial resampling. By that we mean that we do not keep necessarily all the successful particles at the next level, but make N i.i.d. uniform draws among them. Of course, this will lead to more dependence than Algorithm 2, and thus more variance. We nevertheless think that even if the theory does not apply strictly to Algorithm 2, this

variance estimate can in practice give a useful insight. But clearly, more investigations are needed here.

If we restrict it to our rare event setting, with Algorithm 2, we consider

$$V_J^N(\varphi) = \eta_J^N(\varphi)^2 - \frac{N^{J-1}}{(N-1)^{J+1}} \sum_{A^i_{0,J} \neq A^k_{0,J}} \varphi(X_J^i)\varphi(X_J^k),$$

where $A^i_{0,J}$ is the ancestor at time 0 of X^i_J. Then $NV_J^N(\varphi)$ is a consistent estimator (as $N \to \infty$) of $\sigma^2(\varphi)$, and $N\hat{p}^2_{\text{smc}} V_J^N(1)$ a consistent estimator of σ^2. This estimator is actually an unbiased modification of the one proposed earlier in [16].

These estimators are almost free to compute because we only need to store the ancestors of the particles all along the algorithm. The drawback is that if N is not large enough, all the particle will share a very small set of ancestors (or even the same one), and the estimators will not be useful.

3.4 Adaptive Multilevel Splitting (continuous levels)

Let us now discuss some theoretical results related to Algorithm 3.

Well-posedness and CLT for the case $K = 1$

For the case $K = 1$ ("last particle" version), some results have been obtained under additional assumptions. For simplicity, we will only explain here the diffusive case. We refer the interested reader to [12] for details and complements.

Suppose that the process $(X_s)_{s \geqslant 0}$ is a strong solution of a Stochastic Differential Equation

$$dX_s = b(X_s) + \sigma(X_s)dW_s,$$

where b and σ are smooth. Moreover, we assume that the level function Φ is smooth and there exists $\delta > 0$ such that $(\nabla\Phi)^T a \nabla\Phi \geqslant \delta$.

In this context, Algorithm 3 is well-defined in the sense that, almost surely, only one particle is discarded at each step (no equality in the scores) and the algorithm stops after a finite number of steps, meaning that a.s. $M < \infty$.

We then have a law of large numbers which ensures the convergence of the algorithm when $N \to \infty$, as well as a Central Limit Theorem. Interestingly, as will be explained later, the asymptotic variance in Theorem 3.2 can be seen as the "continuous levels" limit of the Sequential Monte-Carlo variance

("discrete levels" case) as given in (3.3). More precisely, let us write the conditional probability

$$\eta_\ell(\varphi) := \mathbb{E}[\varphi(X_{S_\ell})|S_\ell \leqslant \tau].$$

In the same way we denote

$$p_\ell = \mathbb{P}(S_\ell \leqslant \tau).$$

We stress that, as before, ℓ is not an integer, but a real number corresponding to a level. In this respect, we recall that $p = p_{L_{\max}}$ is assumed strictly greater than 0.

Theorem 3.2. *The unbiased estimator $\hat{p}_{\text{ams}}$ satisfies the CLT*

$$\sqrt{N}\left(\hat{p}_{\text{ams}} - p\right) \xrightarrow[N\to\infty]{\text{Law}} \mathcal{N}(0, \sigma^2)$$

where

$$\sigma^2 = -p^2 \ln p - \int_{-\infty}^{L_{\max}} \mathbb{V}_{\eta_\ell}(q^\star) d\left(p_\ell^2\right) = -p^2 \ln p - 2\int_{-\infty}^{L_{\max}} \mathbb{V}_{\eta_\ell}(q^\star) p_\ell dp_\ell. \quad (3.7)$$

In these formulas, the integration is with respect to the level ℓ, meaning that ℓ goes from $-\infty$ to $L_{\max}$. Also note that in the expression of the asymptotic variance σ^2, both terms are positive since $0 < p < 1$, and $\ell \mapsto p_\ell$ is decreasing, making dp_ℓ negative. As in the discrete case (SMC), a CLT can be obtained for the final empirical distribution at the end of Algorithm 3, which we also denote

$$\hat{\eta}_{\text{ams}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{X^i_{\tau^i}}.$$

