VARIANCE REDUCTION METHODS FOR MONTE CARLO SIMULATION

JAN OLUCHA FUENTES

1. Abstract

Consider the following problem: a radioactive source is stored inside a containment cage that absorbs radiation with a *very high* probability, so that the probability of a radioactive particle escaping the cage is of order 10^{-10}. Suppose we wish to simulate a typical trajectory of a radioactive particle conditioned on exiting the cage, e.g: to understand where radiation escapes the cage. If we approach this naively—by simulating many particles and retaining only those that escape—then on average we would need to simulate on the order of 10^{10} particles to obtain a single "successful" trajectory. This leads to estimators with prohibitively large variance and computational cost.

In this report, we study a general class of problems of this type, where the goal is to estimate properties of Markov processes conditioned on rare events, and we investigate algorithms designed to overcome this difficulty. We survey two well-established methods: the Multilevel Splitting algorithm and its adaptive variant. We also study the Weight Windows (WW) algorithm, a method widely used in nuclear engineering but which still lacks a thorough mathematical analysis. We prove unbiasedness of the WW estimator, derive an explicit expression for its variance, and establish a connection between the WW algorithm and the Doob transform of a Markov process.

Statement of authorship: The contents of Sections 2 and 3 follow mostly the discussions of [3] and Chapters 7 and 9 of [4]. The contents of Section 4 consist primarily of original work of the author, carried out under the supervision of Alex Cox and with valuable input and discussions with Julian Hofstadler. All diagrams have been created using the tikzcd package or with Python. Generative AI (ChatGPT-5, OpenAI, https://chatgpt.com) has been used to aid in the production of the Python-created diagrams, as well as for grammatical and spelling checks.

Contents

Notation

- For a process $X_0, X_1, X_2, \cdots$, we write $X_{0:T}$ to refer to the vector $(X_0, \cdots, X_T)$. Likewise, if $A_0, \cdots, A_T$ is a collection of sets, we denote their Cartesian product $A_0 \times \cdots \times A_T$ by $A_{0:T}$.
- We often write measures in *integral form.* For example, when working with a probability kernel $M : E_1 \times \mathcal{B}(E_2) \to [0, 1]$, we often write expressions like
$$\nu(dy) = M(x, dy),$$
as shorthand to indicate that the measure ν on $(E_2, \mathcal{B}(E_2))$ is defined as
$$\nu(A) = \int_A M(x, dy), \qquad A \in \mathcal{B}(E_2)$$
- $\mathbf{N}, \mathbf{Z}, \mathbf{R}$: the natural numbers (i.e: $\{1, 2, \cdots\}$), the integers and the real numbers respectively.
- For a measurable space $(E, \mathcal{E})$, we write $\mathcal{P}(E)$ for the set of all probability measures on $(E, \mathcal{E})$.
- For a measurable space $(E, \mathcal{E})$, we write $\mathcal{B}_b(E)$ as the set of measurable bounded functions on E.
- N_t: number of particles alive at time t.
- For a measurable set A, we write H_A for the hitting time of A.
- We say that $f(n) \asymp g(n)$ for two quantities that depend on some n if there are constants $c < C$ independent of n such that
$$cf(n) \leq g(n) \leq Cg(n).$$

2. Introduction

2.1. **Setup.** Consider a discrete-time Strong Markov process $X = (X_n)_{n\geq 0}$ taking values in a measurable state space $(E, \mathcal{E})$ that contains a cemetery state ∂. Suppose that $A \in \mathcal{E}$ is a measurable set, which we may refer to as a *target set*. Write H_A for the hitting time of A, assume that $H_A < \infty$ almost surely, and let τ be any other almost surely finite stopping time. This stopping time τ may be interpreted as a random death event, at which point X irreversibly takes the cemetery state ∂. Examples of τ include the hitting time of some other set B, or an arrival time from a Poisson process.

If we define $T = H_A \wedge \tau$, then throughout this report we are concerned with the objects

$$\begin{cases} \mathbf{P}_x(X_T \in A) \\ \mathcal{L}(\{X_n\}_{n\leq T} \mid \{X_T \in A\}) \end{cases} \tag{2.1}$$

That is, we are interested in the probability that, starting from some point x, the process reaches the target set A before the random death event, as well as the law of the trajectories of the Markov process conditioned on reaching A before this event. Examples of this setup include a simple symmetric random walk reaching some large positive value before becoming negative, or planar Brownian motion exiting a large ball before returning to a neighbourhood of its starting position.

We are particularly interested in cases where the probability $p = \mathbf{P}(X_T \in A)$ is *very small*, say of order 10^{-10}, since in such situations naive Monte Carlo schemes fail. Indeed, if we simulate N i.i.d. trajectories $(X_n^i : n \leq T)$ for $i = 1, \cdots, N$ and consider the naive estimator

$$\hat{p} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{X_T^i \in A\}, \tag{2.2}$$

then a straightforward calculation shows that Estimator 2.2 has relative variance

$$\frac{\mathrm{Var}(\hat{p})}{p^2} = \frac{1-p}{Np}.$$

In particular, to keep the relative variance stable as $p \searrow 0$, we require $N \asymp p^{-1}$, which quickly becomes computationally infeasible in many applications.

2.2. **Structure of Report.** In this report, we explore several so-called *variance reduction algorithms*, that is, techniques for estimating 2.1 that yield a lower variance than that obtained using Estimator 2.2. In Sections 3 and 3.5 we review two well-studied algorithms: the Multilevel Splitting algorithm as well as its adaptive version, and discuss some of the theoretical justifications behind these algorithms.

In Section 4 we study the so-called Weight Windows algorithm. We state and prove several results concerning the algorithm. To the best of our knowledge, these results do not explicitly appear in the mathematical literature, at least not in the probabilistic language adopted in this report. Certain properties of the Weight Windows algorithm, such as unbiasedness, are well known and widely used in the applied and engineering communities; however, they are often presented heuristically rather than proved in a fully rigorous probabilistic framework. One of the aims of this report is therefore to provide a precise formulation and proof of these properties.

2.3. **Importance Sampling vs Importance Splitting.** When addressing the rare event problem at hand, one may employ a variety of techniques or algorithms. A particularly relevant class of approaches is given by importance-based methods, which can broadly be classified into two main categories: **Importance Sampling** and **Importance Splitting**. The idea behind the former is as follows. Suppose that the dynamics of our Markov process are governed by a probability measure $\mathbf{P}$, under which the event of interest is rare. One may seek a judiciously chosen probability measure $\mathbf{Q}$, absolutely continuous with respect to $\mathbf{P}$, under which the event of interest is less rare. One can then sample directly N i.i.d. trajectories $X_1, \cdots, X_N$ from $\mathbf{Q}$ instead of $\mathbf{P}$, and estimate quantities of interest via

$$\frac{1}{N}\sum_{i=1}^{N} f(X_i)\frac{d\mathbf{P}}{d\mathbf{Q}}(X_i),$$

where $\frac{d\mathbf{P}}{d\mathbf{Q}}$ denotes the Radon–Nikodym derivative, and is often referred to as the likelihood factor. This estimator is clearly unbiased, and it is straightforward to verify (see for example [5, Proposition 5.6]) that variance reduction is achieved whenever samples drawn from $\mathbf{Q}$ are more likely to lie in regions where $|f|$ is large.

The algorithms studied in this report are not of this type, but instead belong to the family of **Importance Splitting** methods. The general philosophy underlying these methods is the following: trajectories are sampled from the original distribution, and rather than specifying an alternative sampling measure, one introduces a so-called **importance function** $u : E \to (0, \infty)$, which quantifies, in some heuristic sense, “how close” a state is to the rare event of interest. The core idea of splitting algorithms is then to split trajectories that enter regions of high importance, thereby obtaining better coverage of those regions, while cleverly eliminating trajectories that are unlikely to contribute significantly to the quantity being estimated. The first algorithm of this kind that we survey in this report is Multilevel Splitting.

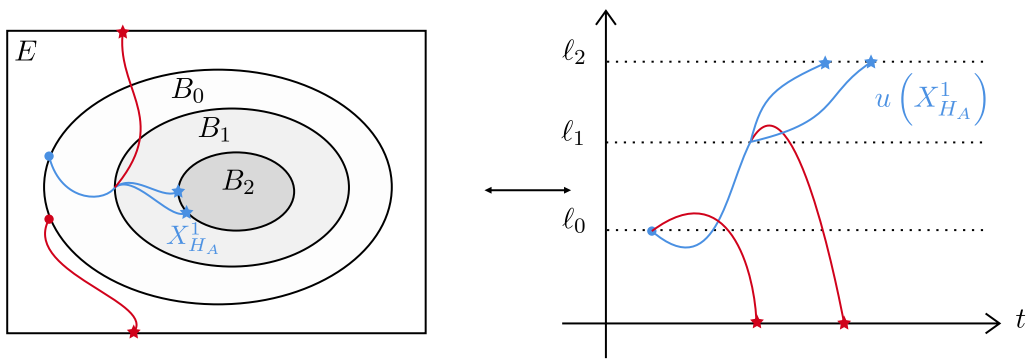

FIGURE 1. Illustration of Multilevel Splitting (Algorithm 1) : the algorithm is initialised with two particles, one of which is killed, and one of which reaches the next level. The surviving particle is split into three new particles, one of which is killed before reaching the final level.

3. Multilevel Splitting

We base the following discussion from the review paper [3].

3.1. **Setup and definition of the Algorithm.** Multilevel Splitting is, in some sense, the most "elementary" class of splitting algorithms one can consider. The idea is as follows: given an importance function $u : E \to (0, \infty)$, we partition the state space E into a decreasing sequence of sets $L_0 \supset L_1 \supset \cdots \supset L_n$, defined by

$$L_j = \{x \in E : u(x) \geq \ell_j\},$$

where $\ell_0 < \ell_1 < \cdots < \ell_n$ are user-specified levels. We assume that trajectories start in the set $L_0 \setminus L_1$, and that the set L_n coincides with the target set A. Recall from the general setup of the Report (see Subsection 2.1) that we are interested in trajectories conditioned on the event $\{H_A < \tau\}$, where τ denotes a terminating time. For clarity of exposition, we shall assume that τ takes the form $\tau = H_B$ for some killing set B.

Once these objects have been specified, the algorithm is as follows (see Figure 1 for an illustration):

Algorithm 1 (General Multilevel Splitting)**.**

1: Initialise $X_0^1, \cdots, X_0^{N_0}$ particles from an initial distribution μ_0.
2: **for** $j = 1, \cdots, n$ **do**
3:     **for** $i = 1, \cdots, N_{j-1}$ **do**
4:         Run trajectory i until next level L_j or final time τ.
5:     **end for**
6:     Discard trajectories that did not reach L_j.
7:     Clone trajectories that did, and denote by N_j the number of resulting trajectories.
8: **end for**

The object of interest produced by this algorithm is the collection of final trajectories $\left(X^i_{[0:H_A \wedge \tau]}\right)$ for $i = 1, \cdots, N_n$, that is, the trajectories of particles that survive until the final level.

Notice that we have kept Line 7 deliberately vague in the definition of Algorithm 1, since there are several possible mechanisms by which one could clone the trajectories. Different cloning mechanisms give rise to different estimators.

Example 3.1 (Fixed splitting)**.** As an example of a cloning mechanism, in Line 7 we could specify: clone each surviving particle into r_j copies, for some "splitting factor" r_j depending on the level. In this fixed scheme, each particle that reaches level ℓ_j produces r_j offspring. Since each of the resulting particles carries a fraction $1/r_j$ of the weight of its parent particle after crossing the j^{th} level, a particle at the final level will carry a weight of $N_0^{-1} \cdot \left(\prod_{j=1}^n r_j^{-1}\right)$. It is then natural to consider estimators of the form

$$\frac{1}{N_0} \cdot \left(\prod_{j=1}^{n} \frac{1}{r_j}\right) \sum_{i=1}^{N_n} f(X^i_{H_A \wedge \tau}), \tag{3.1}$$

to estimate $\mathbf{E}[f(X_{H_A \wedge \tau}) \mid H_A < \tau]$.

The version of Algorithm 1 presented in Example 3.1 is, however, somewhat naive. Poorly chosen splitting ratios may render the algorithm impractical. If the splitting ratios are too small relative to the probability of a trajectory crossing from one level to the next, then too few particles will reach higher levels, leading to a situation similar to naive Monte Carlo. On the other hand, if the splitting ratios are too large, the number of particles grows rapidly, resulting in a high computational cost. For this reason, we will not consider this variant further. Instead, we will discuss the variant of Algorithm 1 which the authors of [3] refer to as the *Sequential Monte Carlo* (SMC) variant. This name is given due to the similarity between the cloning step of this variant of the algorithm, and the resampling step in traditional SMC.

Algorithm 2 (SMC Variant of Algorithm 1)**.**

```
Initialise X_0^1, ..., X_0^N particles from an initial distribution μ_0.
for j = 1, ..., n do
  for i = 1, ..., N_{j-1} do
    Run trajectory i until next level L_j or final time τ.
  end for
  Discard trajectories that did not reach L_j.
  for each discarded trajectory i do
    Choose a surviving trajectory uniformly at random, and replace trajectory
    i by it.
  end for
end for
```

Remark 3.2. An immediate observation from this variant of the algorithm is that the number of particles remains constant and equal to N.

In a similar spirit to Estimator 3.1, the Estimators for this variant of the algorithm will be

$$\hat{p} = \left(\prod_{j=1}^{n} \frac{N_j}{N} \right) \times \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{X^i_{H_A \wedge \tau} \in A\} \tag{3.2}$$

$$\hat{\eta}(f) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X^i_{H_A \wedge \tau}} \tag{3.3}$$

for the probability of the rare event and the law of the particles conditioned on the rare event respectively. In the remaining of this section we will provide the theoretical justification behind this algorithm, however, in the interest of brevity, we will not give the full technical details at each step.

3.2. **A Feynman-Kac interpretation of Multilevel Splitting.** The analysis of Algorithm 2 boils down to noticing that the particle system it builds can be seen as a *McKean interpretation* of a Feynman-Kac model, and then applying the extensive theory from [4]. In this section we show this connection and then state the main results from Feynman-Kac models that enable the study of the algorithm (see [5, Section 2.2] for a quick refresher on some basic concepts).

Recall that we work with a nested sequence of level sets $L_0 \supset L_1 \supset \cdots \supset L_n = A$. Let us define the sequence of stopping times $(T_n)_{n\geq 0}$ by

$$T_n = \inf\{t : X_t \in L_n \cup B\},$$

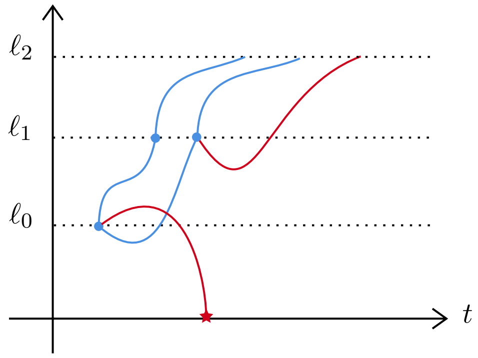

FIGURE 2. Illustration of Algorithm 2: the algorithm is started with 3 particles, one of which (red) is unsuccessful in reaching ℓ_1. This unsuccessful particle chooses one of the surviving particles and its trajectory is cloned.

that is to say, T_n is the first time we reach the n^{th} level set or the killing set. We may now define the process Y_n, given by $Y_n = X_{T_n}$, that is to say, we re-index X by the times it enters successive levels or it is killed.

Reaching the target set A is equivalent to having crossed each intermediate level B_k, and so a natural way of checking if B_k has been crossed is by considering the potentials

$$G_k(x) = \mathbf{1}\{x \in L_k\}.$$

From this it follows that if we let $(\gamma_n)_{n\geq 0}$ and $(\hat{\gamma}_n)_{n\geq 0}$ be the prediction and updated *unnormalised* FK-measures associated to the process $(Y_k)_{k\leq n}$ and the potentials $(G_k)_{k\leq n}$, then

$$\hat{\gamma}_n(f) = \mathbf{E}\left[f(X_{H_A \wedge H_B})\mathbf{1}\{H_A < H_B\}\right],$$

which is precisely the class of expectations we are interested in computing. Similarly, we may define $\eta_n(f) := \gamma_n(f)/\gamma_n(1)$ (resp. $\hat{\eta}_n$), as the prediction (resp. updated) *normalised* FK-measures. Naturally, $\hat{\eta}_n = \mathcal{L}(X_{H_A \wedge H_B} | H_A < H_B)$, and similarly, $\hat{\gamma}_n(1) = \mathbf{P}(H_A < H_B)$. The question then becomes, how do we approximate these measures with particle systems?

A useful result to remember (see [5, Proposition 2.5]) is that the measures $(\hat{\eta}_n)_{n\geq 0}$ satisfy a recursion

$$\hat{\eta}_{n+1} = \psi_{n+1}(\hat{\eta}_n M_{n+1}), \tag{3.4}$$

where ψ_k is an operator on measures defined by $\psi_k(\mu)(dx) := \dfrac{1}{G_k(\mu)} G_k(x)\mu(dx)$. Moreover, from [5, Proposition 2.12], we know that $\psi_n(\eta)$ may be written as $\eta S_{n,\eta}$, where $S_{n,\eta}$ is the non-linear transition kernel

$$S_{n,\eta}(x, dy) = G_n(x)\delta_{\{x\}}(dy) + (1 - G_n(x))\psi_n(\eta)(dy).$$

This transition has a simple interpretation: with probability $G_n(x)$ you remain at position x, and otherwise your position is resampled according to the measure $\psi_n(\eta)$. As such, we can understand the evolution shown in Equation (3.4) in a simple two-step process:

$$\hat{\eta}_n \xrightarrow{\text{mutation}} \eta_{n+1} \xrightarrow{\text{selection}} \hat{\eta}_{n+1} \tag{3.5}$$

Diagram 3.5 gives a natural way to approximate with particles the measures $(\hat{\eta}_n)$. Indeed, suppose at some time n, we have a particle approximation $\hat{\eta}_n^N$ of $\hat{\eta}_n$ given by

$$\hat{\eta}_n^N(f) = \frac{1}{N}\sum_{i=1}^{N} f(\hat{Y}_n^i).$$

We then perform the two step process:

(1) **Mutation:** for each of the N particles at time n, sample a particle $Y_{n+1}^i \sim M_{n+1}(\hat{Y}_n^i, \cdot)$. In our case, this corresponds to evolving a particle that is currently at $\hat{Y}_n^i$ until it reaches level L_{n+1} or the set B. The resulting position is then Y_{n+1}^i.

(2) **Selection:** Once we have mutated each of the particles at time n, we perform a selection step, namely: with probability $G_{n+1}(Y_{n+1}^i)$ the particle at Y_{n+1}^i will remain in its position, and with probability $1 - G_{n+1}(Y_{n+1}^i)$, it will be resampled according to $\psi_{n+1}(\eta_{n+1}^N)$, where

$$\eta_{n+1}^N(f) = \frac{1}{N}\sum_{i=1}^{N} f(Y_{n+1}^i). \tag{3.6}$$

If we go back to our specific case where $G_k(x) = \mathbf{1}\{x \in L_k\}$, we notice that this reduces to: if Y_{n+1}^i has reached level L_{n+1}, it remains at its current position, otherwise, it is resampled from

$$\psi_{n+1}(\eta_{n+1}^N)(dx) = \frac{N}{\sum_{i=1}^N G_{n+1}(Y_{n+1}^i)} \cdot \frac{1}{N}\sum_{i=1}^{N} G_{n+1}(x)\delta_{Y_{n+1}^i}(dx) = \frac{1}{\# I_{n+1}} \sum_{y \in I_{n+1}} \delta_y(dx), \tag{3.7}$$

where I_{n+1} denotes the particles that have reached level L_{n+1} successfully.

At this point we can see the clear correspondence between Algorithm 2 and this Feynman-Kac formulation. Indeed: Equation (3.7) says that to sample from $\psi_{n+1}(\eta_{n+1}^N)(dx)$ one simply samples a surviving particle uniformly at random, which is the same action taken by Algorithm 2 in its resampling step. An important observation however, is that this Algorithm is only well-defined up to the following stopping time:

$$\tau^N := \inf\{t : Y_t^i \in B \ \text{ for all } \ i = 1, \cdots, N\},$$

that is to say, the first time at which all particles are killed. Indeed, if all particles fail to reach a given level, then one cannot perform the resampling step. We refer to this stopping time as the **extinction time**. As has been explained, we will use $\hat{\eta}_n^N(f)$ as an estimate for $\hat{\eta}_n(f)$. The particle model can also be used to give estimates to the remaining quantities of interest, i.e: $\eta_n(f), \gamma_n(f)$, and $\hat{\gamma}_n(f)$. The best way to introduce these estimates is by noticing that from the definition of Feynman-Kac measures, we have

$$\eta_{n-1}(G_n) = \frac{\gamma_{n-1}(G_n)}{\gamma_{n-1}(1)} = \frac{\gamma_n(1)}{\gamma_{n-1}(1)},$$

and so $\gamma_n(1) = \gamma_{n-1}(1)\eta_{n-1}(G_n)$, and a quick calculation with our specific choice of potentials shows

$$\gamma_n(1) = \prod_{p=1}^{n} \eta_{p-1}(G_p).$$

As a result, our particle approximation to $\gamma_n(1)$, which we write as $\gamma_n^N(1)$, is given by $\gamma_n^N(1) = \prod_{j=1}^n \eta_{p-1}^N(G_p) = \dfrac{1}{N^n}\prod_{p=1}^n \# I_p$, where recall, $\# I_p$ is the number of particles that reached level p, and η_n^N is defined as in (3.6). Finally, recalling that $\hat{\gamma}_n(f) = \gamma_n(f \cdot G_n) = \gamma_n(1)\eta_n(f \cdot G_n)$, we write

$$\hat{\gamma}_n^N(1) = \gamma_n^N(1)\eta_n^N(G_n) \tag{3.8}$$

$$= \left(\frac{1}{N^n}\prod_{p=1}^{n} \#I_p\right) \cdot \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{Y_n^i \in A\} \tag{3.9}$$

And Equation (3.9) will be our estimator for $\mathbf{P}(H_A < H_B)$, which coincides with the estimator we claimed hinted in (3.2).

3.3. **Theoretical results of Algorithm 2.** In this section we detail some of the theoretical results regarding Algorithm 2. In particular, we will discuss the unbiasedness of estimators $\gamma_t^N(f)$ as well as a Central Limit Theorem.

The first result of interest is regarding the extinction time τ^N. Should all particles fail to reach some level L_k so that the population goes extinct before reaching the final level, we would estimate $\mathbf{P}(H_A < H_B)$ to be zero, like we would do in the naive Monte Carlo case. Fortunately, the probability of this occuring decays exponentially quickly in the population size. In particular, we have the following result ([4, Theorem 7.4.1]), whose proof we ommit.

Theorem 3.3 (Extinction time)**.** For $n \geq 0$ and $N \geq 1$, there are constants $a(n), b(n) > 0$ such that:

$$\mathbf{P}(\tau^N \leq n) \leq a(n)\exp(-N/b(n))$$

Now we turn to unbiasedness, that is to say, we want to show $\mathbf{E}[\gamma_n^N(f)\mathbf{1}\{\tau^N \geq n\}] = \gamma_n(f)$. The key idea will be to show that the following quantity:

$$\gamma_n^N(f)\mathbf{1}\{\tau^N \geq n\} - \gamma_n(f),$$

is the terminal value of a Martingale zero initial expectation. We provide the proof in order to provide a parallelism with a later proof in Section 4 which follows almost the exact same structure. Let's quickly recall the definition of the semigroup Q. Given the Feynman-Kac model associated to the potentials G and transitions M, we define

$$Q_n(x_{n-1}, dx_n) = G_{n-1}(x_n)M_n(x_{n-1}, dx_n),$$

so that $\gamma_n = \gamma_{n-1}Q_n$. Then we denote $Q_{p,n} = Q_{p+1}\cdots Q_n$. The result is as follows ([4, Proposition 7.4.1])

Proposition 3.4 (Martingale Decomposition)**.** Fix some $n \in \mathbf{N}$. Then for a test function f, the process $(\Gamma_t^N(f))_{t\leq n}$ defined by

$$\Gamma_t^N(f) = \gamma_t^N\left(Q_{t,n}f\right)\mathbf{1}\{\tau^N \geq t\} - \gamma_t\left(Q_{t,n}f\right),$$

is a Martingale with respect to the filtration $(\mathcal{F}_s)$, where $\mathcal{F}_s$ is the sigma algebra generated by all particles up to time s.

Proof. The key observation is that

$$\gamma_t^N(f)\mathbf{1}\{\tau^N \geq t\} - \gamma_t(f) = \sum_{s=0}^{t} \gamma_s^N(Q_{s,t}f)\mathbf{1}\{\tau^N \geq s\} - \gamma_{s-1}^N(Q_{s-1,t}f)\mathbf{1}\{\tau^N \geq s-1\} \tag{3.10}$$

provided we make take the convention $\gamma_{-1}^N(Q_{-1}, tf) = \gamma_t(f)$. Next, we notice that by the particle system definitions, we have

$$\gamma_s^N(Q_{s,t}f)\mathbf{1}\{\tau^N \geq s\} := \gamma_s^N(1)\mathbf{1}\{\tau^N \geq s\}\eta_s^N(Q_{s,t}f) \tag{3.11}$$

and similarly,

$$\gamma_{s-1}^N(Q_{s-1,t}f)\mathbf{1}\{\tau^N \geq s-1\} = \gamma_{s-1}^N(1)\mathbf{1}\{\tau^N \geq s-1\}\eta_{s-1}^N(G_{s-1}M_s(Q_{s,t}f)) \tag{3.12}$$

Now we notice the following facts:

(1) $1 = \mathbf{1}\{\eta_{s-1}^N(G_{s-1}) = 0\} + \mathbf{1}\{\eta_{s-1}^N(G_{s-1}) > 0\}$.