The latter estimates the conditional distribution $\eta = \mathcal{L}(X_\tau | X_\tau \in R)$. We prove in [12] the following CLT:

Theorem 3.3. *If φ is bounded and continuous, then*

$$\sqrt{N}\left(\hat{\eta}_{\text{ams}}(\varphi) - \eta(\varphi)\right) \xrightarrow[N\to\infty]{\text{Law}} \mathcal{N}(0, \sigma^2(\varphi - \eta(\varphi)))$$

with

$$\sigma^2(\varphi) = \mathbb{V}_\eta(\varphi) - \int_{-\infty}^{L_{\max}} \mathbb{V}_{\eta_\ell}(q(\varphi)) d\left(p_\ell^2/p^2\right). \quad (3.8)$$

Here again, the latter CLT can been extended to pathwise bounded and continuous – with respect to uniform convergence on compact time intervals – observables of the empirical distribution

$$\hat{\eta}_{\text{path}} = \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i_{0 \leqslant s \leqslant \tau^i}},$$

which estimates the law η_{path} of the full trajectory conditional on the rare event. The generalized pathwise variance is the same as (3.5), except that the variance of the observable dependent committor function $\mathbb{V}_{\eta_l}(q(\varphi))$ is again replaced by (3.6). The extension of the CLT to non continuous observables remains open.

Interpretation as a limit of the discrete levels case when $K = 1$

Now we would like to discuss a topic that has not been investigated in the literature so far, namely the fact that Algorithm 3 (AMS), which has been introduced as an *adaptive* version of the discrete levels Algorithm 2 (SMC), can in fact be understood as a *continuous levels* limit of the latter.

For this purpose, let us assume that the discrete levels are chosen of the form

$$L_j = L_1 + (j-1)\frac{L_{\max} - L_1}{J-1} \qquad \text{and} \qquad \Phi(X_0) \geqslant L_1 \quad a.s.$$

Then, modify slightly the formulation of Algorithm 2 by simulating in the main loop the whole path – up to the final stopping time τ_i – of each *newborn* particle i, instead of simulating it level brackets by level brackets. Note that this modification does not change the probability distribution of the whole particle system.

It is then easy to check (e.g., with a drawing) that Algorithm 3 is *exactly* Algorithm 2 in the case where the function Φ takes its values in a finite set, for instance using Φ_ε approximated by

$$\Phi_\varepsilon = \varepsilon \lfloor \Phi / \varepsilon \rfloor. \tag{3.9}$$

For J large compared to N, most iterations in Algorithm 2 become useless (nothing happens), except when the level value L_j coincides with the value of the smallest particle score within the current particle system. Note that iterations in Algorithm 3 exclusively select the latter events: this explains why the iteration index was denoted with a different letter, namely m instead of j.

Finally, for instance when X is a pure jump process which takes only a finite number of values, it is clear that the latter algorithm will be the same if performed with any Φ_ε, provided that ε is small enough. This implies that, at least formally, Algorithm 3 is the "limit" when $J \to \infty$ of Algorithm 2.

It is then possible to compute formally the $J \to \infty$ limit in the asymptotic variances obtained in the previous CLTs. Indeed, suppose for simplicity that $p_{L_1} = 1$, then one has

$$\sum_{j=1}^{J} \left(\frac{1}{\theta_j} - 1 \right) = \sum_{j=1}^{J} - \frac{p_{L_j} - p_{L_{j-1}}}{p_{L_j}} \to - \int_{L_1}^{L_{\max}} \frac{dp_\ell}{p_\ell} = - \ln p_{L_{\max}} = - \ln p.$$

In the same way, for any continuous function f,

$$\sum_{j=1}^{J} f(L_j) \theta_j \left(p_{L_{j-1}}^2 - p_{L_j}^2 \right) \to - \int_{L_1}^{L_{\max}} f(\ell) \; d(p_\ell^2).$$

We thus recover the continuous levels variances (3.7) and (3.8) from the limits, when $J \to \infty$, of the discrete levels variances (3.3) and (3.4). In turn, this suggests that the CLTs of Theorems 3.2 and 3.3 are in fact valid in a completely general setting, as soon as the extinction probability becomes small enough when $N \to \infty$.