(2) $\eta_{s-1}^N(G_{s-1}M_s(Q_{s,t}f))\mathbf{1}\{\eta_{s-1}^N(G_{s-1}) = 0\} = 0$.

Combining Facts (1) and (2) into Equation (3.12) gives

$$\gamma_{s-1}^N(1)\mathbf{1}\{\tau^N \geq s-1\}\eta_{s-1}^N(G_{s-1}M_s(Q_{s,t}f)) \tag{3.13}$$

$$= \gamma_{s-1}^N(1)\mathbf{1}\{\tau^N \geq s-1\}\eta_{s-1}^N(G_{s-1}M_s(Q_{s,t}f))(\mathbf{1}\{\eta_{s-1}^N(G_{s-1}) = 0\} + \mathbf{1}\{\eta_{s-1}^N(G_{s-1}) > 0\}) \tag{3.14}$$

$$= \gamma_{s-1}^N(1)\mathbf{1}\{\tau^N \geq s-1\}\eta_{s-1}^N(G_{s-1}M_s(Q_{s,t}f))\mathbf{1}\{\eta_{s-1}^N(G_{s-1}) > 0\} \tag{3.15}$$

$$= \gamma_{s-1}^N(1)\mathbf{1}\{\tau^N \geq s\}\eta_{s-1}^N(G_{s-1}M_s(Q_{s,t}f)) \tag{3.16}$$

where the last equality is due to the fact that $\{\tau^N \geq s\} = \{\tau^N \geq s-1\} \cap \{\eta_{s-1}^N(G_{s-1}) > 0\}$. Indeed, this last event means that there were some particles that reached level $s-1$, and so extinction can occur at least at level s.

Notice now that

$$\gamma_{s-1}^N(1)\eta_{s-1}^N(G_{s-1}M_s(Q_{s,t}f)) = \frac{\gamma_s^N(1)}{\eta_{s-1}(G_s)}\eta_{s-1}^N(Q_s(Q_{s,t}f)) = \gamma_s^N(1)\eta_{s-1}^N S_{n,\eta_{s-1}^N} M_s$$

Combining this with Equation (3.16) and Equation (3.11) into the decomposition of Equation (3.10) with the choice of function being $Q_{t,n}f$ gives:

$$\Gamma_t^N(f) = \sum_{q=0}^{t} \gamma_q(1)\mathbf{1}\{\tau^N \geq q\}\left(\eta_q^N\left(Q_{q,n}f\right) - \eta_{q-1}^N K_{q,\eta_{q-1}^N}\left(Q_{q,n}f\right)\right), \tag{3.17}$$

where $K_{n,\eta_{n-1}^N} := S_{n,\eta_{n-1}^N} M_n$. Finally, it is easy to verify that $\mathbf{1}\{\tau^N \geq q\}$ is $\mathcal{F}_{q-1}$-measurable, and from the construction of the particle system, it is also easy to verify that $\mathbf{E}[\eta_q^N\left(Q_{q,n}f\right)|\mathcal{F}_{q-1}] = \eta_{q-1}^N K_{q,\eta_{q-1}^N}\left(Q_{q,n}f\right)$. □

3.4. **Central Limit Theorem.** Thanks to the connection between Algorithm 2 and the framework of Feynman-Kac measures, we can also provide a Central Limit Theorem for the estimators constructed from Algorithm 2. The central result is [4, Proposition 9.4.1], in which a general Central Limit Theorem is proved for particle approximations of the models $(\eta_n)_{n\geq 0}$ satisfying a recursion of the type $\eta_n = \eta_{n-1}K_{n,\eta_{n-1}}$. However, when using the specific shape of Algorithm 2, the CLT becomes as follows [3, Theorem 3.1]

Theorem 3.5 (Central Limit Theorem)**.** Let $\hat{\gamma}_n^N(1)$, as before, denote the estimator of the rare event probability in Algorithm 2, and let τ^N be the extinction time. Then

$$\mathbf{1}\{\tau^N > n\}\sqrt{N}\left(\hat{\gamma}_n^N(1) - \mathbf{P}(H_A < H_B)\right) \Rightarrow \mathcal{N}(0, \sigma^2),$$

where

$$\sigma^2 = \sum_{j=1}^{n-1} \theta_j (p_{j-1}^2 - p_j^2) \mathrm{Var}_{\hat{\eta}_j}(q^*) + p^2 \sum_{j=1}^{L} \left(\frac{1}{\theta_j} - 1\right), \tag{3.18}$$

and where θ_j denotes the probability of reaching level j given that level $j-1$ has been reached, p_j is the probability of reaching level j, and $q^*(x) = \mathbf{P}_x(H_A < H_B)$.

Remark 3.6 (On the choice of importance function)**.** With expression (3.18) in mind, we may ask what constitutes an optimal choice of importance function u for defining the levels $L_j = \{x : u(x) \geq \ell_j\}$.

Without loss of generality, we may assume that $\ell_n = 1$. If we choose $u(x) = \mathbf{P}_x(H_A < H_B)$, then, by definition, $\hat{\eta}_j$ corresponds to the distribution of particles that have just reached level j before being killed. Moreover, since level L_j consists of points for which particles have probability at least ℓ_j of success, it follows that $\mathrm{Var}_{\hat{\eta}_j}(q^*) = 0$. In this case, the variance reduces to

$$\sigma^2 = p^2 \sum_{j=1}^{L} \left(\frac{1}{\theta_j} - 1\right).$$

Of course, this choice of optimal importance function $u(x)$ is only of theoretical interest, since if $u(x)$ were known explicitly, there would be no need to solve the rare event problem.

3.5. **Adaptive Multilevel Splitting.** To conclude the chapter on Multilevel Splitting, we briefly describe the adaptive variant of Algorithm 2. By adaptive, we mean that the choice of levels $\ell_1 < \ell_2 < \cdots < \ell_n$ is not fixed in advance by the user, but is instead determined dynamically as the algorithm evolves.

To define this algorithm, we assume that the target set A is given by $A = \{x : u(x) > 1\}$, and that the Markov process $(X_t)_{t \geq 0}$ evolves in continuous time with continuous trajectories. Define the entrance times $H_t := \inf\{s : u(X_s) > t\}$, so that we are interested in estimating $\mathbf{P}_x(H_1 < H_B)$, where B is the killing set.

In informal terms, the algorithm proceeds as follows. First, simulate all particles until they either reach the set A or are killed upon hitting the set B. Among all particles, identify the one whose maximal u-value is the smallest, and set ℓ_1 equal to this value. From the perspective of Algorithm 2, all other particles have successfully crossed level ℓ_1. The unsuccessful particle is then resampled uniformly from the successful trajectories up to level ℓ_1, and the procedure is repeated; see Figure 3. We state the algorithm formally below

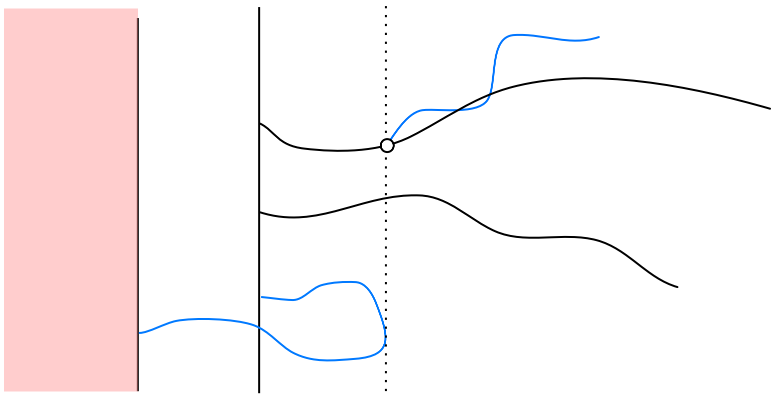

FIGURE 3. AMS Algorithm

Algorithm 3 (AMS)**.** Adaptive Multilevel Splitting is given by the following algorithm:

Require: A Strong Markov Process $(X_t)_{t\geq 0}$, importance function u, an initial distribution μ_0.

1: Initialise by sampling $X_0^{i,0} \overset{\text{iid}}{\sim} \mu_0$ for $i = 1, \cdots, N$.

2: Simulate trajectories $\left\{X_t^{i,0} : t \leq H_1^{i,0} \wedge H_B^{i,0}\right\}$ for $i = 1, \cdots, N$.

3: **for** $j \geq 1$ **do**

4: For each particle $i = 1, \cdots, N$, compute its maximum height:

$$\sup_{t\leq H_1^{i,j-1}\wedge H_B^{i,j-1}} u(X_t^{i,j-1}).$$

Let N_j be the index of the particle with the smallest score, and ℓ_j be the maximum height achieved by this particle. If $\ell_j = 1$ we stop the algorithm, since all particles reached the target set.

5: For all "surviving particles", i.e: for all $i \neq N_j$, leave trajectory as is. That is to say, set

$$\{X_s^{i,j} : s \geq 0\} = \{X_{s\wedge H_1\wedge H_B}^{i,j-1} : s \geq 0\}$$

6: For the lowest scoring particle N_j, select an index M_j uniformly at random in $\{1, \cdots, N\} \setminus \{N_j\}$ and replace the trajectory of $X^{N_j,j}$ in the following way:

(1) Let $\sigma_j = \inf\{s : u(X_s^{M_j,j}) > \ell_j\}$

(2) For $s \leq \sigma_j$, replace $X_s^{N_j,j} = X_s^{M_j,j}$.

(3) For $s > \sigma_j$, simulate a new piece of trajectory $(X_s^{N_j,j})_{s\geq\sigma_j}$ as an independent trajectory according to the law of X conditioned on starting at $X_{\sigma_j}^{M^j,j}$.

7: **end for**

While we will not delve any deeper into the technical details of this adaptive version, it is worth noting that the unnormalised measures $\hat{\gamma}_k$ produced by the particles of Algorithm 3 are also unbiased (see [3, Section 3.2]) and by drawing a connection with Fleming-Viot processes, one can also study a Central Limit Theorem, analogous to that of Theorem 3.5 (see [3, Theorem 3.3]). We now move on to the next and final chapter of this report.

4. Weight Windows

The Weight Windows technique is a method for variance reduction in Monte Carlo simulation that originates in the context of particle transport, in particular in the development of the Monte Carlo N-Particle Transport Code (MCNP) code at Los Alamos Laboratory in the late 1970s and early 1980s, in connection with deep penetration shielding problems. The earliest mentions of Weight Windows we have been able to find are in the technical reports from Los Alamos Laboratory [6] and [8]. In such problems, particles must travel through large amounts of absorbing material before reaching a region of interest, making naive Monte Carlo simulation extremely inefficient.

To address this issue, importance sampling was used to bias trajectories so that particles were more likely to travel long distances before being absorbed. However, as noted by Booth in his historical account of the method [2], this choice of biasing led to the phenomenon that particles located in similar regions of space could carry vastly different likelihood factors. Booth observed that this behaviour was computationally inefficient, and argued that particles in similar regions of space should instead have comparable weights. This observation provided the key motivation for the development of the Weight Windows algorithm, which we now formulate in an abstract setting:

Let M be a Markov kernel on $(E, \mathcal{E})$. Let $\underline{h} \leq h \leq \bar{h}$ be three positive functions on E. The function h is called the **target function**, and the difference $\bar{h} - \underline{h}$ is called the **window-width**. Finally, for any probability measure $\mu_0 \in \mathcal{P}(E)$, define tilted probability measure μ_0^h by $\mu_0^h(dx) \propto \frac{1}{h(x)}\mu_0(dx)$.

The Weight Windows algorithm defines a particle system in which at time t we have N_t weighted particles $\{(X_t^i, W_t^i) : i = 1, \cdots, N_t\}$. We denote by X_t^i and W_t^i the position and weight of some particle $1 \leq i \leq N_t$ at time t. The algorithm is defined as follows:

Algorithm 4 (Weight Windows)**.**

Sample $X_0^i \sim \mu_0^h$, and set initial weights $W_0^i = h\left(X_0^i\right)$ for for $i = 1, \cdots, N_0$.
for $n = 1, 2, \cdots$ **do**
  for $i = 1, \cdots, N_{n-1}$ **do**
    Sample $\widehat{X}_{n+1}^i \sim M(X_n, \cdot)$ and set $\widehat{W}_{n+1}^i = W_n^i$
    if $\widehat{W}_{n+1}^i \geq \bar{h}(\widehat{X}_{n+1}^i)$ **then**
      Split particle i into $\left\lfloor \frac{\widehat{W}_{n+1}^i}{h(\widehat{X}_{n+1}^i)} \right\rfloor + Y$ offspring, where

$$Y \sim \text{Bernoulli}\left(\frac{\widehat{W}_{n+1}^i}{h(\widehat{X}_{n+1}^i)} - \left\lfloor \frac{\widehat{W}_{n+1}^i}{h(\widehat{X}_{n+1}^i)} \right\rfloor\right).$$

      Each offspring with weight $h(\widehat{X}_{n+1}^i)$, and position $\widehat{X}_{n+1}^i$.
    else if $\widehat{W}_{n+1}^i \leq \bar{h}(\widehat{X}_{n+1}^i)$ **then**
      Keep particle with probability $\frac{\widehat{W}_{n+1}^i}{h(\widehat{X}_{n+1}^i)}$, and if kept, set $W_{n+1}^i = h(\widehat{X}_{n+1}^i)$
    else
      Set $W_{n+1}^i = \widehat{W}_{n+1}^i$, and $X_{n+1}^i = \widehat{X}_{n+1}^i$.
    end if
  end for
end for

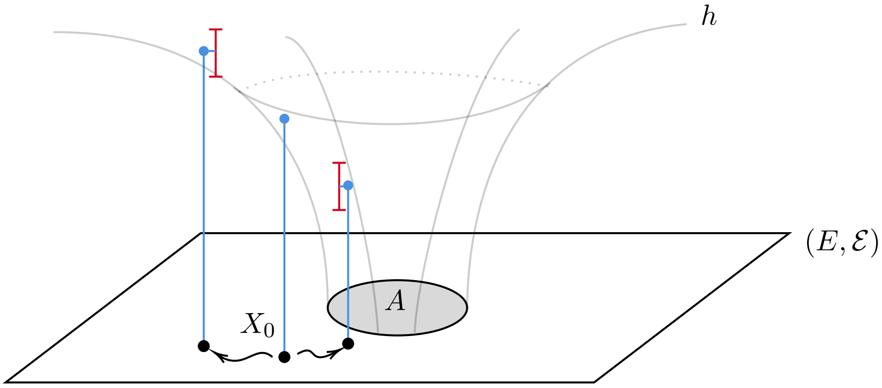

FIGURE 4. Illustration of weight windows

Important Idea ☕. The idea behind the algorithm is shown in Figure 4: imagine a particle starts at location X_0. This particle has a weight equal to its current h-value, represented by the vertical height of the blue line above X_0 in the diagram.

Suppose our target is to reach the set A, and we use a function h represented as in the diagram. Then our particle takes a step and takes position X_1. Suppose the step taken was towards the point closer to A. The red region indicates the window $\underline{h}(X_1) < h(X_1) < \bar{h}(X_1)$. In this case, since the height of the blue line above X_0 is higher than the upper bound of the red region, the particle carries too much weight and so it is split into several particles. In this way, we have achieved more particles in the informative region.

On the other hand, suppose that the particle at X_0 decides to step away from A, as indicated by the left point in the figure. In this case the height of the blue line above X_0 is below the red region, and so we must carry out a Russian Roulette step, potentially removing this less informative particle.

The object of primary interest to us is the measure-valued process $(\widehat{\mu}_n)_{n\geq 0}$ defined by

$$\widehat{\mu}_n = \sum_{i=1}^{N_n} W_n^i \delta_{X_n^i}, \tag{4.1}$$

which we will use to build unbiased estimators for $\mathbf{E}[f(X_t)]$.

Remark 4.1. It is not true that the object in Equation 4.1 is a probability measure, even if $W_0 = 1$. Indeed: consider the case where we start with one single particle and $W_0 = 1$. Suppose that after one transition step, the particle transitions into an area where it is underweight. With positive probability it will survive the Russian Roulette step and have weight $W_1 > W_0$.

4.1. **Unbiasedness and a Martingale Property.** If we carefully inspect Line 6 and Line 8 in Algorithm 4, it appears that the total weight of the system is conserved in expectation. It is therefore reasonable to expect that using $\hat{\mu}_n(f)$, we can construct estimators that are unbiased. We will confirm in the next few results that this is indeed the case. We now discuss the central idea underlying the upcoming proofs and introduce some notation.

A key idea in both the results of this section and those in the forthcoming sections is that the information available at each timestep t *evolves in a two-step fashion.* To see this, consider a single particle X_{t-1} at time $t-1$. First, X_{t-1} undergoes an exploration step, i.e., it moves according to the Markovian dynamics prescribed by M. Throughout this chapter, we use the notation $\widehat{X}_t$ to denote the position of particle X_{t-1} after one exploration step.

Next, the particle at its new position $\widehat{X}_t$ undergoes either a Russian Roulette, Splitting, or "do nothing" step (which we collectively refer to as a correction step). During this step, it produces a number M_t of offspring, where $M_t \in \mathbf{N} \cup \{0\}$. Note that, in this terminology, having 1 offspring means that the particle remains as a single particle, while having 0 offspring means that the particle is killed. In what follows, we use the notation M_t^i to denote the number of offspring produced by particle i after moving from its position at time $t-1$ to its position at time t.

We can therefore represent a single step of the algorithm for one particle by the following diagram:

$$(X_{t-1}, W_{t-1}) \xrightarrow{\text{exploration}} (\widehat{X}_t, \widehat{W}_t) \xrightarrow{\text{correction}} \left\{ \left(X_t^i, W_t^i \right) : i = 1, \cdots, M_t \right\} \tag{4.2}$$

Note for future reference that we assume $\widehat{W}_t$ is $\mathcal{F}_{t-1}$ measurable. If we now consider the entire particle system, an analogous structure arises in terms of filtrations. Let $(\mathcal{F}_t)_{t \geq 0}$ denote the filtration generated by the particles and their weights up to time t. We then define an *intermediate filtration*

$$\widehat{\mathcal{F}_t} = \sigma \left(\mathcal{F}_{t-1}, \{ \widehat{X}_t^i : i = 1, \cdots, N_{t-1} \} \right), \tag{4.3}$$

that is, the filtration containing all information up to time $t-1$, together with the locations *after exploration* of each particle alive at time $t-1$. In particular, this filtration does not reveal the number of offspring produced by each particle after reaching its new location. The corresponding diagram for the filtrations is then

$$\mathcal{F}_{t-1} \xrightarrow{\text{exploration}} \widehat{\mathcal{F}_t} \xrightarrow{\text{correction}} \mathcal{F}_t$$

Note that in particular $\mathcal{F}_{t-1} \subset \widehat{\mathcal{F}}_t \subset \mathcal{F}_t$. We are now ready to state and prove the main results of this section:

Lemma 4.2. Let $(\hat{\mu}_n)_{n\geq 0}$ be the measure-valued process described above. Let $\mathcal{F}_n$ be the filtration generated up to time n by the particles and the weights. Then, for all $f \in \mathcal{B}_b(E)$ we have that

$$\mathbf{E}[\hat{\mu}_n(f) \mid \mathcal{F}_{n-1}] = \hat{\mu}_{n-1}(Mf)$$

Proof. Let us focus on a single particle (X_n, W_n) at time n, and recall that the particle evolves via a two-step procedure as described in Diagram 4.2. We will show that

$$\mathbf{E}\left[\sum_{i=1}^{M_{n+1}} f\left(X_{n+1}^i\right) W_{n+1}^i \middle| \widehat{\mathcal{F}}_{n+1}\right] = \widehat{W}_{n+1} f(\widehat{X}_{n+1}),$$

and since offspring inherit the parent's position, it is enough to prove that

$$\mathbf{E}\left[\sum_{i=1}^{M_{n+1}} W_{n+1}^i \middle| \widehat{\mathcal{F}}_{n+1}\right] = \widehat{W}_{n+1}.$$

We now examine the left-hand side. Noting that all summands are identical, we may write, in view of Algorithm 4,

$$\begin{aligned}
\mathbf{E}\left[M_{n+1} W_{n+1}^i \middle| \widehat{\mathcal{F}}_{n+1}\right] &= \mathbf{E}\left[M_{n+1} W_{n+1}^i \middle| \widehat{W}_{n+1}, \widehat{X}_{n+1}\right] \\
&= \mathbf{E}\left[M_{n+1} W_{n+1}^i \left(\mathbf{1}_{\{\mathrm{RR}\}} + \mathbf{1}_{\{\mathrm{Split}\}} + \mathbf{1}_{\{\mathrm{Pass}\}}\right) \middle| \widehat{W}_{n+1}, \widehat{X}_{n+1}\right] \\
&= \mathbf{1}_{\{\mathrm{RR}\}} \left(0 \cdot \left(1 - \frac{\widehat{W}_{n+1}}{h(\widehat{X}_{n+1})}\right) + 1 \cdot h(\widehat{X}_{n+1}) \cdot \left(\frac{\widehat{W}_{n+1}}{h(\widehat{X}_{n+1})}\right)\right) \\
&\quad + \mathbf{1}_{\{\mathrm{Split}\}} \left(h(\widehat{X}_{n+1}) \cdot \frac{\widehat{W}_{n+1}}{h(\widehat{X}_{n+1})}\right) \\
&\quad + \mathbf{1}_{\{\mathrm{Pass}\}} \left(\widehat{W}_{n+1}\right) \\
&= \widehat{W}_{n+1} \left(\mathbf{1}_{\{\mathrm{RR}\}} + \mathbf{1}_{\{\mathrm{Split}\}} + \mathbf{1}_{\{\mathrm{Pass}\}}\right) = \widehat{W}_{n+1}.
\end{aligned}$$

We now combine this with the exploration step and the Markov property to obtain

$$\begin{aligned}
\mathbf{E}\left[\sum_{i=1}^{M_{n+1}} f\left(X_{n+1}^i\right) W_{n+1}^i \middle| \mathcal{F}_n\right] &= \mathbf{E}\left[\mathbf{E}\left[\sum_{i=1}^{M_{n+1}} f\left(X_{n+1}^i\right) W_{n+1}^i \middle| \widehat{\mathcal{F}}_{n+1}\right] \middle| \mathcal{F}_n\right] \\
&= \mathbf{E}\left[\widehat{W}_{n+1} f(\widehat{X}_{n+1}) \middle| \mathcal{F}_n\right] \\
&= \widehat{W}_{n+1} (Mf)(X_n).
\end{aligned}$$

Here, the final equality follows from the Markov property, together with the fact that $\widehat{W}_n$ is $\mathcal{F}_n$-measurable.

Note that, as Algorithm 4 is currently formulated, we have $\widehat{W}_n = W_n$. Having established the result for a single particle, the general claim follows by linearity of conditional expectation, summing over all particles and their corresponding correction events at time $n+1$. □

In light of this Lemma, we can now state and prove the following Proposition:

Proposition 4.3. Let $f \in \mathcal{B}_b(E)$. Then for a fixed $n \in \mathbf{N}$, the process $(\Gamma_t^n(f))_{t \le n}$ defined by

$$\Gamma_t^n(f) := \hat{\mu}_t(M^{n-t}f) - \hat{\mu}_0(M^n f)$$

is an $\mathcal{F}_t$-Martingale.

Proof. Lemma 4.2 establishes that, for any $f_i \in \mathcal{B}_b(E)$, we have

$$\mathbf{E}\left[\hat{\mu}_i(f_i) - \hat{\mu}_{i-1}(Mf_i) | \mathcal{F}_{i-1}\right] = 0.$$

For n and M as in the statement of the proposition, define $f_i = M^{n-i}f$, which is easily seen to belong to $\mathcal{B}_b(E)$. Finally, by a telescoping argument, we obtain

$$\Gamma_t^n(f) = \sum_{i=1}^{n} \left(\hat{\mu}_i(f_i) - \hat{\mu}_{i-1}(Mf_i)\right).$$

The martingale property then follows immediately. □

Corollary 4.4 (Unbiasedness Property)**.** For a function $f \in \mathcal{B}_b(E)$, define

$$\hat{\nu}_t(f) \equiv \hat{\nu}_{t,N_0}(f) := \frac{\mu_0(1/h)}{N_0}\hat{\mu}_t(f). \tag{4.4}$$

Then $\hat{\nu}_t(f)$ is an unbiased estimator for $\mathbf{E}_{\mu_0}[f(X_t)]$

Proof. Since the Martingale increments have mean zero,

$$\mathbf{E}[\hat{\mu}_t(f)] = \mathbf{E}[\hat{\mu}_0(M^t f)].$$

Now, if the initial particles are sampled from the inverse-tilted law

$$\mu_0^h(dx) = \frac{h(x)^{-1}}{\mu_0(1/h)}\,\mu_0(dx),$$

then

$$\mathbf{E}\left[\hat{\mu}_0(M^t f)\right] = \sum_{i=1}^{N_0} \mathbf{E}\big[W_0^i (M^t f)(X_0^i)\big] \tag{4.5}$$

$$= N_0 \cdot \mathbf{E}_{\mu_0^h}\big[h(X_0)(M^t f)(X_0)\big] \tag{4.6}$$

$$= \frac{N_0}{\mu_0(1/h)} \cdot \mathbf{E}_{\mu_0}\big[(M^t f)(X_0)\big] \tag{4.7}$$

$$= \frac{N_0}{\mu_0(1/h)} \cdot \mathbf{E}_{\mu_0}[f(X_t)], \tag{4.8}$$

as required. □

4.1.1. *A modification of the algorithm to include killing.* So far, we have presented the Weight Windows algorithm as a means of generating an “artificial push” at the population level towards a region of high interest; see Figure 4. However, in many applications one is interested in Markov chains with killing. That is, there is a subset $E^\partial \subset E$ such that, upon entering E^∂, particles are immediately killed. For instance, one may be interested in computing

$$\mathbf{E}[f(X_t)\mathbf{1}\{H_\partial > t\}],$$

where H_∂ denotes the hitting time of E^∂.