Finally, note that a similar interpretation of Algorithm 3 as a limit of Algorithm 2 in the case where $K \neq 1$ is possible, but requires a substantial modification of Algorithm 2. In such a modified algorithm, the number of particles a each iteration is non constant and decreasing – particles are not duplicated – until at least K particles are killed, which triggers the creation of new particles to recover the maximal population size N.

About the asymptotic behavior for the case $K/N \to \theta > p$

Interestingly enough, a kind of converse remark can be made: in the case where $1 - K/N \to \theta \in (p, 1)$ when $N \to \infty$, Algorithm 3 is very similar to the discrete levels case of Algorithm 2, for the specific choice of equiprobable levels, meaning that the levels L_j, $j \in \{1, \ldots, J-1\}$, are such that

$$\begin{cases} J = \lfloor \log p / \log \theta \rfloor + 1 \\ \theta_j = \theta \quad \text{for } \; j = 1, \ldots, J-1 \\ \theta_J = p/\theta^{J-1} \in (\theta, 1]. \end{cases} \tag{3.10}$$

This point view has been systematically studied in [14] where a CLT is obtained in the static case. However, the assumptions therein, in particular the use of multinomial resampling, do not apply to Algorithm 3.

Nonetheless, the relationship between the two algorithms can be described assuming at least that for each $j \in \{1, \ldots, J-1\}$, there exists a unique level L_j such that $\mathbb{P}(\tau \leqslant S_{L_j}) = p_{L_j} = \theta^j$. In that case, the random level reached at iteration $m \leqslant J-1$ of Algorithm 3, denoted $\hat{L}_m$, should converge in probability towards the deterministic level L_m associated with the probability θ^m. As a consequence, at each iteration in both algorithms, a fraction K/N (or an empirical estimation of it for Algorithm 2) of particles fail to reach the next level in the sequence $L_1 < \ldots < L_J$ (or in an empirical estimation of it for Algorithm 3), both algorithms being thus formally similar when $N \to +\infty$.

Finally, the results in [14] suggest that a CLT may be obtained for $\hat{p}_{\text{ams}}$ and $\hat{\eta}_{\text{ams}}$ with exactly the same variances as in (3.3) and (3.4) in the case (3.10). Although these results are strongly believed to be true, they still require a rigorous proof.

3.5 On the importance function

In practice, the main source of variance in Algorithms 2 and 3 comes from a bad choice of the importance function Φ (reaction coordinate). For this reason, it is of crucial interest to try to minimize the asymptotic variances (3.3) or (3.7) with respect to the choice of Φ.

For simplicity, we assume that the initial condition is deterministic $X_0 = x_0$, and that Φ is at least continuous. We can then remark that, on the one hand, the target open set $\{x,\ \Phi(x) > L_{\max}\}$ depends on Φ only through its boundary

$$\{x,\ \Phi(x) = L_{\max}\} \subset \mathbb{R}^d,$$

while, on the other hand, Algorithms 2 and 3 depend on all the intermediate level sets $\{x,\ \Phi(x) = \ell\}$ from $\ell = L_{\min} = \Phi(x_0)$ to $\ell = L_{\max}$.

This implies that the latter algorithms are unchanged when Φ is multiplied by a constant, so we can assume without loss of generality that $L_{\max} = 1$. More importantly, the target set is unchanged if Φ is modified while keeping the set $\{x,\ \Phi(x) = L_{\max}\}$ fixed. As a consequence, we will now try to optimize σ^2 in the set of continuous $\Phi \leqslant 1$ such that $\Phi = 1$ on the latter set.