Algorithm 4 admits a simple modification. The evolution of a single particle is now described by
(4.9)

$$(X_{t-1}, W_{t-1}) \xrightarrow{\text{exploration}} (\widehat{X}_t, \widehat{W}_t) \xrightarrow{\text{correction/killing}} \begin{cases} \{(X_t^j, W_t^j) : j = 1, \dots, M_t\}, & \widehat{X}_t \in E \setminus E^\partial, \\ \dagger, & \widehat{X}_t \in E^\partial. \end{cases}$$

Here $\dagger$ means that the particle is killed. Equivalently, the particle is removed from the computation entirely, or one may adjoin a cemetery state $\dagger$ and extend every test function by setting $f(\dagger) = 0$.

At first sight, it is not immediate how Corollary 4.4 changes under this modification, since mass may now be lost through killing. However, a simple calculation shows that the modified algorithm remains unbiased for the killed dynamics.

Proposition 4.5. Let K be the killed kernel, defined by

$$K(x, dy) = M(x, dy)\mathbf{1}\{y \in E \setminus E^\partial\}.$$

Then, under the modification shown in (4.9),

$$\mathbf{E}[\hat{\nu}_t(f)] = \mu_0(K^t f) = \mathbf{E}[f(X_t)\mathbf{1}\{H_\partial > t\}]\,.$$

Proof. It suffices to prove the analogue of Lemma 4.2. In the standard version of the algorithm, one has

$$\mathbf{E}[\hat{\mu}_t(f) \mid \mathcal{F}_{t-1}] = \hat{\mu}_{t-1}(Mf).$$

We now show that, under the modification (4.9),

$$\mathbf{E}[\hat{\mu}_t(f) \mid \mathcal{F}_{t-1}] = \hat{\mu}_{t-1}(Kf).$$

Consider a single parent particle (X_t, W_t) at time t. After the exploration step, let the proposed position be $\widehat{X}_{t+1}$ and set $\widehat{W}_{t+1} = W_t$. If $\widehat{X}_{t+1} \in E^\partial$, the particle is killed and produces no offspring. If $\widehat{X}_{t+1} \in E \setminus E^\partial$, it produces M_t offspring and the usual Weight Windows correction is applied.

Conditioning first on the intermediate filtration $\widehat{\mathcal{F}}_{t+1}$, the usual unbiasedness calculation for the correction step gives

$$\mathbf{E}\left[\sum_{i=1}^{M_t} W_{t+1}^i f(X_{t+1}^i) \middle| \widehat{\mathcal{F}}_{t+1}\right] = \mathbf{1}\{\widehat{X}_{t+1} \in E \setminus E^\partial\}\widehat{W}_{t+1} f(\widehat{X}_{t+1}).$$

Taking conditional expectation with respect to $\mathcal{F}_t$ gives

$$\begin{aligned}\mathbf{E}\left[\sum_{i=1}^{M_t} W_{t+1}^i f(X_{t+1}^i) \middle| \mathcal{F}_t\right] &= W_t \int_E M(X_t, dy)\mathbf{1}\{y \in E \setminus E^\partial\} f(y) \\ &= W_t (Kf)(X_t).\end{aligned}$$

Summing over all parent particles alive at time t and using linearity of conditional expectation yields

$$\mathbf{E}[\hat{\mu}_{t+1}(f) \mid \mathcal{F}_t] = \hat{\mu}_t(Kf).$$

As required. □

4.2. **Some convergence results.** With Corollary 4.4 at hand, we can immediately derive some convergence results as $N_0 \to \infty$. The key observation is that the particle system can be decomposed into N_0 independent genealogies, each of which acts as a version of the algorithm with one single initial particle.

Theorem 4.6. Let $\hat{\nu}_{t,N_0}(f)$ be defined as in Equation (4.4). Then for any t:

(1) $\hat{\nu}_{t,N_0}(f) \to \mu_0(M^t f)$ almost surely as $N_0 \to \infty$.
(2) As $N_0 \to \infty$,
$$\sqrt{N_0}\left(\hat{\nu}_{t,N_0}(f) - \mu_0(M^t f)\right) \Rightarrow \mathcal{N}(0, \sigma_t^2(f)),$$
where
$$\sigma_t^2(f) = \mu_0(1/h)^2 \mathrm{Var}\left(\hat{\mu}_t^{(1)}(f)\right),$$
and $\hat{\mu}_t^{(1)}(f)$ denotes the estimator obtained by running the algorithm with a single initial particle.

Proof. For each initial particle X_0^i, $i = 1, \cdots, N_0$, define

$$Y_t^i(f) = \mu_0(1/h) \sum_{j=1}^{N_t^i} W_t^{i,j} f(X_t^{i,j}), \tag{4.10}$$

where N_t^i denotes the number of particles in the lineage of the "original ancestor" i at time t, and $(X_t^{i,j}, W_t^{i,j})$ are the corresponding particles and weights of the offspring at time t.

In other words, $Y_t^i(f)$ is precisely the estimator (4.4) restricted to the descendants of a single initial particle. It follows that

$$\hat{\nu}_{t,N_0}(f) = \frac{1}{N_0} \sum_{i=1}^{N_0} Y_t^i(f).$$

Since each lineage evolves independently and the initial particles are i.i.d., the random variables $Y_t^1(f), \cdots, Y_t^{N_0}(f)$ are independent and identically distributed.

Moreover, by Corollary 4.4, each $Y_t^i(f)$ satisfies

$$\mathbf{E}[Y_t^i(f)] = \mathbf{E}_{\mu_0}[f(X_t)].$$

Therefore, the strong law of large numbers yields the almost sure convergence

$$\hat{\nu}_{t,N_0}(f) \to \mathbf{E}_{\mu_0}[f(X_t)] = \mu_0(M^t f).$$

The central limit theorem then gives

$$\sqrt{N_0}\left(\hat{\nu}_{t,N_0}(f) - \mu_0(M^t f)\right) \Rightarrow \mathcal{N}\big(0, \mathrm{Var}(Y_t^1(f))\big).$$

Finally, observing from (4.10) that $Y_t^1(f) = \mu_0(1/h)\, \hat{\mu}_t^{(1)}(f)$, we obtain

$$\mathrm{Var}(Y_t^1(f)) = \mu_0(1/h)^2 \mathrm{Var}\left(\hat{\mu}_t^{(1)}(f)\right),$$

which completes the proof. □

Remark 4.7. Theorem 4.6 holds for the case where we have killing, with the modifications of M for the killed kernel K.

4.3. **Analysis of Variance.** Recall from the previous section that for a fixed $n \in \mathbf{N}$, and $f \in \mathcal{B}_b(E)$, the process $\{\Gamma_t^n(f)\}_{t\leq n}$ defined by

$$\Gamma_t^n(f) = \hat{\mu}_t(M^{n-t}f) - \hat{\mu}_0(M^n f) \overset{4.3}{=} \sum_{s=1}^{t} \hat{\mu}_s(f_s) - \hat{\mu}_{s-1}(Mf_s) := \sum_{s=1}^{t} \Delta_s^t$$

was an $\mathcal{F}_t$-Martingale. In particular, this implied that $\hat{\nu}_t(f)$ is an unbiased estimate for $\mathbf{E}_{\mu_0}[f(X_t)]$ with variance

$$\mathrm{Var}(\hat{\nu}_t(f)) = \frac{\mu_0(1/h)^2}{N_0}\mathrm{Var}\left(\hat{\mu}_t^{(1)}(f)\right),$$

where the superscript in $\hat{\mu}_t^{(1)}(f)$ denotes that the algorithm is run with one initial particle. To obtain a fully explicit formula for $\mathrm{Var}(\hat{\nu}_t(f))$, we need to investigate further $\mathrm{Var}\left(\hat{\mu}_t^{(1)}(f)\right)$. This will be the objective of this section. In particular, we have the following decomposition of variance:

Theorem 4.8. With $\hat{\mu}_t(f)$ as in Equation (4.1), we have that

$$\mathrm{Var}(\hat{\mu}_t(g)) = \sum_{s=1}^{t} \mathbf{E}\left[\sum_{i=1}^{N_{s-1}} M\left(g_s^2\Psi(\cdot,\widehat{W}_s^i)\right)(X_{s-1}^i)\right] \tag{4.11}$$

$$+\sum_{s=1}^{t} \mathbf{E}\left[\sum_{i=1}^{N_{s-1}} \left(\widehat{W}_s^i\right)^2 \left(Mg_s^2 - (Mg_s)^2\right)(X_{s-1}^i)\right] \tag{4.12}$$

$$+\mathrm{Var}(\hat{\mu}_0(M^t g)) \tag{4.13}$$

where

$$\Psi(x,w) = \begin{cases} 0, & \text{if } w \text{ lies inside the weight window at } x, \\ h(x)^2\,\dfrac{w}{h(x)}\left(1-\dfrac{w}{h(x)}\right), & \text{if } w \leq \underline{h}(x), \\ h(x)^2\left\{\dfrac{w}{h(x)}\right\}\left(1-\left\{\dfrac{w}{h(x)}\right\}\right), & \text{if } w > \bar{h}(x). \end{cases}$$

As the decomposition shows, the variance splits into three distinct contributions. The first, (4.11), captures the variance introduced by the correction mechanism, namely the Russian roulette and splitting steps (the more often we perform these corrections, the more extra randomness we introduce). This contribution decreases as the width of the weight window increases. As an extreme example, from the definition of Ψ, we see that when the window is infinitely wide, corrections are never triggered, and so this term vanishes altogether.

In contrast, term (4.12) reflects the variance introduced by the transitions of the chain, amplified by the square of the weight of each particle. A more detailed discussion regarding this term will have to wait until we explore a connection with the Doob transform in section 4.4. The final term, (4.13), simply captures the variance due to the choice of initial measure.

Proof. We begin by noting the following useful decomposition:

$$\hat{\mu}_t(g) - \mathbf{E}[\hat{\mu}_t(g)] = (\hat{\mu}_t(g) - \hat{\mu}_0(M^t g)) + (\hat{\mu}_0(M^t g) - \mathbf{E}[\hat{\mu}_t(g)]) \tag{4.14}$$

$$= (\hat{\mu}_t(g) - \hat{\mu}_0(M^t g)) + (\hat{\mu}_0(M^t g) - \mathbf{E}[\hat{\mu}_0(M^t g)]). \tag{4.15}$$

Where to obtain Equation (4.15), we have used the fact that $\mathbf{E}[\hat{\mu}_t(g)] = \mathbf{E}[\hat{\mu}_0(M^t g)]$. Squaring Equation (4.15) and taking expectations gives

$$\operatorname{Var}(\hat{\mu}_t(g)) = \mathbf{E}\left[\left(\hat{\mu}_t(g) - \hat{\mu}_0(M^t g)\right)^2\right] + \operatorname{Var}(\hat{\mu}_0(M^t g)) \tag{4.16}$$

$$+ \operatorname{Cov}\left(\hat{\mu}_t(g) - \hat{\mu}_0(M^t g),\, \hat{\mu}_0(M^t g) - \mathbf{E}[\hat{\mu}_0(M^t g)]\right). \tag{4.17}$$

Now notice that the covariance term in (4.17) is zero because $\hat{\mu}_0(M^t g) - \mathbf{E}[\hat{\mu}_0(M^t g)]$ is $\mathcal{F}_0$-measurable, and

$$\hat{\mu}_t(g) - \hat{\mu}_0(M^t g) = \sum_{s=1}^{t} \Delta_s^t \quad \text{and} \quad \mathbf{E}[\Delta_s^t \mid \mathcal{F}_0] = 0.$$

Hence,

$$\operatorname{Cov}\left(\hat{\mu}_t(g) - \hat{\mu}_0(M^t g),\, \hat{\mu}_0(g) - \mu_0(M^t g)\right) = \sum_{s=1}^{t} \mathbf{E}\left[\Delta_s^t \times (\hat{\mu}_0(g) - \hat{\mu}_0(M^t g))\right],$$

and by conditioning on $\mathcal{F}_0$ in this last expectation, we see that the covariance equals zero. In summary, our starting point is

$$\operatorname{Var}(\hat{\mu}_t(g)) = \mathbf{E}\left[\left(\hat{\mu}_t(g) - \hat{\mu}_0(M^t g)\right)^2\right] + \operatorname{Var}(\hat{\mu}_0(M^t g)).$$