We first start with the AMS variance as given by (3.7) (continuous levels case), and immediately remark that the term $-p^2 \ln p$ cannot be modified while the other term vanishes as soon as the following condition holds true:

$$\mathbb{V}_{\eta_\ell}(q^\star) = 0 \qquad \forall \ell \in [L_{\min}, L_{\max}]. \tag{3.11}$$

In the diffusive case – and, interestingly, only in the diffusive case –, that is when $(X_s)_{s\geqslant 0}$ has continuous trajectories, the condition is satisfied for an explicit choice of Φ which turns out to be (any function of) the committor function

$$\Phi = f(q^\star) \Rightarrow \mathbb{V}_{\eta_\ell}(q^\star) = 0,$$

where f is any continuous strictly increasing function. Indeed, by continuity of the trajectories, the conditional distribution η_ℓ has necessarily for each ℓ its support in the associated level set

$$\eta_\ell(\mathbf{1}_{\{\Phi=\ell\}}) := \mathbb{P}(\Phi(X_{S_\ell}) = \ell | S_\ell \leqslant \tau) = 1,$$

so that $\mathbb{V}_{\eta_\ell}(f^{-1}(\Phi)) = 0$.

Next, for the SMC asymptotic variance (3.3) (discrete levels case) and the choice $\Phi = f(q^\star)$ for the importance function, the remaining variance term reduces to

$$p^2 \sum_{j=1}^{J} \left(\frac{1}{\theta_j} - 1 \right),$$

which, in turn, can be optimized for each fixed J. Indeed, the only constraint is $\theta_1 \times \ldots \theta_J = p$, so that a standard convex optimization implies that the optimum is reached for $\theta_j = p^{1/J}$ for all j. As a consequence, the minimal possible variance is

$$\sigma^2 = p^2 J \left(p^{-1/J} - 1 \right),$$

which is decreasing towards $-p^2 \ln p$ when $J \to \infty$, i.e. when $\theta_j \to 1$. This accounts for the choice $K = 1$ in order to minimize the variance.

This simple remark also explains why in the AMS Algorithm 3 (with no equality in the scores for simplicity), the number K of particles which is discarded is the same at each step. In full generality, one could decide to discard K_1 particles at the first step, then K_2 at the second step, etc., run the same algorithm and consider the estimator

$$\hat{p}_{\text{ams}} = \left\{ \prod_{m=1}^{M} \frac{N - K_m}{N} \right\} \times \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_R(X_{\tau_i}^i).$$

But the constrained optimization problem above proves that, in the ideal situation where one would have the committor function at disposal, the best thing to do is to discard the same number of trajectories at each step. In other words, the idea is to minimize in $(\theta_1, \ldots, \theta_J)$ the lower bound for σ^2 given by (3.3).

One can also notice that the asymptotic variance (3.8) for the conditional empirical distribution can be bounded as follows (see Section 2.4 in [12])

$$\mathbb{V}_\eta(\varphi) \leqslant \sigma^2(\varphi - \eta(\varphi)) \leqslant \mathbb{V}_\eta(\varphi) + \|\varphi - \eta(\varphi)\|_\infty^2 \left(\frac{\sigma^2}{p^2} - \ln p\right),$$

with σ^2 as in (3.7). The lower bound is the variance we would get with an i.i.d. sample from η. As noticed above, at best the second term in the r.h.s. reduces to $-2\|\varphi - \eta(\varphi)\|_\infty^2 \ln p$.

Concerning the asymptotic variance σ^2 of $\hat{p}_{\text{ams}}$ in Theorem 3.2, we can also show (see Corollary 2.10 in [12]) that one always has

$$-p^2 \log p \leqslant \sigma^2 \leqslant 2p(1-p).$$

In other words, it is impossible with AMS to do better than $-p^2 \log p$, even with the optimal importance function at disposal, but it is also impossible to do worse than $2p(1-p)$, which is twice the variance of a naive Monte-Carlo method (see Section 1). In comparison, it is well-known that for Importance Sampling, there always exists an optimal (usually out of reach) sampling distribution such that the resulting variance is equal to 0 (see for example Section 6.2 in [36] for the connection with the committor function). But on the opposite, as mentioned in the introduction, a bad choice for the sampling distribution may lead to an infinite variance. Hence, the take-home message is that in the best case, Importance Sampling is much better than AMS, but in the worst case, Importance Sampling is much worse than AMS.

To conclude this section, let us say a few words about efficiency. To take into account both computational complexity and variance, Hammersley and Handscomb [29] have proposed to define the efficiency of a Monte-Carlo method as "inversely proportional to the product of the sampling variance and the amount of labour expended in obtaining this estimate." If we consider that the cost of the simulation of a single trajectory is one, than the inverse of the efficiency of a naive Monte-Carlo method to estimate p is just $p(1-p)$. For AMS with $K = 1$, it is shown in Corollary 2.9 of [12] that the number of iterations is $-N \log p + O_P(\sqrt{N})$. Taking into account the (quick)sorting of the particles and the fact that an iteration amounts to simulate one new trajectory, the complexity scales like $-N \log N \log p$. As explained above, the asymptotic variance satisfies $-p^2 \log p \leqslant \sigma^2 \leqslant 2p(1-p)$, hence the following bounds for the inverse of the efficiency of AMS:

$$(p \log p)^2 \log N \leqslant E_{\text{ams}}^{-1} \leqslant -2p(1-p) \log p \log N.$$

Therefore, in the best case, AMS is much more efficient than naive Monte-Carlo, whereas in the worst case, it is less efficient: larger variance (by a

factor 2) and larger complexity (by a factor $-\log p \log N$). However, notice that the latter happens only for a very bad choice of the importance function.

4 Examples and applications

4.1 Return times

A generic application, proposed in [37], to rare event probability estimation is to compute return times to some unfrequent event for a stationary ergodic process X with a.s. continuous trajectories. In this work, the return time is defined as the average of the hitting time of a given event with an initial condition X_0 distributed according to the stationary distribution. The considered event is often in the form $\{\Phi(x) \geqslant L_{\max}\}$ for some function Φ.

Let us denote $r(L_{\max})$ the expectation of the corresponding return time and let us assume that this return time is much larger than the mixing time of X, which happens for $L_{\max}$ large enough. In that asymptotics, the authors argue that the sequence of the hitting times of the considered event is distributed according to a Poisson process with parameter $1/r(L_{\max})$, so that the return time is explicitly related to the probability of reaching the considered event on a unit time interval.

Thus a standard approach to estimate is $r(L_{\max})$ is to use a block estimator: one simulates X for a long time $M * T$. On each time block $[(m-1)T, mT]$, let $s_m(L_{\max}) = 1$ if the level $L_{\max}$ is reached by X, and zero otherwise. One can then use the following block estimator, valid when $T \ll r(L_{\max})$:

$$\hat{r}_B^1(L_{\max}) = \frac{MT}{\sum_{m=1}^{M} s_m(L_{\max})}.$$

A better estimator, based on the exponential distribution, and this time valid when T and $r(L_{\max})$ are comparable, is the modified block estimator

$$\hat{r}_B^2(L_{\max}) = \frac{T}{\log(1 - \frac{1}{M}\sum_{m=1}^{M} s_m(L_{\max}))}.$$

Using AMS Algorithm 3, one can adapt the above estimator while using AMS to estimate the probability of reaching level $L_{\max}$ on $[0, T]$. In AMS, we have a set of M trajectories that evolve on $[0, T]$, and we denote by $\hat{p}_{\text{ams}}$ the associated estimator of the probability of reaching $L_{\max}$ defined in Algorithm 3. One can then adapt the above estimator as follows

$$\hat{r}_B^3(L_{\max}) = \frac{T}{\log(1 - \hat{p}_{\text{ams}})}.$$

Note that, although the initial distribution in the latter AMS variant is supposed to be the stationary distribution of the process, this assumption could be relaxed if we still consider the case where the mixing time of the process is very small as compared to T, so that the influence of a non stationary initial condition is very limited.

Finally, because AMS is used with a deterministic final time, it is worth noted that the best importance function should depend on time (see [37] section IIIC, see also [7]). Also note that using all these trajectories with intermediate levels ℓ, it is easy to generate an approximation of the curve $\ell \mapsto r(\ell)$.