Fortunately, the expectation above can be understood thanks to Proposition 4.3. In particular, since

$$\hat{\mu}_t(g) - \hat{\mu}_0(M^t g) = \sum_{s=1}^{t} \Delta_s^t,$$

and $\mathbf{E}[\Delta_s^t \mid \mathcal{F}_{s-1}] = 0$, we see, due to orthogonality of the increments, that

$$\mathbf{E}\left[\left(\hat{\mu}_t(g) - \hat{\mu}_0(M^t g)\right)^2\right] = \sum_{s=1}^{t} \mathbf{E}\left[\left(\Delta_s^t\right)^2\right] = \sum_{s=1}^{t} \mathbf{E}\left[\mathbf{E}\left[\left(\Delta_s^t\right)^2 \middle| \mathcal{F}_{s-1}\right]\right] = \sum_{s=1}^{t} \mathbf{E}\left[\operatorname{Var}\left(\Delta_s^t \middle| \mathcal{F}_{s-1}\right)\right]. \tag{4.18}$$

Now we focus on the conditional variance $\operatorname{Var}\left(\Delta_s^t \middle| \mathcal{F}_{s-1}\right)$. We use the law of total variance to insert the intermediate filtration $\mathcal{F}_{s-1} \subset \widehat{\mathcal{F}_s}$, and obtain

$$\operatorname{Var}\left(\Delta_s^t \middle| \mathcal{F}_{s-1}\right) = \mathbf{E}\left[\operatorname{Var}\left(\Delta_s^t \middle| \widehat{\mathcal{F}_s}\right) \middle| \mathcal{F}_{s-1}\right] + \operatorname{Var}\left(\mathbf{E}\left[\Delta_s^t \middle| \widehat{\mathcal{F}_s}\right] \middle| \mathcal{F}_{s-1}\right). \tag{4.19}$$

We now focus on each of these terms individually. First, consider $\operatorname{Var}\left(\Delta_s^t \middle| \widehat{\mathcal{F}_s}\right)$. Recall that Δ_s^t is precisely $\hat{\mu}_s(g_s) - \hat{\mu}_{s-1}(M g_s)$, and since $\hat{\mu}_{s-1}(M g_s)$ is $\mathcal{F}_{s-1} \subset \widehat{\mathcal{F}_s}$-measurable, it follows that

$$\operatorname{Var}\left(\Delta_s^t \middle| \widehat{\mathcal{F}_s}\right) = \operatorname{Var}\left(\hat{\mu}_s(g_s) \middle| \widehat{\mathcal{F}_s}\right).$$

Let us now study this variance:

$$\operatorname{Var}\left(\hat{\mu}_s(g_s)\middle|\widehat{\mathcal{F}_s}\right) = \operatorname{Var}\left(\sum_{i=1}^{N_s} W_s^i g_s(X_s^i)\middle|\widehat{\mathcal{F}_s}\right) \tag{4.20}$$

$$= \operatorname{Var}\left(\sum_{i=1}^{N_{s-1}} W_s^{i\star} M_s^i g_s(\widehat{X}_s^i)\middle|\widehat{\mathcal{F}_s}\right) \tag{4.21}$$

$$= \sum_{i=1}^{N_{s-1}} g_s(\widehat{X}_s^i)^2 \operatorname{Var}\left(W_s^{i\star} M_s^i\middle|\widehat{\mathcal{F}}_s\right), \tag{4.22}$$

where $W_s^{i\star}$ denotes the post-correction weight associated with particle i at time $s-1$, namely the weight inherited by each of its offspring after the correction step.

Now we need to determine $\operatorname{Var}\left(W_s^{i\star} M_s^i\middle|\widehat{\mathcal{F}}_s\right)$. This variance arises from the randomness introduced during the correction step. The key observation is that this conditional variance is a function $\Psi(\widehat{X}_s^i, \widehat{W}_s^i)$ of the position and weight after exploration. We distinguish three cases:

- **Pass:** Suppose that $\widehat{W}_s^i \in (\underline{h}(\widehat{X}_s^i), \bar{h}(\widehat{X}_s^i)]$. Then there is no correction step, so $M_s^i = 1$ and $W_s^{i\star} = \widehat{W}_s^i$, which is $\widehat{\mathcal{F}}_s$-measurable. Hence the conditional variance is zero.
- **Splitting:** Suppose that $\widehat{W}_s^i > \bar{h}(\widehat{X}_s^i)$. Then we perform splitting. The weight of each offspring becomes $h(\widehat{X}_s^i)$, which is $\widehat{\mathcal{F}}_s$-measurable, and the only randomness comes from the number of offspring:
$$M_s^i = \left\lfloor \frac{\widehat{W}_s^i}{h(\widehat{X}_s^i)} \right\rfloor + Y_s^i, \qquad Y_s^i \sim \text{Bernoulli}\left(\frac{\widehat{W}_s^i}{h(\widehat{X}_s^i)} - \left\lfloor \frac{\widehat{W}_s^i}{h(\widehat{X}_s^i)} \right\rfloor\right).$$
In this case the conditional variance is
$$h(\widehat{X}_s^i)^2 \left\{\frac{\widehat{W}_s^i}{h(\widehat{X}_s^i)}\right\}\left(1 - \left\{\frac{\widehat{W}_s^i}{h(\widehat{X}_s^i)}\right\}\right),$$
where $\{a\}$ denotes the fractional part of a.
- **RR:** If $\widehat{W}_s^i \leq \underline{h}(\widehat{X}_s^i)$, then we keep one offspring with probability $\dfrac{\widehat{W}_s^i}{h(\widehat{X}_s^i)}$ and weight $h(\widehat{X}_s^i)$, which again yields conditional variance
$$h(\widehat{X}_s^i)^2 \cdot \frac{\widehat{W}_s^i}{h(\widehat{X}_s^i)}\left(1 - \frac{\widehat{W}_s^i}{h(\widehat{X}_s^i)}\right).$$

In summary,
$$\operatorname{Var}\left(W_s^{i\star} M_s^i\middle|\widehat{\mathcal{F}}_s\right) = \Psi(\widehat{X}_s^i, \widehat{W}_s^i),$$
where
$$\Psi(x, w) = \begin{cases} 0, & \text{if } w \text{ lies inside the weight window at } x, \\ h(x)^2 \dfrac{w}{h(x)}\left(1 - \dfrac{w}{h(x)}\right), & \text{if } w \leq \underline{h}(x), \\ h(x)^2 \left\{\dfrac{w}{h(x)}\right\}\left(1 - \left\{\dfrac{w}{h(x)}\right\}\right), & \text{if } w > \bar{h}(x). \end{cases}$$

Therefore,

$$\begin{aligned}\mathbf{E}\left[\operatorname{Var}\left(\Delta_s^t\middle|\widehat{\mathcal{F}_s}\right)\middle|\mathcal{F}_{s-1}\right] &= \mathbf{E}\left[\sum_{i=1}^{N_{s-1}} g_s(\widehat{X}_s^i)^2\Psi(\widehat{X}_s^i,\widehat{W}_s^i)\middle|\mathcal{F}_{s-1}\right] \\ &= \sum_{i=1}^{N_{s-1}} M\left(g_s^2\Psi(\cdot,\widehat{W}_s^i)\right)(X_{s-1}^i).\end{aligned}$$

Now we return to Equation (4.19) and consider the second term,

$$\operatorname{Var}\left(\mathbf{E}\left[\Delta_s^t\middle|\widehat{\mathcal{F}_s}\right]\middle|\mathcal{F}_{s-1}\right).$$

We first note that

$$\begin{aligned}\mathbf{E}\left[\Delta_s^t\middle|\widehat{\mathcal{F}}_s\right] &= \mathbf{E}\left[\hat{\mu}_s(g_s)-\hat{\mu}_{s-1}(Mg_s)\middle|\widehat{\mathcal{F}}_s\right] \\ &= \mathbf{E}\left[\hat{\mu}_s(g_s)\middle|\widehat{\mathcal{F}}_s\right]-\hat{\mu}_{s-1}(Mg_s) \\ &= \sum_{i=1}^{N_{s-1}} \widehat{W}_s^i g_s(\widehat{X}_s^i)-\hat{\mu}_{s-1}(Mg_s).\end{aligned}$$

Here the third equality follows from the proof of Lemma 4.2. Since $\hat{\mu}_{s-1}(Mg_s)$ is $\mathcal{F}_{s-1}$-measurable, we obtain

$$\begin{aligned}\operatorname{Var}\left(\mathbf{E}\left[\Delta_s^t\middle|\widehat{\mathcal{F}_s}\right]\middle|\mathcal{F}_{s-1}\right) &= \operatorname{Var}\left(\sum_{i=1}^{N_{s-1}} \widehat{W}_s^i g_s(\widehat{X}_s^i)\middle|\mathcal{F}_{s-1}\right) \\ &= \sum_{i=1}^{N_{s-1}} (\widehat{W}_s^i)^2\operatorname{Var}(g_s(\widehat{X}_s^i)\mid\mathcal{F}_{s-1}) \\ &= \sum_{i=1}^{N_{s-1}} (\widehat{W}_s^i)^2\left(Mg_s^2-(Mg_s)^2\right)\left(X_{s-1}^i\right).\end{aligned}$$

Combining and substituting into Equation (4.18), we obtain

$$\begin{aligned}\mathbf{E}\left[\left(\hat{\mu}_t(g)-\hat{\mu}_0(M^t g)\right)^2\right] &= \sum_{s=1}^{t}\mathbf{E}[\operatorname{Var}(\Delta_s^t\mid\mathcal{F}_{s-1})] \\ &= \sum_{s=1}^{t}\mathbf{E}\left[\sum_{i=1}^{N_{s-1}} M\left(g_s^2\Psi(\cdot,\widehat{W}_s^i)\right)(X_{s-1}^i)\right] \\ &\quad+\sum_{s=1}^{t}\mathbf{E}\left[\sum_{i=1}^{N_{s-1}}(\widehat{W}_s^i)^2\left(Mg_s^2-(Mg_s)^2\right)\left(X_{s-1}^i\right)\right].\end{aligned}$$

□

Remark 4.9. In a similar spirit as before, the result for Theorem 4.8 holds for the case of chain with killing under the appropriate modifications: replacing M by K and g_s become $K^{t-s}f$ instead of $M^{t-s}f$.

4.4. **A connection with Doob's transform and a many-to-one formula.** For a function $u : E \to (0, \infty)$ and a transition kernel M, we can define the u-transformed kernel (called Doob's transform) by

$$M^u(x, dy) \propto \frac{u(y)}{u(x)} M(x, dy).$$

The idea behind this construction is to bias the dynamics of the Markov chain so that transitions that increase the value of u are favoured, whereas transitions that decrease it are penalised. In this way, trajectories are tilted toward regions where u is large. See [5, Section 5.2] for more details on the Doob transform.

Let us now consider the Weight Windows Algorithm with target function h. This algorithm does not modify the transition kernel: particles always evolve according to the original dynamics prescribed by M. Instead, the algorithm modifies the particle population through splitting and killing. In transitions where h is reduced, particles are split and therefore become more numerous, whereas in transitions where h is increased, particles are killed with some probability.

Thus, although the individual particle dynamics are unchanged, it seems that the empirical distribution of particles evolves as if transitions toward regions of smaller h were favoured and transitions toward regions of larger h were penalised - the opposite to what the Doob transform does. This suggests a connection between the Weight Windows Algorithm with target function h and the Doob transform associated with the function $u = 1/h$. This intuition is in fact correct, and it is made precise in the following theorem, which resembles the form of a so-called *many-to-one* formula (confer [7, Theorem 1.1], or [1, Lemma 25]).

Theorem 4.10 (Doob transform and Weight Windows)**.** Let a function $u : E \to (0, \infty)$ be given, and consider the weight windows algorithm with target function $h = \frac{1}{u}$, a zero-width weight window, and initial law μ_0. Let $(X_s^i : s \le t)$ for $i = 1, \cdots, N_t$ be the trajectories of the particles alive at time t. Then for a function $F \in \mathcal{B}_b(E^{t+1})$, we have for all $0 \le k \le t-1$

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F\left(X^i_{[0:t]}\right)\middle|\mathcal{F}_k\right] = \sum_{i=1}^{N_k} \mathbf{E}^u\left[\left(\prod_{s=k}^{t-1} Q(X_s)\right) F\left(X^i_{[0:t]}\right)\middle|X^i_{[0:k]}\right] \tag{4.23}$$

where $\mathbf{E}^u$ is expectation with respect to the probability measure under which $(X_s)_{s\ge 0}$ is a Markov chain with dynamics given by the $u-$transformed kernels M^u, and $Q(x) = (Mu)(x)/u(x)$.