4.2 Neutronics

The AMS Algorithm 3 was successfully used for problems of neutral particle transport in [40, 39]. The adaptation of AMS is quite straightforward, with care to be taken on how to efficiently store the particles trajectories. As we have seen above, the choice of the importance function/reaction coordinate is crucial. Here the authors were able to use some code already developed for other variance reduction techniques as a good importance function, leading to a high figure of merit. The authors also used the fact that AMS does not only give an approximation of a rare event probability, but also a set of empirical trajectories reaching it, which can in turn be used to estimate any observable, given the rare event.

4.3 Air Traffic Management

The Multilevel Splitting approach to rare events simulation is now quite widely used in Air Traffic Management (ATM). It is an important application because it has a real life impact, it is not just a simulation to illustrate a theoretical result. Problems faced by people working in ATM include checking that a proposed new regulation in air traffic will not increase the risk of accident (e.g., two planes crashing on each other) above the required safety level. Typically, these safety levels are very low, and therefore one clearly needs to use some variance reduction techniques.

The splitting approach is well suited to this application because it allows one to use a large simulation code already developed separately to simulate air traffic scenarios. The interesting point here is that the splitting paradigm can be used in conjunction with a complex simulation code, that was not initially designed for this purpose. A detailed account of this application can be found in [4] and references therein. Note that due to the presence

of a discrete component in the state space, the SMC Algorithm 2 had to be adapted to be really efficient to make sure that all the discrete modes are represented.

4.4 Molecular dynamics

Recently, the AMS Algorithm 3 (case $K = 1$) was used in a real Molecular Dynamics problem in [43]. The problem is to estimate the expected dissociation time between a protein and a ligand. An abstract view of the problem can be given as follow. The state of the system (configuration of the molecules) can be modeled by a stochastic differential equation in $\mathbb{R}^d$ (d can be large):

$$dX_t = -\nabla V(X_t)dt + \sqrt{2\varepsilon}dW_t,$$

where V is a (sufficiently smooth) potential, ε a temperature parameter, and W a standard Brownian motion.

We consider two sets of configurations A and B. A is the set where the protein and its ligand are tightly associated, and B the set where they are completely dissociated. Starting in A, we want to estimate the expected time until the diffusion hits the set B. A is actually a neighborhood of a local minimum of V, and it is assumed to be recurrent. A trajectory leaving a neighborhood A^δ of A and hitting B before going back to A is called a reactive trajectory. Algorithm 3 is designed to sample these reactive trajectories efficiently. It represents a rare event because the timescale at which the diffusion can be simulated is orders of magnitudes lower than the timescale of the transitions from A to B.

The solution is not completely straightforward here because AMS simulates reactive trajectories, and estimates their duration, but not the time spent leaving A^δ and going back to A again without reaching B. We are also not in the situation considered in Section 4.1 because the duration of a reactive trajectory is much smaller than the time needed to see one happen. The approach in [43], also found in [15], is to write the expectation of the transition time T from A to B as

$$\mathbb{E}[T] = \frac{1-p}{p}\mathbb{E}[T_1] + \mathbb{E}[T_2],$$

where p is the (very low) probability of a reactive trajectory, T_1 is the time taken starting from A (at equilibrium) to leave A^δ, and go back directly to A, and T_2 the time taken starting from A to leave A^δ and go to B without going back to A (reactive trajectory).

Note that this decomposition is exact if the boundaries of A and A^δ are level sets of the committor $q^\star$ (see [15] section 4.2). We can nevertheless reasonably conjecture that in most practical cases, it will provide a sharp approximation. We can use Algorithm 3 to estimate p and $\mathbb{E}[T_2]$, and a direct numerical simulation to estimate $\mathbb{E}[T_1]$ (no rare events here), and thus get an estimate of $\mathbb{E}[T]$.

Finally, let us mention another recent application of AMS to isomerization when using the NAMD simulation program [38].

Acknowledgements

This work was partially supported by the French Agence Nationale de la Recherche, under grant ANR-14-CE23-0012, and by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement number 614492.

References

[1] M. Amrein and H.R. Künsch. A variant of importance splitting for rare event estimation: Fixed number of successes. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 21(2):13, 2011.

[2] S.K. Au and J.L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

[3] S.K. Au and J.L. Beck. Subset simulation and its application to seismic risk based on dynamic analysis. *Journal of Engineering Mechanics*, 129(8):901–917, 2003.

[4] H.A.P. Blom, G.J. Bakker, and J. Krystul. Rare event estimation for a large-scale stochastic hybrid system with air traffic application. In *Rare event simulation using Monte Carlo methods*, pages 193–214. Wiley, Chichester, 2009.

[5] Z.I. Botev and D.P. Kroese. An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting. *Methodol. Comput. Appl. Probab.*, 10(4):471–505, 2008.

[6] C.-E. Bréhier, M. Gazeau, L. Goudenège, T. Lelièvre, and M. Rousset. Unbiasedness of some generalized adaptive multilevel splitting algorithms. *Ann. Appl. Probab.*, 26(6):3559–3601, 2016.

[7] C.-E. Bréhier and T. Lelièvre. On a new class of score functions to estimate tail probabilities of some stochastic processes with adaptive multilevel splitting. *arXiv preprint arXiv:1811.06289*, 2018.

[8] Charles-Edouard Bréhier, Maxime Gazeau, Ludovic Goudenège, Tony Lelièvre, and Mathias Rousset. Unbiasedness of some generalized adaptive multilevel splitting algorithms. *arXiv preprint arXiv:1505.02674*, 2015.

[9] J.A. Bucklew. *Introduction to rare event simulation*. Springer Series in Statistics. Springer-Verlag, New York, 2004.

[10] F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Stat. Comput.*, 22(3):795–808, 2012.

[11] F. Cérou, P. Del Moral, F. Le Gland, and P. Lezaud. Genetic genealogical models in rare event analysis. *ALEA Lat. Am. J. Probab. Math. Stat.*, 1:181–203, 2006.

[12] F. Cérou, B. Delyon, A. Guyader, and M. Rousset. On the Asymptotic Normality of Adaptive Multilevel Splitting. *SIAM/ASA Journal on Uncertainty Quantification*, 7(1):1–30, 2019.

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

[14] F. Cérou and A. Guyader. Fluctuation analysis of adaptive multilevel splitting. *Ann. Appl. Probab.*, 26(6):3319–3380, 2016.

[15] F. Cérou, A. Guyader, T. Lelièvre, and D. Pommier. A multiple replica approach to simulate reactive trajectories. *The Journal of Chemical Physics*, 134(5):054108, 2011.

[16] H.P. Chan and T.L. Lai. A general theory of particle filters in hidden Markov models and some applications. *Ann. Statist.*, 41(6):2877–2904, 2013.

[17] N. Chopin and C.P. Robert. Properties of nested sampling. *Biometrika*, 97(3):741–755, 2010.

[18] P. Del Moral. *Feynman-Kac formulae, Genealogical and interacting particle systems with applications*. Springer-Verlag, New York, 2004.

[19] P. Del Moral. *Mean field simulation for Monte Carlo integration*, volume 126 of *Monographs on Statistics and Applied Probability*. CRC Press, Boca Raton, FL, 2013.

[20] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(3):411–436, 2006.

[21] P. Del Moral, A. Jasra, A. Lee, C. Yau, and X. Zhang. The Alive Particle Filter and Its Use in Particle Markov Chain Monte Carlo. *Stoch. Anal. Appl.*, 33(6):943–974, 2015.

[22] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, New York, 2001.

[23] M.J.J. Garvels. *The splitting method in rare event simulation*. Thesis, University of Twente, 2000.

[24] M.B. Giles. Multilevel monte carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[25] M.B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.

[26] P. Glasserman and Y. Wang. Counterexamples in importance sampling for large deviations probabilities. *Ann. Appl. Probab.*, 7(3):731–746, 1997.

[27] E. Gobet and G. Liu. Rare event simulation using reversible shaking transformations. *SIAM J. Sci. Comput.*, 37(5):A2295–A2316, 2015.