Before proving Theorem 4.10, we are going to give the intuition behind Equation 4.23. Assume we wish to compute the conditional expectation of the quantity

$$\sum_{i=1}^{N_t} F\left(X^i_{[0:t]}\right)$$

given all the information up to time k. The theorem shows that this can be understood in the following way. Consider the particles that are alive at time k, and condition on their trajectories up to time k. From time k onward, each particle evolves as an independent Markov chain with dynamics given by the Doob-transformed kernel M^u. Along each such future trajectory, the function F is evaluated, but the contribution of

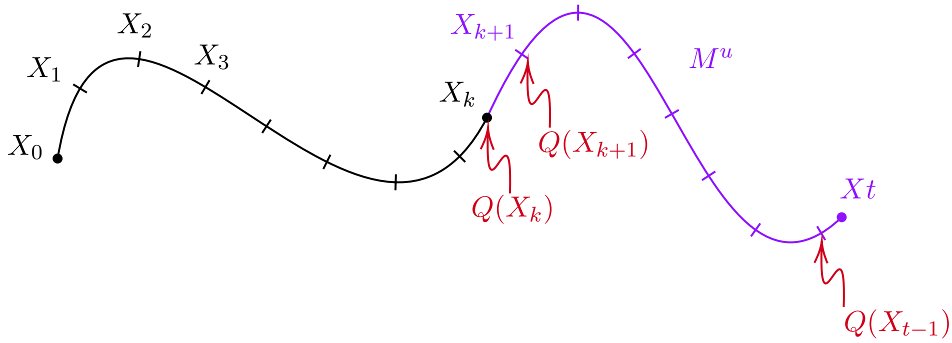

FIGURE 5. The trajectory is fixed up to time k, and then is let to run according to the dynamics prescribed by M^u. The final value of the function evaluated along the trajectory is weighted by the expected future growth of the population.

the trajectory must be multiplied by the factor

$$\prod_{s=k}^{t-1} Q(X_s), \quad \text{where} \quad Q(x) = (Mu)(x)/u(x)$$

The interpretation of this factor is very natural: as will be explained shortly (see Remark 4.12), $Q(x)$ is exactly the expected number of offspring produced by a particle located at x after one step in the zero-width Weight Windows algorithm. Thus, the product $\prod_{s=k}^{t-1} Q(X_s)$ represents the expected population growth along the future trajectory. In other words, the theorem states that the future contribution of the particle system can be computed by letting each particle evolve under the Doob-transformed dynamics and weighting each possible future path by the expected number of descendants generated along that path.

The proof of Theorem 4.10 begins with the following Lemma:

Lemma 4.11 (Expected number of offspring)**.** Consider a particle (X_n, W_n) at time n in Algorithm 4 in the zero-width regime with target function h. Recall that the particle evolves according to the two-step process

$$(X_{t-1}, W_{t-1}) \xrightarrow{\text{exploration}} (\widehat{X}_t, \widehat{W}_t) \xrightarrow{\text{correction}} \left\{\left(X_t^i, W_t^i\right) : i = 1, \cdots, M_t\right\}$$

Then, $\mathbf{E}\left[M_t \mid \widehat{\mathcal{F}}_t\right] = \frac{\widehat{W}_t}{h(\widehat{X}_t)}$, where $\widehat{\mathcal{F}}_t$ is the intermediate filtration defined in Equation 4.3.

Proof of Lemma 4.11. Since we impose a zero-width weight window, we see that each particle undergoes either a Russian roulette or a splitting event at every step (indeed: even if the weight after exploration is exactly the target weight, by looking at Line 7, we see that a Russian Roulette with survival probability 1 is triggered). We therefore examine the number of offspring in both cases.

Russian roulette. By Line 8 in Algorithm 4, particle X_t produces one offspring with probability

$$\frac{\widehat{W}_t}{h(\widehat{X}_t)},$$

and zero offspring otherwise. Hence,

$$\mathbf{E}[M_t \mid \widehat{\mathcal{F}}_t] = \frac{\widehat{W}_t}{h(\widehat{X}_t)}.$$

Splitting. By Line 6 in Algorithm 4, the number of offspring is

$$\left\lfloor \frac{\widehat{W}_t}{h(\widehat{X}_t)} \right\rfloor + Y,$$

where

$$Y \sim \text{Bernoulli}\left(\frac{\widehat{W}_t}{h(\widehat{X}_t)} - \left\lfloor \frac{\widehat{W}_t}{h(\widehat{X}_t)} \right\rfloor \right).$$

It follows that the conditional expectation in this case is also

$$\mathbf{E}[M_t \mid \widehat{\mathcal{F}}_t] = \frac{\widehat{W}_t}{h(\widehat{X}_t)}.$$

□

Remark 4.12 (Relationship between $Q(x)$ and the number of offspring)**.** In the zero-width version of Algorithm 4, a correction is applied at every step, so that we always have

$$\widehat{W}_t = h(X_{t-1}) \quad \text{for all } t.$$

By Lemma 4.11, the expected number of offspring at time t satisfies

$$\mathbf{E}\left[M_t \mid \widehat{\mathcal{F}}_t\right] = \frac{h(X_{t-1})}{h(\widehat{X}_t)}.$$

Taking conditional expectation with respect to $\mathcal{F}_{t-1}$ and recalling that $h = 1/u$, we obtain

$$\mathbf{E}[M_t \mid \mathcal{F}_{t-1}] = \frac{\mathbf{E}\big[u(\widehat{X}_t) \mid \mathcal{F}_{t-1}\big]}{u(X_{t-1})} = \frac{(Mu)(X_{t-1})}{u(X_{t-1})} = Q(X_{t-1}).$$

Therefore, $Q(x)$ has a clear probabilistic interpretation: it is exactly the expected number of offspring produced by a particle located at x after one step in the zero-width Weight Windows algorithm. This explains why the factor $\prod_{s=k}^{t-1} Q(X_s)$ appears in Theorem 4.10: it represents the expected population growth along the future trajectory.

We're now ready to prove Theorem 4.10.

Proof of Theorem 4.10. The proof begins by noting the following identity:

$$\sum_{i=1}^{N_t} F(X_0^i, \cdots, X_t^i) = \sum_{i=1}^{N_{t-1}} M_t^i F(X_0^i, \cdots, X_{t-1}^i, \widehat{X_t^i}).$$

Taking conditional expectation with respect to $\widehat{\mathcal{F}}_t$ and applying Lemma 4.11 along with the fact that since we are performing a correction after every step, we always have $\widehat{W}_n = h(X_{n-1})$, yields

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F(X_0^i, \cdots, X_t^i)\middle|\widehat{\mathcal{F}}_t\right] = \sum_{i=1}^{N_{t-1}} F(X_0^i, \cdots, X_{t-1}^i, \widehat{X}_t^i)\frac{h(X_{t-1}^i)}{h(\widehat{X}_t^i)}.$$

And now, we take expectation with respect to the smaller sigma-algebra $\mathcal{F}_{t-1}$:

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F(X_0^i, \cdots, X_t^i)\middle|\mathcal{F}_{t-1}\right] = \sum_{i=1}^{N_{t-1}} h(X_{t-1}^i)\mathbf{E}\left[\frac{F(X_0^i, \cdots, X_{t-1}^i, \widehat{X}_t^i)}{h(\widehat{X}_t^i)}\middle|\mathcal{F}_{t-1}\right] \tag{4.24}$$

Let's analyse carefully the conditional expectation in the right-hand-side of 4.24. By the Markov Property, if we define $G(x_0, \cdots, x_{t-1})$ by:

$$G(x_0, \cdots, x_{t-1}) = \int_E \frac{F(x_0, \cdots, x_{t-1}, y)}{h(y)} M(x_{t-1}, dy), \tag{4.25}$$

then it follows that

$$\mathbf{E}\left[\frac{F(X_0^i, \cdots, X_{t-1}^i, \widehat{X}_t^i)}{h(\widehat{X}_t^i)}\middle|\mathcal{F}_{t-1}\right] = G(X_0^i, \cdots, X_{t-1}^i).$$

However, if we carefully look at Equation 4.25, we see that

$$G(x_0, \cdots, x_{t-1}) = Q(x_{t-1})u(x_{t-1})\int_E F(x_0, \cdots, x_{t-1}, y)\frac{u(y)}{Q(x_{t-1})u(x_{t-1})} M(x_{t-1}, dy) \tag{4.26}$$

$$= Q(x_{t-1})u(x_{t-1})\int_E F(x_0, \cdots, x_{t-1}, y) M^u(x_{t-1}, dy). \tag{4.27}$$

In other words,

$$\mathbf{E}\left[\frac{F(X_0^i, \cdots, X_{t-1}^i, \widehat{X}_t^i)}{h(\widehat{X}_t^i)}\middle|\mathcal{F}_{t-1}\right] = Q(X_{t-1}^i)u(X_{t-1}^i)\mathbf{E}^u\left[F(X_0^i, \cdots, X_{t-1}^i, X_t^i)\middle|X_{[0:t]}^i\right]. \tag{4.28}$$

Substituting into Equation 4.24 we obtain

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F(X_0^i, \cdots, X_t^i)\middle|\mathcal{F}_{t-1}\right] = \sum_{i=1}^{N_{t-1}} Q(X_{t-1}^i)\mathbf{E}^u\left[F(X_{[0:t]}^i)\middle|X_{[0:t-1]}^i\right] \tag{4.29}$$

$$= \sum_{i=1}^{N_{t-1}} \mathbf{E}^u\left[Q(X_{t-1}^i)F(X_{[0:t]}^i)\middle|X_{[0:t-1]}^i\right]. \tag{4.30}$$

Now a simple (backwards) inductive argument shows that for any $0 \le k \le t-1$, we have that

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F(X_0^i, \cdots, X_t^i)\middle|\mathcal{F}_k\right] = \sum_{i=1}^{N_k} \mathbf{E}^u\left[\left(\prod_{s=k}^{t-1} Q(X_s^i)\right) F(X_{[0:t]}^i)\middle|X_{[0:k]}^i\right] \tag{4.31}$$

Indeed: the base case is already given by Equation 4.30, and so if we assume Equation 4.31 holds for some k, we can show it holds for $k-1$ using the same argument:

(4.32)

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F(X_0^i, \cdots, X_t^i)\middle|\mathcal{F}_{k-1}\right] = \mathbf{E}\left[\mathbf{E}\left[\sum_{i=1}^{N_t} F(X_0^i, \cdots, X_t^i)\middle|\mathcal{F}_k\right]\middle|\mathcal{F}_{k-1}\right]$$

$$= \mathbf{E}\left[\sum_{i=1}^{N_k} \underbrace{\mathbf{E}^u\left[\left(\prod_{s=k}^{t-1} Q(X_s^i)\right) F(X_{[0:t]}^i)\middle|X_{[0:k]}^i\right]}_{\tilde{F}(X_{[0:k]}^i)}\middle|\mathcal{F}_{k-1}\right] \tag{4.33}$$

$$= \sum_{i=1}^{N_{k-1}} \mathbf{E}^u\left[Q(X_{k-1}^i)\tilde{F}(X_{[0:k]}^i)\middle|X_{[0:k-1]}^i\right] \tag{4.34}$$

$$= \sum_{i=1}^{N_{k-1}} \mathbf{E}^u\left[Q(X_{k-1}^i)\mathbf{E}^u\left[\left(\prod_{s=k}^{t-1} Q(X_s^i)\right) F(X_{[0:t]}^i)\middle|X_{[0:k]}^i\right]\middle|X_{[0:k-1]}^i\right] \tag{4.35}$$

$$= \sum_{i=1}^{N_{k-1}} \mathbf{E}^u\left[\left(\prod_{s=k-1}^{t-1} Q(X_s^i)\right) F(X_{[0:t]}^i)\middle|X_{[0:k-1]}^i\right] \tag{4.36}$$

□

We immediately have the following consequence by applying Theorem 4.10 with $k=0$ integrating one final time to remove the randomness in X_0:

Corollary 4.13. With the same notation as in Theorem 4.10, we have that

$$\mathbf{E}\left[\left(\frac{N_t}{N_0}\right)\frac{1}{N_t}\sum_{i=1}^{N_t} F(X_{[0:t]}^i)\right] = \mathbf{E}_{\mu_0}^u\left[\left(\prod_{s=0}^{t-1} Q(X_s)\right) F(X_{[0:t]}^i)\right].$$

Where

$$\mathbf{E}_{\mu_0}^u[H(X_0, \cdots, X_t)] = \int_E \mathbf{E}^u[H(X_0, \cdots, X_t)|X_0 = x]\mu_0(dx)$$

Remark 4.14. Notice that by using $F=1$ in the corollary above, we see that

$$\mathbf{E}\left[\frac{N_t}{N_0}\right] = \mathbf{E}_{\mu_0}^u\left[\prod_{s=0}^{t-1} Q(X_s)\right].$$

We can write this in a further more illustrative way:

$$\mathbf{E}\left[\prod_{s=0}^{t-1} \frac{N_{s+1}}{N_s}\right] = \mathbf{E}_{\mu_0}^u\left[\prod_{s=0}^{t-1} Q(X_s)\right].$$

This is of course not a surprise, since as we discussed in Remark 4.12, $Q(x)$ is the expected number of offspring produced by a particle currently at x after its exploration step under the Weight Windows Algorithm. In the case where we have a kernel M and a function u that satisfy an equation of the type $Mu = \lambda u$, the equation above collapses to a simple $\mathbf{E}[N_t/N_0] = \lambda^t$.

4.4.1. *Back to the variance of $\hat{\mu}_t$.* As promised, we now return to the variance decomposition of Theorem 4.8, and in particular to Term (4.12). Recall from the discussion following Theorem 4.8 that Term (4.11) can be interpreted as the variance introduced by the correction mechanism. In particular, as the weight windows become infinitely wide, this term vanishes.