[28] A. Guyader, N. Hengartner, and E. Matzner-Løber. Simulation and estimation of extreme quantiles and extreme probabilities. *Applied Mathematics and Optimization*, 64:171–196, 2011.

[29] J.M. Hammersley and D.C. Handscomb. *Monte Carlo methods*. Methuen & Co., Ltd., London; Barnes & Noble, Inc., New York, 1965.

[30] H. Kahn and T.E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Appl. Math. Series*, 12:27–30, 1951.

[31] A. Lagnoux and P. Lezaud. Multilevel branching and splitting algorithm for estimating rare event probabilities. *Simulation Modelling Practice and Theory*, 72:150–167, 2017.

[32] A. Lagnoux-Renaudie. A two-step branching splitting model under cost constraint for rare event analysis. *Journal of Applied Probability*, 46(2):429–452, 2009.

[33] F. Le Gland and N. Oudjane. A sequential particle algorithm that keeps the particle system alive. In *Signal Processing Conference, 2005 13th European*, pages 1–4. IEEE, 2005.

[34] P. L'Ecuyer, V. Demers, and B. Tuffin. Splitting for rare-event simulation. In *Proceedings of the 38th conference on Winter simulation*, pages 137–148. Winter Simulation Conference, 2006.

[35] A. Lee and N. Whiteley. Variance estimation in the particle filter. *Biometrika*, 105(3):609–625, 2018.

[36] T. Lelièvre and G. Stoltz. Partial differential equations and stochastic methods in molecular dynamics. *Acta Numer.*, 25:681–880, 2016.

[37] T. Lestang, F. Ragone, C.-E. Bréhier, C. Herbert, and F. Bouchet. Computing return times or return periods with rare event algorithms. *Journal of Statistical Mechanics: Theory and Experiment*, 2018(4):043213, 2018.

[38] L.J.S. Lopes, C.G. Mayne, C. Chipot, and T. Lelièvre. Adaptive multilevel splitting method: Isomerization of the alanine dipeptide. *arXiv preprint arXiv:1707.00950*, 2017.

[39] H. Louvin. *Development of an adaptive variance reduction technique for Monte Carlo particle transport*. PhD thesis, Université Paris-Saclay, 2017.

[40] H. Louvin, E. Dumonteil, T. Lelièvre, M. Rousset, and C. M. Diop. Adaptive Multilevel Splitting for Monte Carlo particle transport. *EPJ Web Conf.*, 153:06006, 2017.

[41] J. Skilling. Nested sampling for general Bayesian computation. *Bayesian Anal.*, 1(4):833–859, 2006.

[42] J. Skilling. Nested sampling for Bayesian computations. In *Bayesian statistics 8*, Oxford Sci. Publ., pages 491–524. Oxford Univ. Press, Oxford, 2007.

[43] I. Teo, C.G. Mayne, K. Schulten, and T. Lelièvre. Adaptive multilevel splitting method for molecular dynamics calculation of benzamidine-trypsin dissociation time. *Journal of chemical theory and computation*, 12(6):2983–2989, 2016.

[44] E. Ullmann and I. Papaioannou. Multilevel estimation of rare events. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):922–953, 2015.

[45] M. Villén-Altamirano, A. Martinez-Marron, J. Gamo, and F. Fernandez-Cuesta. Enhancement of the accelerated simulation method RESTART by considering multiple thresholds. In Jacques Labetoulle and James W. Roberts, editors, *Fundamental Role of Teletraffic in the Evolution of Telecommunication Networks : Proceedings of the 14th International Teletraffic Congress, Antibes Juan–les–Pins 1994*, pages 787–810. Elsevier, Amsterdam, June 1994.

[46] M. Villén-Altamirano and J. Villén-Altamirano. RESTART : a method for accelerating rare event simulation. In Jacob Willem Cohen and Charles David Pack, editors, *Queueing, Performance and Control in ATM : Proceedings of the 13rd International Teletraffic Congress, Copenhagen 1991*, number 15 in North–Holland Studies in Telecommunications, pages 71–76. North–Holland, Amsterdam, June 1991.