It is then natural to ask how the choice of weight windows influences Term (4.12). At a heuristic level, this term resembles a weighted sum of squares, and it is well known that, under a fixed total mass constraint, such sums are minimised when the summands are equal. This suggests that enforcing frequent corrections—potentially at every step, as in the zero-width regime—may help reduce this contribution to the variance.

Of course, the situation here is more delicate: the total mass is not conserved, and the squared weights are further multiplied by local transition variances. Nonetheless, it remains plausible that tighter control of the weights may lead to a reduction of Term (4.12).

While we have not been able to establish this rigorously, we now provide a heuristic argument based on the example of a simple symmetric random walk (SSRW) on the interval $E_k := \{-k+1, \ldots, k-1\}$, killed upon reaching $\pm k$.

Example 4.15 (Random Walk with Killing)**.** Consider a SSRW on the interval $E_k := \{-k+1, \ldots, k-1\}$ with killed kernel

$$K(x, y) = \frac{1}{2}\mathbf{1}\{y = x - 1,\ |y| < k\} + \frac{1}{2}\mathbf{1}\{y = x + 1,\ |y| < k\}.$$

It is a simple calculation to check that K has principal eigenpair u and λ given by

$$u(x) = \cos\left(\frac{\pi x}{2k}\right), \qquad \lambda = \cos\left(\frac{\pi}{2k}\right).$$

Let us write Term (4.12) as

$$A_s = \mathbf{E}\left[\sum_{x \in N_{s-1}} (\widehat{W}_s^x)^2 V_s(X_{s-1}^x)\right], \qquad V_s(y) = \mathrm{Var}(g_s(\widehat{X}_s) \mid X_{s-1} = y).$$

We wish to understand how the choice of weight windows affects A_s. To this end, consider weight windows of the form

$$\underline{h} = \frac{h}{\alpha}, \qquad \bar{h} = \alpha h, \qquad \alpha = 1 + \varepsilon,$$

where $\varepsilon > 0$ is small. Since the width of the windows is small, the weights remain close to the function h. More precisely, for each particle x at time $s-1$, we may write

$$\left(\widehat{W}_s^x\right)^2 = h(X_{s-1}^x)^2 \left(R_{s-1}^x\right)^2,$$

where

$$R_{s-1}^x := \frac{\widehat{W}_s^x}{h(X_{s-1}^x)} \in \left[\frac{1}{\alpha}, \alpha\right].$$

Since R_{s-1}^x remains close to 1 when ε is small, we may, at a heuristic level, focus on the leading contribution

$$A_s \approx \mathbf{E}\left[\sum_{x \in N_{s-1}} h(X_{s-1}^x)^2 V_s(X_{s-1}^x)\right]. \tag{4.37}$$

Since we are working with a SSRW, a simple computation gives

$$V_s(x) = \frac{1}{4}\left(g_s(x+1) - g_s(x-1)\right)^2, \tag{4.38}$$

where $g_s(\pm k) = 0$. Now, if we are interested in computing $\mathbf{P}_x(H^\partial > t)$, we want to estimate $\mu_0(K^t 1)$, and hence

$$g_s(x) = (K^{t-s}1)(x).$$

We now work in the time regime $1 \ll s \ll t$. In particular, since $t-s$ is large, we approximate $(K^{t-s}1)(x)$ by its principal spectral component:

$$g_s(x) \approx c_k \lambda^{t-s} u(x),$$

where

$$c_k = \frac{\langle 1, u\rangle}{\langle u, u\rangle} = \frac{\sum_{|y|<k} u(y)}{\sum_{|y|<k} u(y)^2} = \frac{1}{k}\cot\left(\frac{\pi}{4k}\right).$$

Plugging this approximation into (4.38) gives, in the regime $s \ll t$,

$$\begin{aligned} V_s(x) &\approx C_k \lambda^{2(t-s)} \left(u(x+1) - u(x-1)\right)^2 \\ &= C_k' \lambda^{2(t-s)} \sin^2\left(\frac{\pi x}{2k}\right). \end{aligned} \tag{4.39}$$

Combining (4.39) with (4.37), and using $h = 1/u$, gives

$$\begin{aligned} A_s &\approx C_k' \lambda^{2(t-s)} \mathbf{E}\left[\sum_{x \in N_{s-1}} \frac{1}{\cos^2\left(\frac{\pi x}{2k}\right)} \sin^2\left(\frac{\pi x}{2k}\right)\right] \\ &= C_k' \lambda^{2(t-s)} \mathbf{E}\left[\sum_{x \in N_{s-1}} \tan^2\left(\frac{\pi x}{2k}\right)\right]. \end{aligned} \tag{4.40}$$

From Equation (4.40) we see that, under this approximation, the dominant effect of the choice of weight windows is through the distribution of particle locations. In particular, since $\tan^2(\pi x/2k)$ grows large as $|x| \to k$, the term A_s becomes large if particles spend significant time near the boundary points $\pm k$.

At this point, the intuition for why narrow weight windows improve A_s becomes clear. As shown in Corollary 4.13, if the width of the weight window is zero, Equation (4.40) can be evaluated using the Doob-transformed dynamics. In this case,

$$A_s \approx C_k'' N_0 \lambda^{2t-(s+1)} \mathbf{E}^u\left[\tan^2\left(\frac{\pi X_{s-1}}{2k}\right)\right]. \tag{4.41}$$

Under the Doob-transformed dynamics, the chain is pushed towards 0, making visits close to the boundary points $\pm k$ infrequent. Moreover, the invariant measure of the Doob-transformed kernel K^u is proportional to $u(x)^2$. Therefore, if $s \gg 1$ so that the transformed chain has had time to mix, then

$$\begin{aligned} \mathbf{E}^u\left[\tan^2\left(\frac{\pi X_{s-1}}{2k}\right)\right] &\approx \mathbf{E}_{\pi^u}\left[\tan^2\left(\frac{\pi X}{2k}\right)\right] \\ &= \frac{\sum_{|x|<k} \sin^2\left(\frac{\pi x}{2k}\right)}{\sum_{|x|<k} \cos^2\left(\frac{\pi x}{2k}\right)} = \frac{k-1}{k} < 1. \end{aligned} \tag{4.42}$$

Thus, under the Doob transform, the inward push is sufficient to keep the expectation in Equation (4.41) bounded above by 1.

By contrast, when the weight window is widened, the particle distribution is no longer governed exactly by the Doob transform. Even a small increase in the time spent near the boundary can significantly increase the expectation in (4.40), due to the growth of the $\tan^2$ term.

This suggests that, at least heuristically and in the regime $1 \ll s \ll t$, narrower weight windows—and in particular the zero-width regime—are favourable for controlling the contribution of Term (4.12).

We conclude the report by noting a few comments on the case where we don't immediately take $\widehat{W}_t = W_{t-1}$, but rather introduce an extra factor that is $\mathcal{F}_{t-1}$-measurable.

4.5. **A variant of the algorithm.** From Corollary 4.13, we know that

$$\mathbf{E}[N_t/N_0] = \mathbf{E}^u\left[\prod_{s\le t} Q(X_s)\right].$$

Note that in the case where $Mu = \lambda u$, then this right hand side simplifies to λ^t. In this particular example, if $\lambda < 1$, then on average, population decreases after every generation, and so it could be that our algorithm dies out. Conversely, if $\lambda > 1$, population is increasing on average by a factor of λ at each generation, so it may quickly grow too large. An idea is then to modify the algorithm so that the ratio N_t/N_0 remains 1 in expectation. The trick will be in Line 4 of Algorithm 4, instead of simply taking $\widehat{W}_{n+1} = W_n$, we can add an extra factor that will correct for the number of offspring. Indeed, suppose that $G : E \to [0,\infty)$ is given, then we can replace

$$\widehat{W}_{n+1} \leftarrow W_n \qquad \text{by} \qquad \widehat{W}_{n+1} = G(X_n)W_n.$$

We will refer to this variant as the modified version of the Weight Windows Algorithm. We now investigate how Lemma 4.2 and Theorem 4.10 change after this modification. The first is simple. Indeed, as shown in the Proof of Lemma 4.2, we know that for a single particle performing a step of the algorithm, we have

$$\mathbf{E}\left[\sum_{i=1}^{M_n} f(X_n^i)W_n \middle| \mathcal{F}_{n-1}\right] = \widehat{W}_n(Mf)(X_{n-1}), \tag{4.43}$$

so in our modified case, this right-hand side becomes $G(X_{n-1})(Mf)(X_{n-1})$. The argument now goes very much like that of Equations (4.32) through (4.36). Indeed,

$$G(X_{n-1})(Mf)(X_{n-1}) = \mathbf{E}[G(X_{n-1})f(X_n)|\mathcal{F}_{n-1}],$$

so, if we were to condition 4.43 on $\mathcal{F}_{n-2}$, the right hand side would become

$$G(X_{n-2})\mathbf{E}[G(X_{n-1})\mathbf{E}[f(X_n)|\mathcal{F}_{n-1}]|\mathcal{F}_{n-2}] = \mathbf{E}[G(X_{n-2})G(X_{n-1})f(X_n)|\mathcal{F}_{n-2}].$$

Iterating down, we get that

$$\mathbf{E}\left[\sum_{i=1}^{M_n} f(X_n^i)W_n \middle| \mathcal{F}_k\right] = \mathbf{E}\left[\left(\prod_{s=k}^{n-1} G(X_s)\right) f(X_n) \middle| \mathcal{F}_k\right].$$

A similar modification applies to Theorem 4.10. Indeed, Equation 4.24 becomes

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F(X_0^i,\cdots,X_t^i)\middle|\mathcal{F}_{t-1}\right] = \sum_{i=1}^{N_{t-1}} G(X_{t-1}^i)h(X_{t-1}^i)\mathbf{E}\left[\frac{F(X_0^i,\cdots,X_{t-1}^i,\widehat{X_t^i})}{h(\widehat{X_t^i})}\middle|\mathcal{F}_{t-1}\right],$$

and the G can be absorbed into the F. The computation then carries through in the same way, and we obtain the modified version of Equation (4.23), which reads

$$\mathbf{E}\left[\sum_{i=1}^{N_t} F\left(X^i_{[0:t]}\right)\middle|\mathcal{F}_k\right] = \sum_{i=1}^{N_k} \mathbf{E}^u\left[\left(\prod_{s=k}^{t-1} G(X_s)Q(X_s)\right) F\left(X^i_{[0:t]}\right)\middle| X^i_{[0:k]}\right].$$

A consequence is that if we choose $G = 1/Q$, we obtain that $\mathbf{E}[N_t/N_0] = 1$.

5. Conclusion and Future Work

In this report, we have studied a collection of importance-splitting variance reduction algorithms for rare event problems, including an overview of Multilevel Splitting and its adaptive variant, as well as an in-depth analysis of the Weight Windows algorithm. Regarding the latter, we have stated and proved results concerning unbiasedness, convergence, variance analysis, and a connection with Doob's h-transform which, to the best of our knowledge, are not currently available in the mathematical literature.

An important limitation of the Weight Windows algorithm, as studied in this report, is the random nature of the particle population size. Indeed, as illustrated in Corollary 4.13, in the zero-width case it is easy to see that if $(Mu)(x) > \lambda u(x)$ for some $\lambda > 1$, then the particle population grows exponentially in expectation. Conversely, if $(Mu)(x) < \lambda u(x)$ for some $\lambda < 1$, then the particle population decreases exponentially and the algorithm may terminate prematurely. It is therefore natural to consider extensions of the algorithm in which an explicit population control mechanism is introduced. For example, when updating the weights, instead of taking naively $\widehat{W}^i_t = W^i_{t-1}$, as was done throughout this report, one could include a compensating factor accounting for the current population growth, such as

$$\widehat{W}^i_t = W^i_{t-1} \times \frac{N_{t-2}}{N_{t-1}}.$$

However, the introduction of such terms complicates the analysis, since the factor N_{t-2}/N_{t-1} depends on the entire population rather than on individual parent particles. For example, genealogies would no longer evolve independently. The study of such population-controlled variants therefore constitutes a natural direction for future work.

References

[1] Julien Berestycki. Topics on branching brownian motion, 2014. Lecture notes.
[2] Thomas Booth. Genesis of the weight window and the weight window generator in mcnp -a personal history los alamos report la-ur-06-5807, 08 2006.
[3] Frédéric Cérou, Arnaud Guyader, and Mathias Rousset. Adaptive multilevel splitting: Historical perspective and recent results. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29:043108, 04 2019.
[4] Pierre Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems With Applications*. 05 2004.
[5] Jan Olucha Fuentes. Feynman-kac formulae and sequential monte-carlo. Reading course report, University of Bath, 2025.
[6] Los Alamos Scientific Laboratory. Mcnp – a general monte carlo code for neutron and photon transport. Technical Report LA-7396-M, Rev. 1, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1979.
[7] Zhan Shi. *Branching Random Walks*, volume 2151 of *Lecture Notes in Mathematics*. Springer, Cham, 2015. École d'Été de Probabilités de Saint-Flour XLII – 2012.
[8] W. L. Thompson and Edmond D. Cashwell. The status of monte carlo at los alamos. Technical report, Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1980.